

움직임 에너지를 이용한 동적 물체 추적 시스템의 실시간 구현

류성희, 김진율
수원대학교 전자공학과

Real-time Implementation of a DSP System for Moving Object Tracking Based on Motion Energy

Sung Hee Ryu, Jin Yul Kim
Department of Electronic Engineering, University of Suwon

Abstract - This work describes a real-time method, based on motion energy detection, for detecting and tracking moving object in the consecutive image sequences. The motion of moving objects is detected by taking the difference of the two consecutive image frames. In addition an edge information of the current image is utilized in order to further increase the accuracy of detection. We can track the moving objects continuously by detecting the motion of objects from the sequence of image frames. A prototype system has been implemented using a TI TMS320C6201 EVM fixed-point DSP board, which can successfully track a moving human in real-time.

1. 서 론

최근, 지하 주차장이나 은행의 무인 자동화 창구, 공장 등 보안이나 범죄예방이 요구되는 구역에서 감시 카메라의 영상을 입력받아 경비인력 없이 외부 침입자를 식별하고, 이동 방향을 연속으로 계산하여 움직임 물체를 지속적으로 추적할 수 있도록 해주는 영상처리 시스템의 필요성이 크게 증대되고 있다. 특히, 보안구역을 24시간 감시해야 할 경우, 경비인력을 줄이기 위해선 무인 감시 능력이 절대적으로 필요하다. 이러한 무인 감시 시스템의 요구사항을 만족시키기 위하여 CCD 감시 카메라의 영상으로부터 사람이나 물체의 움직임을 자동으로 감지하고, 움직임 영역의 최소 경계영역(Minimum Bounding Box, MBB)을 추출하여 물체를 연속적으로 추적하는 기능을 탑재한 감시 카메라 시스템을 DSP (Digital Signal Processor)를 사용하여 실시간으로 구현하는 것이 본 논문의 주된 목적이다.

기존에 영상신호로부터 물체의 움직임을 추적하는 여러 방법이 제안되어 왔으나 대부분이 실시간을 고려하지 않고 계산량이 많은 수학적 접근법을 사용하여 시뮬레이션을 수행하거나 알고리즘이 지나치게 복잡하여 실시간 구현에 용이하지 않다[1, 2, 3]. 특히, 명암 및 색깔 정보를 포함한 많은 양의 영상 데이터를 처리해야 하는 알고리즘[4], 영상처리 후반부에 단일 DSP로 구현하기 어려운 복잡한 영역 라벨링(Region Labeling), 클러스터링 및 매칭(Region Clustering/Matching) 과정을 적용한 알고리즘[5], 그리고 반복연산이 많은 Hidden Markov Model, Kalman Filter 등의 복잡한 확률/통계적 접근방법을 적용한 알고리즘[6, 7] 등은 최소의 하드웨어 자원으로 실시간 구현해야 하는 본 논문에는 적합하지 않다. 따라서 본 논문에서는 비교적 용이하게 실시간으로 구현할 수 있는 움직임 에너지 접근법(Motion Energy Approach)을 바탕으로 움직이는 물체를 감지/추적하는 알고리즘을 제시하고 TMS320C6201 고정소수점 DSP를 이용하여 실시간 구현한 결과에 대하여 기술한다. 제시한 추적 알고리즘의 구성은 1) 감시지역 내 움직임의 감지, 2)배경영역으로부터의 움직임 영역의 분리/격리, 3)움직임 영역의 최소 경계영역(MBB) 추출,

4)연속 영상으로부터 최소 경계영역(MBB) 추적을 통한 움직임 방향 추적 등으로 이루어진다.

본 논문의 구성은 다음과 같다. 먼저 2.1장에서 움직임 물체 추적 알고리즘에 대해 설명하고, 2.2장에서 TMS320C6201 EVM(Evaluation Module)을 사용한 데모 시스템의 구성을 간략히 기술한다. 그리고 2.3장에서 실시간으로 구현된 데모 시스템의 수행 결과에 대해 설명하며, 마지막으로 3장에서 결론을 기술한다.

2. 본 론

2.1 움직임 물체 감지 및 추적 알고리즘

본 논문에서 실시간 구현한 알고리즘은 움직임 에너지 접근법(Motion Energy Approach)을 모태로 사용한 것이며, 이 접근방법은 추적하고자 하는 물체의 3D 모델 등 추가 정보없이 움직임 감지에만 의존하여 영상내의 물체를 추적하는 방법이다. 따라서 이 움직임 에너지 접근방법은 추적 대상의 크기나 모양 등의 특성을 고려하지 않고도 움직임 물체의 감지 및 추적을 수행할 수 있는 장점이 있다. 이러한 접근방법을 적용한 움직임 물체 추적 알고리즘의 전체 구성도는 그림 1과 같이 도시된다.

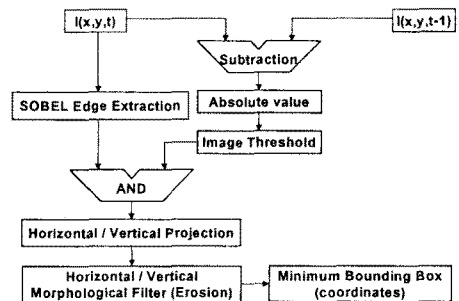


그림1: 움직임 감지 / 추적 과정 구성도

그림1의 추적 알고리즘은 움직임 에너지를 감지하기 위해 연속된 두 영상의 차를 구한 후, 적절한 임계값으로 이진 차영상을 얻는다. 이 이진 차영상은 물체의 움직임 에너지를 나타내므로 이 영상을 통해 영상내에 움직임 물체 유무를 판별하게 된다. 그리고 움직임 물체의 정확한 현재 위치를 구하기 위해 현재 영상의 에지 영상을 추출하여 이진 차영상과 AND 논리연산을 취한다[8]. 이렇게 얻어진 영상은 현재 위치의 움직임 물체 에지만을 나타내는 움직임 에지영상이다. 참고문헌[8]에서의 추적 알고리즘은 AND 논리연산을 하기 전에 이진 차영상에 대하여 형태학적 필터링인 오픈링(Opening) 연산을 취하여 이진 차영상내에 존재하는 잡음성분을 제거하여 보다 정밀한 움직임 에지영상을 구한다. 이 형태학적 필터링에서 사용하는 마스크의 크기와 제거될 수 있는 잡음성분의 픽셀크기는 비례하는 성질이 있는데 보다 정

밀한 움직임 에지영상을 구하기 위해서는 마스크 크기가 커야 한다. 하지만 마스크 크기가 클수록 그 연산량이 많아지는 단점이 있다. 본 논문에서는 이진 차영상과 현재 프레임의 에지 영상의 AND 논리연산을 취한 후, 움직임 물체의 최소 경계영역(MBB)의 좌표를 추출하기 위해 수평/수직방향으로 투영(Horizontal/Vertical Projection)시켜 x-축/y-축의 1차원 배열을 얻는다. 그리고 잡음성분을 제거하여 보다 정밀한 움직임 최소 경계영역(MBB)을 추출하기 위해 수직/수평방향으로 투영시켜 얻은 x-축/y-축의 배열에 대하여 각각 1차원 형태학적 필터링을 취한다. 이러한 1차원 형태학적 필터링은 참고 문헌[8]에서 제시한 2차원 영상에 대한 형태학적 필터링에 비해 그 연산량을 감소시키는 효과를 얻는다.

개략적으로 언급한 알고리즘 설명을 정리하면 그림1의 추적 알고리즘은 크게 세 부분으로 구분될 수 있는데, 첫 번째가 영상내 움직임 물체를 감지하고 감지된 움직임 영역을 배경영역과 분리(Segmentation)하기 위한 움직임 에너지 추출 부분이고, 두 번째는 움직임 물체의 현재위치를 얻어내는 움직임 에지 영상 추출 부분이다. 마지막으로 세 번째는 두 번째 단계에서 얻은 움직임 에지 영상으로부터 수평/수직방향 투영(Horizontal/Vertical Projection)을 통해 움직임 물체의 최소 경계영역(MBB)을 얻어내어 그 물체의 이동위치를 연속적으로 추적하는 부분이다. 다음은 각 부분에 대하여 기술한 것이다.

2.1.1 움직임 에너지 추출

움직임 추적을 수행하기 위해서는 먼저 영상내에서 움직임을 감지하고 움직이는 물체를 배경화상으로부터 분리(Segmentation)시켜야 한다. 카메라로부터 연속으로 들어오는 영상신호에서 움직임을 감지하기 위한 기본원리는 연속된 영상에, 시간에 대한 미분을 취하여 얻는 각 픽셀(Pixel) 명암값의 변화량을 이용하는 것이다. 즉, 적당한 임계값 이상인 변화량만을 움직임 에너지(Motion Energy)로 추출하여 움직이는 물체를 감지해내고 움직임 에너지를 갖는 픽셀만을 영상내의 움직임 영역으로 판별하는 방법으로 움직임 영역과 배경영역을 분리한다. 따라서 변화량이 작거나 없는 픽셀은 배경영역내에 포함된 것으로, 반대의 경우는 움직임 영역내의 픽셀로 판별할 수 있다. 본 논문의 알고리즘은 움직임 에너지를 추출하기 위해 식(1)과 같이 시간에 대한 미분을 연속된 영상내 픽셀들 사이의 차(Image Subtraction)를 구함으로써 근사화한다[8, 9].

$$\frac{dI(x, y, t)}{dt} \cong I(x, y, t) - I(x, y, t - \Delta t) \quad \text{식(1)}$$

식(1)에서 $I(x, y, t)$ 은 시간 t 에 입력된 영상 F 의 (x, y) 에 위치한 픽셀의 명암값(Intensity Value)을 나타내며, 이때 픽셀의 위치를 나타내는 (x, y) 는 $x, y \in F$ 를 만족한다. 그리고 Δt 는 연속된 두영상의 시간 간격이다. 식(1)과 같이 구한 차영상에 절대값을 취한 후, 이에 적절한 임계값으로 이진화하여 움직임 영역의 분리(Segmentation)와 함께 영상내의 잡음을 제거하므로써 이진 차영상을 얻는다. 이 이진 차영상은 움직임 에너지를 추출한 결과인 동시에 움직임 영역과 배경영역을 분리한 결과이다. 2.3장의 그림4이 본 논문을 통해 구현한 데모 시스템에서 얻은 이진 차영상이다.

2.1.2 움직임 에지 추출

움직임 에너지 추출을 통해 얻은 이진 차영상은 2.3장의 그림4에서 볼 수 있듯이 현재 영상의 움직임 영역과 더불어 이전 영상의 움직임 자취가 나타나게 된다. 따라서 현재 영상으로부터 에지영상을 추출한 후, 이진 차영상과의 AND 논리연산을 수행하여 움직임 에지 영상(Moving Edge Image)을 얻는다. 이 움직임 에지영상은 보다 정확한 현재 위치의 움직임 물체 에지만을 나타낸다.[1]. 움직임 에지영상을 얻기 위해 우선 현재 영상에서 에지정보를 추출해야 한다. 이를 위해 식(2)의 3x3 소벨 마스크(SOBEL Edge-detection Mask)로 각각 독립적으로 현재 프레임의 영상과 연산하며, 각 마스크

연산 결과에 절대값을 취한 후 식(3)과 같이 서로 더하여 영상의 에지를 추출한다(그림5)[10]. 이렇게 추출한 현재 프레임의 에지영상은 앞서 구한 이진 차영상과 AND 논리연산을 취하여 움직임 에지영상을 얻는다. 2.3장의 그림6가 움직임 에지영상의 한 예이다.

$$A_h = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, A_v = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{식(2)}$$

$$G = |A_h| + |A_v| \quad \text{식(3)}$$

소벨 알고리즘으로 구한 에지영상은 움직임 물체의 공간적인 위치정보를 나타낸다. 결국, 그림1에서 도시한 것과 같이 연속된 영상의 시간적 변화량인 이진 차영상과 물체의 공간적인 위치정보인 에지영상의 AND 논리연산은 움직이는 물체의 정확한 현재 위치만을 찾기 위해 입력영상의 시간/공간적인 정보를 모두 이용함을 의미한다. 이진 차영상과 에지영상의 AND 논리연산 결과로 얻어진 움직임 에지 영상으로부터 움직임 물체를 연속적으로 추적하기 위해 움직임 에지의 MBB를 추출한다. 다음은 이에 대한 설명이다.

2.1.3 움직임 물체 추적

추적 알고리즘의 최종 단계는 앞 단계에서 얻은 움직임 에지 영상으로부터 최소 경계영역(MBB)을 추출하여 이를 이용해 움직임 물체의 이동위치를 연속적으로 추적하는 것이다. 이를 위해 우선 2차원의 움직임 에지 영상을 그림2와 같이 수평/수직방향으로 투영(Horizontal/Vertical Projection)하여 1차원 위치정보로 변환한 후, 적당한 임계값 이상의 투영 결과만을 최소 경계영역(MBB)으로 결정하고 그 좌표를 추출한다. 최소 경계영역(MBB)의 좌표를 사용하여 영상내의 움직임 물체의 이동위치를 추적한다. 그림2의 수평/수직방향 투영은 움직임 에지영상에서 에지성분을 갖는 픽셀수를 수평/수직방향으로 누산함으로써 얻는다. 움직임 에지영상으로부터 수평/수직방향으로 투영시켜 얻은 각 x-축/y-축 1차원 배열은 2차원 영상내의 움직임 물체에 대한 1차원 위치정보를 나타낸다. 투영 결과로 얻은 x-축/y-축 위치정보의 한 예가 2.3장의 그림8, 9에 도시되어 있다.

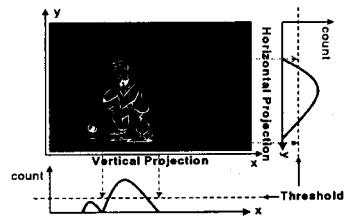


그림2: 수평 / 수직방향 투영

본 논문에 기술된 움직임 에너지 접근방법을 통한 움직임 물체 추적 알고리즘의 정확성 유무는 알고리즘의 초기단계인 배경영역과 움직임영역의 성공적인 분리(Segmentation)에 달려 있다. 그 이유는 조도 변화 등의 원인에 따른 배경영역내의 변화에 의해 작은 수의 픽셀들로 이루어진 백색잡음이 존재하여, 실제 움직임 물체외의 영역을 움직임 영역으로 잘못 인식하기 때문이다. 따라서 참고문헌[8]에서는 움직임 에지영상을 추출하기 앞서 이진 차영상에 대해 형태학적 필터링(Morphological Filtering)을 취하므로써 백색잡음을 제거한다. 이진 차영상에 대한 형태학적 필터링은 영상내에 존재하는 물체의 외형에 영향을 주지 않고 잡음을 제거하거나, 그 물체의 형태를 복원하고 강조시킬 때 흔히 사용되며, 그 대상은 영상내의 물체뿐만 아니라, 숫자, 기호, 문자 등이 될 수 있다[10]. 형태학적 필터링은 여러가지 방법이 있으며, 이들 중 많이 사용되는 방법은 영상확장(Dilation), 영상수축(Erosion) 그리고 이 둘을 조합시킨 Opening과 Closing 등이 있다[10]. 영상확장은 영상내에서 좁은 범위를 차지하는 영역을 전체적인 형태에는 영향없이 확장시키는 결과를 얻을 수 있고 영상수

축은 마스크 크기보다 작은 영역을 제거시키는 결과를 얻는다. 특히, Opening은 영상확장과 영상수축 연산을 조합하여 사용하므로써 마스크 크기보다 작은 픽셀 크기를 갖는 영상 잡음들을 제거하는 효과를 얻게 된다. 즉, 영상수축 연산과정에서 영상에 남아있는 잡음들이 제거되며, 영상확장 연산과정시 잡음이 제거된 영상내 대상 물체의 모양에 영향없이 그 크기와 형태가 복원 및 강조되는 효과를 낸다. 그리고 형태학적 필터링시 사용되는 마스크 크기와 제거될 수 있는 잡음의 픽셀 크기는 비례관계가 있음을 알 수 있다.

참고문헌[8]에서는 배경영역에 남아있는 잡음을 제거하여 보다 정밀한 움직임 에지영상을 얻기 위해 9×9 마스크를 사용하여 이전 차영상에 대해 형태학적 필터링인 Opening 연산을 취한다. 하지만 이 Opening 연산은 2차원 이전 차영상에 대해 9×9 마스크를 사용하여 각각 영상수축과 영상확장 연산을 함으로써 이루어지는데, 그 계산량이 많기 때문에 실시간으로 구현하기에는 적합하지 않다. 따라서 본 논문에서는 움직임 에지영상을 얻어 움직임 물체의 정확한 현재위치를 추출한 후, 그림2와 같이 수평/수직방향으로 투영시켜 얻은 움직임 물체의 위치정보인 x-축/y-축 1차원 배열에 식(4)와 같이 1×3 마스크로 수평/수직 형태학적 필터링(Horizontal/Vertical Morphological Filtering)을 취한다.

$$M_h = M_v = [m_0 \ m_1 \ m_2], \quad m_i = 0,1 \text{ and } i = 0,1,2 \text{ 식(4)}$$

예를 들어 $M \times N$ 크기의 움직임 에지영상으로부터 수직방향으로 투영한 결과로 얻은 x-축 1차원 배열을 식(5)과 같이 나타낼 수 있다. x-축의 투영 결과인 x_i 에 대하여 1차원 수평 형태학적 필터링을 위한 결과인 MO_1 은 식(6)로 기술된다. 식(6)에서 얻은 MO_1 의 값은 0, 1의 논리값으로 표현되며, 수평방향 투영 결과인 y-축 배열도 식(6)과 마찬가지로 기술된다. 그리고 식(6)에서 볼 수 있듯이 이 1차원 수평/수직 형태학적 필터링은 수평/수직방향 투영 결과에 대해 영상수축 연산과 임계값을 적용한 것임을 확인할 수 있다. 이로써 계산량이 현저히 감소되는 결과를 얻을 뿐만 아니라 참고문헌[1]의 Opening 연산 결과와 마찬가지로 배경영역의 백색잡음을 제거하는 효과를 얻을 수 있다.

$$PRJ_x = [x_0 \ x_1 \ x_2 \ \dots \ x_M] \quad \text{식(5)}$$

$$MO_1 = (m_0 \text{ AND } (x_0 > TH_h)) \text{ AND} \\ (m_1 \text{ AND } (x_1 > TH_h)) \text{ AND} \\ (m_2 \text{ AND } (x_2 > TH_h)), \quad TH_h: \text{ Threshold value} \quad \text{식(6)}$$

$$x_i \in \text{Moving Object if } MO_1 = 1 \quad \text{식(7)}$$

수평/수직방향으로 투영하여 얻은 1차원 위치정보인 x-축/y-축 배열에서 움직임 물체의 최소 경계영역(MBB)으로 추출되는 조건은 식(7)과 같다. 또한, 앞서 설명한 수평/수직방향 투영에 반영된 잡음 성분이 수평/수직 형태학적 필터링을 통해 제거되고 움직임 물체만을 정확히 인식할 수 있는 근거는 다음과 같이 설명된다. 움직임 영역으로부터 투영되었을 경우, 1차원 x-축/y-축 상에서의 투영된 값의 크기가 클 뿐만 아니라 조밀한 분포를 갖고, 백색잡음이 존재하는 배경영역으로부터 투영되었을 경우는 투영된 값의 크기가 작거나 그 분포가 조밀하지 않은 결과를 얻을 것이다. 그러므로 인접한 세 개의 픽셀 위치에 대한 투영된 결과값이 모두 적절한 임계값 이상인 경우, 그 영역이 움직임 물체라고 판별한다. 배경영역에 흩어져 있는 백색잡음으로부터 투영된 경우라도 그 크기가 임계값을 초과할 수 있다. 하지만 백색잡음의 분포 밀집도가 여전히 낮기 때문에 1차원 배열에 대하여 수평/수직 형태학적 필터링을 수행함으로써 배경영역내 존재하는 백색잡음을 제거하고 움직임 영역만을 보다 정확하게 인식할 수 있다. 이는 2차원 영상에 남아있는 백색잡음의 상대적으로 낮은 밀집도가 1차원 배열로의 투영 결과에 그대로 반영되므로 2차원 이전 영상에 대한 Opening 연산의 잡음제거 효과를 1차원 배열에 대한 형태학적 필터링에서도 충분히 얻을 수 있음을 의미한다.

본 논문에서는 앞에서 설명한 내용을 토대로 참고문헌[8]에서의 9×9 마스크를 사용하는 Opening 연산 대신에 움직임 물체의 위치정보인 각 x-축/y-축 1차원 배열에 1×3 마스크와 적절한 임계값을 사용하여 수평/수직으로 형태학적 필터링을 수행하는 동시에 MBB의 추출 연산을 수행함으로써 움직임 물체 추적에 충분한 잡음 제거와 연산량을 감소시키는 결과를 얻었다. 또한 움직임 물체의 최소 경계영역(MBB)의 좌표를 연속적으로 입력되는 영상열로부터 추출하고 이 좌표를 사용하여 움직임 물체의 이동위치 추적을 수행한다. 움직임 물체의 최소 경계영역(MBB)을 추출하고 그 좌표를 계산하여 도식한 결과는 2.3장의 그림7에서 확인할 수 있다.

2.2 실시간 움직임 물체 추적 카메라 데모 시스템

2.1장에서 설명한 움직임 물체 추적 알고리즘은 그림3과 같이 구성된 데모 시스템에서 실시간으로 구현하였다.

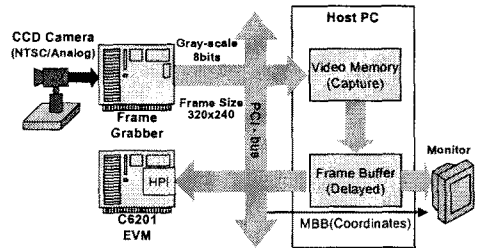


그림3: 자동 감지 / 추적 카메라 데모 시스템 구성도

CCD 감시 카메라로부터 입력되는 NTSC 아날로그 영상신호를 Philips의 비디오 디코더(Video Decoder) SAA7110, PCI Controller SAA7146 등으로 구성된 프레임 그레버(Frame Grabber)를 통해서 YUV 형식의 디지털 영상신호로 획득한다. 이 YUV 형식의 디지털 신호로부터 명암 정보를 갖는 Y 신호만을 취하여, 320×240의 프레임 크기(Frame Size)를 갖는 8bits 그레이(Gray-scaled) 영상신호를 처리하게 된다. 이 디지털 영상신호는 PCI버스(Bus)를 거쳐 호스트 PC의 비디오 메모리(Video Memory)로 전달되며, 이는 다시 TMS320C6201 EVM[11]으로 입력된다. 실시간 구현에 사용한 TMS320C6201 EVM(Evaluation Module)은 Texas Instrument사의 C62x 고정 소수점 DSP 제품군의 하나인 TMS320C6201 DSP를 채용하고 있으며, 외부 메모리와 A/D 기능 그리고 호스트 PC와 PCI 버스를 통한 인터페이스 환경을 갖추고 있는 DSP 전용 개발 보드이다. 그림1에 도시한 움직임 물체 추적 알고리즘은 TMS320C6201 EVM에서 모두 수행되며, 그 결과로 움직임 영역에 대한 최소 경계영역(MBB)의 모서리 위치를 표시하는 두쌍의 좌표값을 호스트 PC로 전달한다. 이때 호스트 PC는 프레임 그레버로부터 획득된 디지털 영상을 모니터상에 보여주는 동시에 MBB에 대한 두쌍의 좌표값으로 디스플레이(Display)되는 영상위에 각각영역을 오버레이(Overlay)시킴으로써 움직임 물체 추적 알고리즘의 수행 결과를 관찰할 수 있도록 한다.

2.3 데모 시스템 수행 결과

움직임 물체 추적 알고리즘(그림1)을 TMS320C6201 EVM에서 실시간으로 구현한 데모 시스템의 수행 결과를 이 장에 기술하였다. 데모 시스템을 통한 실험에서는 영상 프레임율(Frame Rate)을 10 Frame/sec로 설정하였으며, 영상 프레임 크기(Frame Size)가 320×240×8bit인 영상을 사용하였다. 또한, 2.1장에서 기술한 추적 알고리즘을 토대로 한 사람이 건물 안쪽 복도에서 움직이고 있고 카메라는 고정된 상황으로 설정하여 실험을 수행하였다. 그림4~9에 도시된 영상 데이터들은 데모 시스템의 수행을 통해 얻은 움직임 물체 추적 알고리즘(그림1)의 주요 블록에 대한 연산 결과를 도시한 것이다.

그림4는 연속된 두 프레임의 영상에 대해 각 픽셀사이의 차를 구하고 이에 절대값을 취한 후 적절한 임계값으로 이전

영상화합으로써 움직임 에너지를 추출한 결과로 얻는 이진 차영상이다. 이 영상 데이터에서 볼 수 있듯이 영상내에서 일정 방향으로 걷고 있는 한 사람의 움직임을 확인할 수 있다. 즉, 이 이진 차영상을 통해서 물체의 움직임을 감지하게 되며, 움직임영역과 배경영역의 분리(Segmentation)가 이루어지며, 움직임영역을 추출할 수 있다. 그리고 그림5은 연속으로 들어오는 영상열에서 소벨 알고리즘을 사용하여 현재 프레임의 영상으로부터 추출한 에지영상을 도시한 것이다. 에지영상은 우리가 예측하였듯이 움직임 물체의 에지정보를 가지고 있음을 확인할 수 있다. 연속된 프레임의 영상에 대한 그림4의 이진 차영상에는 물체의 현재와 이전 움직임 자취가 모두 나타나게 되는데, 움직임 영역의 정확한 현재 위치를 추출하기 위해 그림5의 에지영상과 그림4의 이진 차영상의 AND 논리연산을 수행한다. 이진 차영상과 에지영상의 AND 논리연산으로 얻은 결과가 그림6의 움직임 에지영상이다. 그림6의 움직임 에지영상에서 물체의 현재 이동위치를 확인할 수 있으므로 움직임 물체 추적을 수행하기 위한 물체의 최소 경계영역(MBB) 추출을 쉽게 할 수 있다. 그림7은 그림6의 움직임 에지영상을 수평/수직방향으로 투영하여 얻는 각 x-축/y-축 1차원 배열에 수평/수직 형태학적 필터링을 수행한 후, 최소 경계영역(MBB)을 추출한 결과로 PC 모니터에 사각영역으로 오버레이(Overlay)시킨 것이다. 그림7을 통해서 영상내에 움직이는 사람을 정확히 감지하고 추적함을 확인할 수 있다.



그림4: 이진 차영상 추출 결과



그림5: 에지영상 추출 결과



그림6: 움직임 에지 추출 결과



그림7: MBB 추출 결과

그림6의 2차원 영상으로부터 수평/수직방향으로 투영하여 1차원 위치정보로 변환한 후, 그 결과에 수평/수직 형태학적 필터링을 취하여 얻은 x-축/y-축 1차원 배열의 투영값을 그림8과 그림9에 도시하였다. 그림8과 9의 각 그래프는 가로축이 픽셀의 위치정보를 나타내며, 세로축이 각 픽셀위치에서의 수평/수직방향으로 투영을 통해 누산된 값을 나타낸다. 특히, 2차원 영상의 x-축상의 1차원 위치정보를 나타내는 그림9의 수직방향 투영 결과로부터 움직임 물체가 영상내의 우측에서 좌측으로 이동하고 있음을 확인할 수 있다. 그림8의 y-축상의 위치정보를 나타내는 수평방향 투영 결과는 실내에서 걷고 있는 사람의 특성상 일정한 길이로 유지됨을 확인할 수 있다. 그리고 우리가 예측하였듯이 그림8과 9의 수평/수직방향 투영 결과에 대한 수평/수직 형태학적 필터링이 배경영역 내에 존재하는 백색잡음을 제거하고 움직임 영역에서 투영된 결과만이 선택되어 정확하게 그림7과 같이 추적 물체의 최소 경계영역(MBB)를 추출할 수 있음을 확인할 수 있다. 그림7에 표시된 움직임 영역의 최소 경계영역(MBB)은 그림8, 9의 수평/수직방향 투영 결과로부터 적절한 임계값을 사용하여 추출한 결과이다.

지금까지 나열한 데모시스템의 수행 결과를 통하여 본 논문에서 기술한 움직임 물체 추적 알고리즘이 자동 감시 카메라 시스템으로 실시간 구현이 가능하며 10Frame/sec의 프레임

임률(Frame Rate)로 성공적으로 수행됨을 확인할 수 있다.

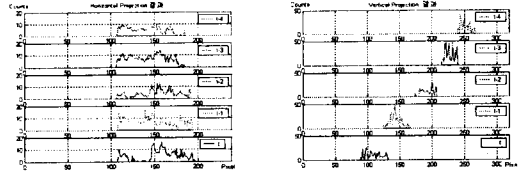


그림8: 수평방향 투영 결과(y-축) 그림9: 수직방향 투영 결과(x-축)

3. 결 론

본 연구에서 움직임 에너지 접근방법을 토대로한 움직임 물체 추적 알고리즘을 제시하고 이를 TMS320C6201 EVM를 이용하여 실시간으로 구현하였다. 주요 영상처리 루틴들은 C6201 컴파일 도구에서 지원되는 Linear Assembly 언어로 작성하여 C 언어로 작성된 부분과 링크(Link)하였다. 이들 언어로 작성된 프로그램은 최적화 과정을 거쳐서 소프트웨어 파이프라인화 되고 병렬처리된다. 본 논문에서는 움직임 물체 추적 알고리즘을 설명하고 실시간으로 구현된 데모 시스템의 수행 결과를 도시하였다. 본 논문에서는 C6201의 내부 데이터 메모리 용량 부족으로 영상 데이터를 외부 메모리로 할당하여 최적의 처리속도는 얻지 못했지만 실시간으로 구현하기에 충분한 10 Frame/sec의 처리속도로 데모 시스템을 구현하고 실험하였다. 데모 시스템의 수행 결과를 통해서 움직임 물체 추적 알고리즘이 성공적으로 실시간 구현됨을 확인할 수 있다. 또한 이 실시간 구현된 데모 시스템을 통해서 다수의 물체의 움직임을 감지하고 추적할 수 있는 시스템으로 확장할 수 있는 가능성을 확인할 수 있다.

(참 고 문 헌)

- [1] G. L. Foresti, "Real-Time Detection of Multiple Moving Objects in Complex Image Sequences", John Wiley & Sons, Inc. 1999
- [2] Young Ho Kim, Kyu Won Lee, Jun Geun Jeon and Kyu Tae Park, "Moving object discrimination and tracking for unmanned surveillance system", SPIE, vol. 2727, pp. 175-185, 1996
- [3] Jakub Segen, Sarma Pingali, "A Camera-Based System for Tracking People in Real Time", IEEE Proceedings of ICPR '96, 1996
- [4] Johnson I Agbinya, David Rees, "Multi-Object Tracking in Video", Real-Time Imaging 5, pp. 295-304, 1999
- [5] Holger Lange "Real Time Motion Detection for Target Acquisition "on the move" based on a Nonlinear Filter using Short Time and Medium Time Image Differences", SPIE Conference on Real-Time Imaging IV, vol. 3645, 1999
- [6] Kyu Won Lee, Jinwoong Kim "Moving Object segmentation based on Statistical Motion Model", Electronics Letters, Volume: 35 Issue: 20, pp. 1719-1720, 30 Sept. 1999
- [7] G. Rigoll, B. Winterstein, S. Muller, "Robust Person Tracking in Real Scenarios with Non-Stationary Background Using a Statistical Computer Vision Approach", Visual Surveillance, 1999. Second IEEE Workshop on, (VS'99), pp. 41-47, 1999
- [8] Don Murray, Anup Basu, "Motion Tracking with an Active Camera", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.16, no.5 may 1994
- [9] Q. Cai, A. Mitiche, and J.K. Aggarwal, "Tracking Human Motion in an Indoor Environment", IEEE, 1995
- [10] Anil K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, 1989
- [11] Nasser Kehtarnava, "DSP System Design : Using The TMS320C6000", Prentice-hall, 2001.