

논문 2004-41SP-5-21

# 이동 카메라 영상에서 히스토그램과 컬러 정보를 이용한 다수 보행자 검출 및 추적

(Multiple Pedestrians Detection and Tracking using Histogram and Color Information from a Moving Camera)

임 종 석\*, 광 현 욱\*\*, 김 욱 현\*

(Jong Seok Lim, Hyun Wook Kwak, and Wook Hyun Kim)

## 요 약

본 논문에서는 이동 카메라로부터 획득한 컬러 영상에서 다수의 보행자를 검출하고 추적하기 위한 히스토그램과 컬러 정보 기반 알고리즘을 제안하였다. 제안한 알고리즘은 RGB 컬러 히스토그램 기법을 이용하여 인접한 보행자를 검출하고 검출된 보행자는 RGB 평균값을 이용하여 추적하였다. 따라서 보행자가 서로 인접해 있거나 약간 겹쳐진 경우에도 검출할 수 있고 보행자의 형태가 변하는 경우에도 효율적으로 추적하였다. 제안된 알고리즘에 대하여 비디오 카메라로 녹화한 영상을 컴퓨터에서 입력받아 보행자 검출과 추적 실험을 수행하여 제안한 알고리즘의 우수성을 입증하였다.

## Abstract

This paper presents a novel histogram and color information based algorithm for detecting and tracking multiple pedestrians from a moving camera. In the proposed method, RGB color histogram is used to detect adjacent pedestrians and RGB mean value is used to track detected pedestrians. Therefore, our algorithm detect contiguous or a few occluded pedestrians and track in case pedestrian's shape change. The experimental results on our test sequences demonstrate the high efficiency of our method.

**Keywords :** pedestrian detection, pedestrian tracking, color histogram

## I. 서 론

동영상에서 이동물체 추적은 보안 또는 감시 시스템, 횡단보도 신호 제어 시스템 등 실세계 다양한 응용 분야에서 사용될 수 있으므로 이것에 대한 연구가 활발히 진행되고 있다<sup>[1, 2, 3]</sup>. 이동물체 추적은 카메라로부터 획득한 영상에서 이동하는 물체를 검출하고, 그것이 어떤 물체인지 인식하여 움직임을 추적하는 것이다. 이동물체의 종류는 사람, 동물, 자동차 등이 있는데, 보통 사람을 추적하는 경우가 대부분이다. 따라서 본 연구에서는 동물이나 자동차가 아닌 걸어다니는 사람 즉, 보행자를 검

출하고 추적하였다.

기존의 이동물체 검출 알고리즘으로는 상관 기반 방법(correlation based method)<sup>[4]</sup>, 광류 기반 방법(optical flow based method)<sup>[5]</sup>, 형태 기반 방법(shape based method)<sup>[6]</sup>, 움직임 기반 방법(motion based method)<sup>[7, 8, 9]</sup>, 차 영상 기반 방법(difference image based method)<sup>[10]</sup> 등이 있고, 추적 알고리즘으로는 영역 기반 방법(region based method)<sup>[11]</sup>, 특징 기반 방법(feature based method)<sup>[12, 13]</sup>, 모델 기반 방법(model based method)<sup>[14, 15]</sup>, 윤곽선 기반 방법(contour based method)<sup>[16]</sup> 등이 있다.

상관 기반 방법은 한 영상의 국소 부분과 다른 영상의 국소 부분을 분리하여 상관계수법에 의해 이동물체를 검출하는 방법으로, 카메라 움직임이 있어도 동작하

\* 정회원, \*\* 학생회원, 영남대학교 컴퓨터공학과  
(Department of Computer Engineering, Yeungnam University)  
접수일자: 2004년4월28일, 수정완료일: 2004년6월16일

지만 물체의 외형이 변하는 유동적인 물체를 검출하기는 어렵다. 광류 기반 방법은 밝기의 국부적인 시공간 경사와 속도 사이의 관계를 모델링한 광류 방정식(optical flow equation)을 사용하는 방법이다. 그러나 물체 표면의 광강도 값은 일정해야 하며, 주위의 인접해 있는 점들은 같은 형태로 움직인다는 속도 평활화 제약(smoothness constraint)조건을 만족할 경우 좋은 결과를 얻지만 프레임간 간격이 넓거나, 조명이 변하거나, 잡음에 의해 일정한 형태를 유지하지 않을 경우, 물체의 형태가 시간이 변함에 따라 변할 경우 등에서는 좋지 못한 결과를 얻을 수 있는 단점이 있다. 형태 기반 방법은 물체의 외관 특징에 의존하는 방법으로, 움직이거나 정지한 물체도 검출할 수 있다. 그러나 물체의 외적인 형태, 유동적인 움직임, 밝기, 색깔 등 광범위한 변화 상태를 고려해야 하는 단점이 있다. 움직임 기반 방법은 영상내의 움직이는 물체의 동작 패턴을 이용하는 방법으로, 대상 물체의 형태나 크기 변화에 관계없이 검출할 수 있고 연산이 간단한 장점이 있다. 그러나 조명이 변하거나 배경의 미세한 움직임에도 검출할 수 없는 단점이 있다. 차 영상 기반 방법은 영상간의 그레이 레벨 차를 이용하는 방법으로 배경이 변하지 않는 경우 계산이 간단하고 다양한 유형의 이동물체를 검출할 수 있다. 영역 기반 방법은 연속 영상에서 특정 영역을 정하고 영역의 상관도를 측정하여 이동물체를 추적하는 방법으로, 현재의 배경을 측정하여 입력되는 영상과의 차 영상을 구하여 이동물체를 검출하여 추적한다. 이 방법은 차 영상에 임계값 이상의 화소를 연결한 영역은 찾을 수 있으나 혼잡한 환경에서 각각의 물체를 분할해야 하는 단점이 있다. 특징 기반 방법은 물체의 전체를 추적하지 않고 이동하는 물체의 특징을 추출한 다음 추적하는 방법으로 부분적인 가려짐이 발생해도 추적이 가능하다. 모델 기반 방법은 미리 정의된 2차원 또는 3차원 모델과 영상을 실시간으로 매칭하는 방법으로, 모델에 의한 이동과 회전이 고려되어야 한다. 또한 매칭 방법과 정의된 모델의 크기 등 정확한 기하학적 모델이 주어져야 하고 추적 모델이 소수로 한정된다는 단점이 있다. 윤곽선 기반 방법은 물체의 경계를 윤곽선으로 표현하고 그것을 동적으로 갱신하면서 추적하는 방법으로, 추적 물체가 부분적으로 가려질 경우 이동물체를 추적하기 어려운 단점이 있다.

보행자를 검출하고 추적하는 기존의 방법들은 고정된 카메라에서 획득한 영상에서 한 명의 보행자를 검출하여 추적하는 것<sup>[12,13,14,15,16]</sup>이 대부분이었지만, 본 논문

에서는 이동 카메라에서 획득한 컬러 영상을 이용하여 다수의 보행자를 검출하고 추적하는 방법을 제안하였다. 보행자를 추적할 때 가장 중요한 단계는 동영상에서 이동물체를 검출하고 보행자로 인식하는 단계이다. 이 과정에서 보행자를 배경으로부터 정확하게 분리해내는 것이 핵심 요소이다. 카메라가 고정되어 있는 경우 배경 영상이 변하지 않으므로 현재의 입력 영상에서 미리 저장된 배경 영상과의 영상 차를 구하는 차 영상 방법이 많이 이용되고 있다. 하지만 카메라가 이동하는 경우 배경 영상이 변하므로 이 방법을 적용할 수 없다. 따라서 배경이 변하는 연속 영상에서 움직임 벡터를 구하여 이동물체를 검출하는 방법을 이용하여야 한다.

이동물체를 검출하고 보행자로 인식되었으면 다음 단계는 추적 단계이다. 자동차와 같이 이동물체의 형태가 단순하고 변화가 없거나 움직임의 범위가 단순한 경우 모델 기반 방법이나 윤곽선 기반 방법 등을 사용할 수 있다. 하지만 보행자와 같이 형태가 비슷하고 변화가 많고, 움직임의 범위가 자유로운 경우 이러한 방법을 적용하여 추적하기는 어렵다. 따라서 본 연구에서는 배경이 변하는 영상에서 다수의 보행자를 검출하기 위해 RGB 컬러 히스토그램 기반 알고리즘을 제안하고, 다수의 보행자를 추적하기 위해 RGB 컬러 평균값을 이용한 추적 알고리즘을 제안한다.

먼저 이동 카메라에서 획득한 컬러 영상으로부터 BMA(Block Matching Algorithm)을 이용하여 특정 영역에 대해 움직임 벡터를 계산한다. 이 움직임 벡터 값을 이용하여 차 영상을 생성하고, 이진 영상으로 변환한 후 불필요한 잡음 등을 제거하면 이동물체가 검출된다. 그리고 검출된 이동물체를 보행자로 인식하고 각각의 보행자로 분리시키기 위하여 프로젝션을 수행한다. 또한 인접한 보행자를 분리시키기 위하여 컬러 히스토그램 기반 알고리즘을 적용한다. 보행자 추적은 검출된 각각의 보행자 주위에 있는 바운딩 박스 영역 내의 RGB 평균값을 이용한다. 그리고 연속적으로 추적을 원활히 하기 위해 추적 영역을 재구성한다.

## II. 본 론

### 1. 제안한 다수 보행자 검출 및 추적 알고리즘 개요

본 논문에서 제안하고 있는 다수 보행자 검출 및 추적 알고리즘의 흐름도는 그림 1과 같다. 비디오 영상에서 연속적인 두 개의 프레임에 입력받아 BMA를 수행하여 움직임 벡터를 검출한다. 이것을 이용하여 차 영

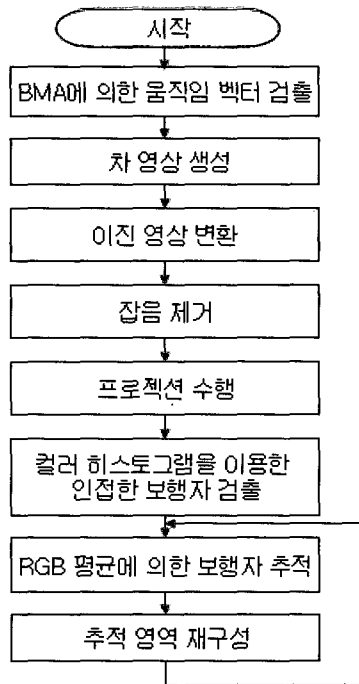


그림 1. 알고리즘 흐름도  
Fig. 1. Algorithm flowchart.

상을 생성하고 이진 영상으로 변환한다. 그리고 불필요한 잡음을 제거한 후 프로젝션을 수행하여 보행자를 검출한다. 프로젝션 방법에서 보행자가 서로 인접한 경우는 검출할 수 없으므로 RGB 컬러 히스토그램 기법을 적용하여 분리시킨다. 보행자 추적은 검출된 각각의 보행자 주위에 있는 바운딩 박스 영역 내의 RGB 평균값을 이용하여 추적한다. 그리고 연속적으로 추적을 원활히 하기 위해 추적 영역을 재구성한다.

2. 제안한 다수 보행자 검출 알고리즘

가. BMA를 이용한 움직임 벡터 검출

카메라가 고정되어 있지 않고 움직이는 경우 배경이 변하므로 기존의 차 영상 방법을 이용하여 이동물체를 검출할 수 없다. 따라서 연속적인 두 프레임간의 움직임 벡터를 검출하고 이것을 이용하여 차 영상을 생성하면 이동물체를 검출할 수 있다.

움직임 벡터는 이전 프레임  $f_{n-1}$ 의 구획을 상하좌우로 움직이고 현재 프레임  $f_n$ 의 차분 총합  $D$ 가 최소로 되는 방향(dx, dy)을 움직임 벡터로 계산하는 BMA를 이용한다. 움직임 벡터를 계산하는 식은 다음과 같다.

$$D = \sum_i \sum_j |f_n(x_i, y_j) - f_{n-1}(x_i + dx, y_j + dy)| \quad (1)$$

식 (1)에서 i, j는 영상의 가로 세로 크기까지 변하는

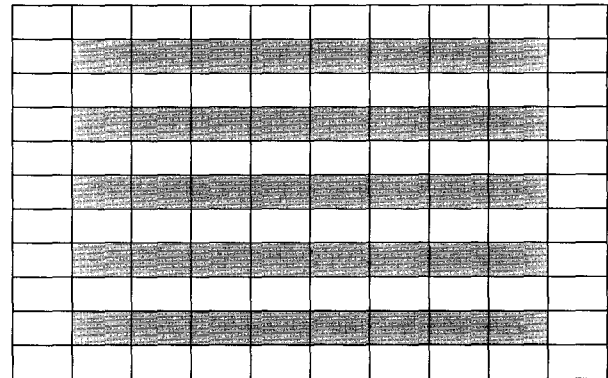


그림 2. 움직임 벡터 계산 영역  
Fig. 2. The computation area of motion vector.



그림 3. 움직임 벡터 영상  
Fig. 3. Motion vector image.

참자이고,  $x_i, y_j$ 는 영상에서 픽셀의 위치를 나타낸다.

움직임 벡터를 계산할 때 각 구획의 크기는 16×16으로 설정하고 구획을 움직이는 최대량을 x방향, y방향 모두 8로 설정한다. 그리고 영상 내의 모든 구획에 대하여 움직임 벡터를 계산하지 않고 그림 2와 같이 음영이 있는 특정 구획에 대해서만 계산한다. 만약 모든 구획에 대하여 움직임 벡터를 계산하면 처리시간이 많이 걸리고, 더 작은 구획에 대하여 계산하면 처리시간은 작아지지만 카메라 이동이 큰 경우 움직임 객체를 검출하는데 나쁜 영향을 미치게 된다. 그래서 적당한 구획에 대하여 계산하면 처리시간도 줄일 수 있고 검출 결과도 모든 구획에서 계산한 것과 별 차이가 없음을 실험을 통하여 알 수 있었다.

움직임 벡터가 구해지면 이동량을 찾기 위해 대표 움직임 벡터를 구한다. 대표 움직임 벡터는 계산된 움직임 벡터 중에서 가장 빈번하게 발생한 움직임 벡터로 설정한다.

그림 3은 식 (1)을 적용하여 검출한 움직임 벡터 영상을 보여주고 있다. 그리고 여기서 대표 움직임 벡터는  $x=4, y=0$ 이다. 이것은 카메라가 x방향으로 4픽셀만큼, y방향으로 0픽셀만큼 이동하였음을 의미한다.

나. 차 영상 생성

기존의 차 영상 방법은 연속적인 두 프레임의 그레이 레벨 차를 이용하여 영상 내에 이동하는 물체를 검출하는 방법이다. 이 방법은 배경이 변하지 않는 영상일 경우 잘 동작하지만 카메라 이동에 의한 배경 화면이 변할 경우 이동물체를 검출하지 못하는 단점이 있다. 따라서 본 논문에서는 움직임 벡터를 이용하여 카메라 이동량만큼 보정함으로써 카메라 이동에 의한 배경 화면이 변해도 이동물체를 검출할 수 있다.

일반적으로 차 영상을 구하는 식은 다음과 같다.

$$d(i, j) = |f_n(i, j) - f_{n-1}(i, j)| \tag{2}$$

한편, 움직임 벡터를 이용하여 움직임 보정을 통한 차 영상을 구하는 식은 다음과 같다.

$$d(i, j) = |f_n(i + x, j + y) - f_{n-1}(i, j)| \tag{3}$$

식 (3)에서 x, y는 움직임 벡터 성분이다.

그림 4는 대표 움직임 벡터를 이용하여 움직임 보정을 수행한 후의 차 영상과 움직임 보정 없이 수행한 차 영상을 비교한 것이다. 움직임 보정한 차 영상 (a)가 그렇지 않은 (b)보다 배경이 더 깨끗함을 알 수 있다.

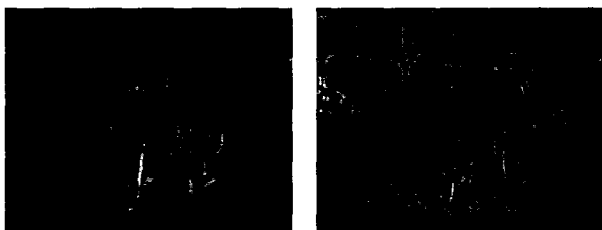
다. 전처리 과정

차 영상 방법에 의해 이동물체가 검출되면 보행자인지 아닌지 인식하기 위한 과정이 필요하다. 따라서 전처리 과정을 통하여 차 영상을 이진 영상으로 변환하고 불필요한 잡음 등을 제거한다.

먼저 이진 영상은 차 영상에서 임계값 보다 큰 픽셀은 255로 설정하고 그렇지 않으면 0으로 설정한다. 차 영상을 이진 영상으로 변환하는 식은 다음과 같다.

$$B(i, j) = \begin{cases} 255 & \text{If } d(i, j) \geq Th \\ 0 & \text{Otherwise} \end{cases} \tag{4}$$

식 (4)에서 B(i,j)는 이진 영상, d(i,j)는 차 영상을 나타내고 Th는 임계값을 의미한다.



(a) 움직임 보정한 차 영상 (b) 움직임 보정없는 차 영상

그림 4. 차 영상 비교

Fig. 4. The comparison of the difference image.

변환된 이진 영상에는 불필요한 잡음 등이 많이 존재하므로 이를 제거해야한다. 잡음 제거에는 여러 가지 방법이 존재하는데 본 논문에서는 영역 제거 방법을 제안한다. 영역 제거 방법은 n×n크기의 슬라이딩 윈도우를 만들어 이동하면서 윈도우 영역 내에 존재하는 픽셀 개수를 카운트하여 임계값 보다 작으면 잡음으로 간주하여 윈도우 영역 내의 모든 픽셀들을 제거하는 방법이다.

본 논문에서는 슬라이딩 윈도우의 크기를 그림 5와 같이 가로 25, 세로 30으로 설정하고, 윈도우 내에 내접하는 원을 그렸을 때 원둘레의 크기에 해당하는 값인 72를 임계값으로 사용하였다. 이 슬라이딩 윈도우에 내접하는 원은 검출하고자 하는 보행자의 얼굴에 해당하는 크기이다.

그림 6은 차 영상을 이진 영상으로 변환한 영상과 잡음을 제거한 영상을 나타내고 있다. (a)는 차 영상에 임계값 60을 사용하여 변환한 이진 영상이고 (b)는 (a)에 그림 5를 적용하여 잡음을 제거한 영상이다.

라. 프로젝션 수행

전처리 과정이 끝나면 보행자 검출을 위하여 프로젝션을 수행한다. 프로젝션은 수평 방향과 수직 방향으로 수행하면서 0보다 큰 픽셀 값의 개수를 카운트하여 히스토그램을 만든다. 다음 식 (5)와 식 (6)은 수평 방향과 수직 방향에 대해 프로젝션 히스토그램을 생성하는 식이다.

$$H_i = \sum_{j=0}^{n-1} B(i, j) \tag{5}$$

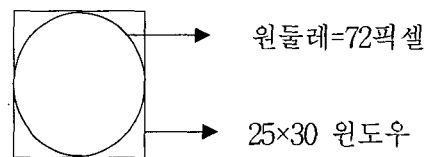


그림 5. 슬라이딩 윈도우와 임계값

Fig. 5. The sliding window and threshold value.



(a) 이진 영상

(b) 잡음 제거 영상

그림 6. 이진 영상과 잡음 제거 영상

Fig. 6. The binary image and noise deleted image.

$$V_j = \sum_{i=0}^{m-1} B(i, j) \quad (6)$$

식 (5)와 (6)에서 B(i,j)는 이진 영상, i는 0에서 m-1, j는 0에서 n-1까지이고, m은 영상의 세로 크기, n은 영상의 가로 크기이다.

프로젝션 히스토그램이 생성되면 인간의 체형 정보를 바탕으로 보행자를 검출한다. 보행자는 자동차나 동물 등과는 다른 형태를 가지고 있으므로 쉽게 인식할 수 있다. 그러나 여러 명의 보행자가 서로 인접해 있거나 약간 겹쳐져 있는 경우, 또는 카메라 이동이 심하여 배경이 변하는 경우 보행자를 검출하지 못하는 단점이 있다.

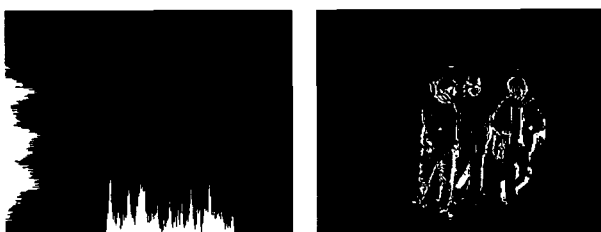
그림 7은 프로젝트션 히스토그램과 이것을 이용하여 보행자를 검출한 결과를 나타내고 있다. (a)는 수평 방향과 수직 방향 프로젝트션 히스토그램을 나타낸 것이고, (b)는 보행자 검출 결과를 바운딩 박스로 나타낸 것이다. 그러나 보행자들이 서로 인접하여 있으므로 보행자 검출에 실패하였음을 알 수 있다.

마. 영상처리에서 사용되는 컬러 모델

일반적으로 영상처리에서 사용되고 있는 컬러 모델로 RGB 컬러 모델, HSI 컬러 모델 그리고 YCbCr 컬러 모델 등이 있다<sup>[17,18]</sup>. 이들 컬러 모델들은 컬러들과 다른 컬러들과의 관계를 표현하는 방법이다. 다양한 영상처리 시스템은 적용분야에 따라 서로 다른 컬러 모델을 사용한다.

(1) RGB 컬러 모델

RGB 컬러 모델은 서로 가산될 수 있는 삼원색인 빨간색, 초록색, 파란색으로 구성된다. 이들 컬러의 분광 요소들이 부가적으로 복합되어 결과적인 컬러를 만든다. 컬러 채널당 8비트를 가지는 24비트 컬러 그래픽스 시스템에서 거의 자연색상 그대로 표현한다.



(a) 프로젝트션 히스토그램 (b) 보행자 검출 영상  
 그림 7. 프로젝트션 히스토그램과 보행자 검출 영상  
 Fig. 7. The projection histogram and pedestrians detected image.

RGB 컬러 모델은 컴퓨터 그래픽스 시스템의 설계를 간단하게 하지만 컬러 요소들 간의 상호관계가 매우 강하여 몇몇 영상처리 알고리즘의 수행에 어려움이 있다.

(2) HSI 컬러 모델

색상(hue), 채도(saturation) 그리고 명도(intensity)라는 세 가지 성분을 이용하는 컬러 모델이다. HSI 컬러 모델에서 어떤 컬러를 만들어 내기 위하여 몇 퍼센트의 파란색이나 초록색이 필요한지 알 필요가 없다. 진한 빨간색을 분홍색으로 바꾸기 위해 단순히 채도만을 조절하고, 어두운 색을 밝은 색으로 바꾸려면 명도를 변경한다.

HSI 컬러 모델은 인간이 컬러를 인식하는 방법과 동일한 방법으로 컬러 특징들을 다룰 수 있으므로 컴퓨터 비전 등 많은 영상처리 응용분야에서 서로 다른 물체들의 컬러를 식별하는데 사용된다.

(3) YCbCr 컬러 모델

아날로그 신호인 YUV 컬러 모델을 컴퓨터가 처리할 수 있도록 디지털로 변환한 것이다. YUV 컬러 모델은 인간의 시각 특성상 색차신호보다 밝기신호에 더 민감하기 때문에 밝기신호와 색차신호를 분리하면 영상을 압축할 때 더 편하기 때문에 사용된다.

YCbCr 컬러 모델에서 밝기는 Y로 기호화되고 푸른 색차정보와 붉은 색차정보는 Cb와 Cr로 기호화된다.

본 논문에서는 여러 가지 컬러 모델 중에서 RGB 컬러 모델을 사용하였다. RGB 컬러 모델은 다른 모델에 비해 색상 정보를 많이 가지고 있으므로, 이 색상 정보를 이용하여 보행자를 분리시키고 추적하고자 하는 것이다. 또한 카메라로부터 얻어지는 대부분의 비디오 영상들은 RGB 컬러 모델로 저장되므로 다른 컬러 모델로 변환 과정 없이 그대로 사용할 수 있어 처리 시간을 절약할 수 있기 때문이다.

바. 컬러 히스토그램을 이용한 인접 보행자 검출

프로젝션 방법으로 보행자 검출이 불가능할 경우 RGB 컬러 정보를 기반으로 한 컬러 히스토그램 기법을 이용하여 보행자를 검출한다. 프로젝트션 히스토그램에 의해 보행자가 존재하는 전체 영역은 알 수 있으므로 이 영역에서 보행자 머리부분이 있는 영역에 대해서만 컬러 히스토그램을 작성한다. 그 이유는 머리를 구분하면 보행자를 검출한 것과 동일하기 때문이다. 본 논문에서 검출하고자 하는 보행자의 머리 부분은 대략 세로

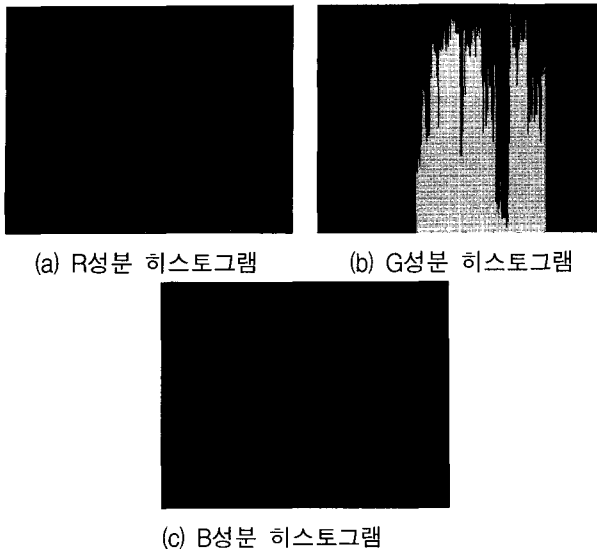


그림 8. RGB 성분 히스토그램  
Fig. 8. The histogram of R, G and B component.

로 30픽셀 정도이다.

다음은 현재 프레임에서 보행자가 존재하는 영역에 대해 RGB 컬러를 이용하여 히스토그램을 작성하는 방법이다.

- ① 보행자가 존재하는 영역의 첫 번째 라인 상에 있는 각 픽셀의 RGB성분을 히스토그램으로 각각 나타낸다.
- ② 두 번째 라인부터 30라인까지 ①번 과정을 반복한다.

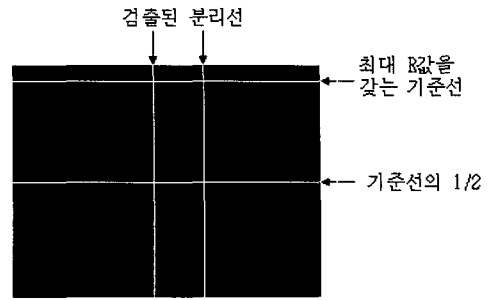
그림 8은 보행자 머리부분 영역에 대한 RGB성분 히스토그램을 각각 나타낸 것이다.

그림 8을 보면 R성분 히스토그램이 G성분이나 B성분 히스토그램보다 명확하므로 보행자를 분리시키는데 R성분 히스토그램만으로도 충분함을 알 수 있다.

히스토그램을 이용하여 보행자를 분리시키기 위한 방법은 다음과 같다.

- ① R성분 히스토그램에서 가장 큰 값을 갖는 위치를 찾는다.
- ② ①에서 찾은 위치를 기준으로 히스토그램의 1/2되는 위치를 구한다.
- ③ ②에서 찾은 위치에서부터 가로 방향으로 진행하면서 R성분이 0인 부분을 카운트한다.
- ④ ③에서 구한 값이 4이상이면 구한 값의 1/2되는 위치를 분리점으로 설정한다.

그림 9(a)는 위의 방법으로 검출한 보행자의 분리점을 세로선으로 나타낸 것이다. 그림 9(b)는 그림 7(b)의



(a) 검출한 보행자 분리선



(b) 보행자 검출 영상

그림 9. 분리선과 보행자 검출 영상

Fig. 9. The separate line and pedestrians detected image.

보행자 검출 영상과 보행자 분리선을 이용하여 바운딩 박스를 만들고 보행자의 상하좌우 공백을 제거하여 최종적으로 인접한 보행자를 분리시킨 영상을 나타내고 있다.

### 3. 다수 보행자 추적 및 추적 영역 재구성

#### 가. 다수 보행자 추적

보행자 검출이 완료되면 검출된 보행자를 RGB 컬러 정보를 이용하여 추적한다. 검출된 다수의 보행자 중에서 가장 왼쪽에 있는 보행자를 먼저 추적하고 계속해서 오른쪽의 보행자를 추적한다. 보행자를 추적할 때 바운딩 박스 내에 있는 RGB 컬러의 평균값을 다음 프레임에서 가장 잘 일치되는 곳을 찾는다. 이때 탐색 영역은 현재 프레임의 바운딩 박스 위치에서 상하좌우 5픽셀 범위를 대상으로 한다. 따라서 현재 프레임의 탐색 대상 바운딩 박스 내의 RGB 컬러 평균값과 다음 프레임의 탐색 영역 내의 RGB 컬러 평균값의 차를 구하여 최소가 되는 영역을 기준으로 바운딩 박스를 생성한다.

그림 10은 추적 시 사용하는 탐색 대상과 탐색 영역을 나타내고 있다.

#### 나. 추적 영역 재구성

보행자는 걸어갈 때 팔을 흔들거나 다리를 내딛는 등 다양한 형태 변화를 보인다. 따라서 이러한 형태 변화에 적응하면서 강건하게 추적할 수 있도록 추적 영역을

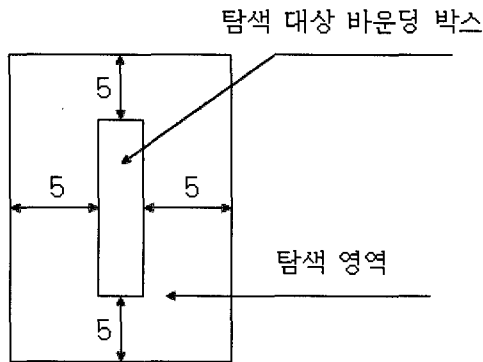


그림 10. 탐색 대상과 탐색 영역  
Fig. 10. The search target and area.



그림 11. 세그먼트 영상  
Fig. 11. The segment image.

재구성해야 한다. 본 논문에서는 보행자의 위치를 추적한 후 추적 영역을 수정하여 다음 추적에 사용하였다.

새로운 추적 영역은 식 (7)을 만족하는 점들이 추적한 바운딩 박스의 경계를 벗어날 때 그 점들이 연속된 점일 경우 포함시켜 구성하였다. 이때 바운딩 박스의 상단과 하단만을 고려한다.

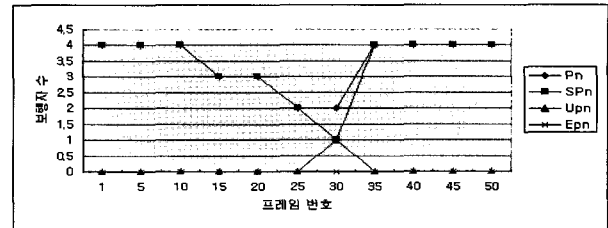
$$I_{RCB}(i,j) = \left\{ \begin{array}{l} \text{If } I_R(i,j) < 90 \quad \text{AND} \\ |I_R(i,j) - I_G(i,j)| < 70 \quad \text{Then } I_{RCB}(i,j) \\ \text{Otherwise} \quad \quad \quad 255 \end{array} \right\} \quad (7)$$

여기서  $I_{RGB}(i,j)$ 는 현재 프레임의 RGB 컬러 영상을 나타낸다.

그림 11은 식 (7)을 이용한 세그먼트 영상을 나타낸 것이다.

### III. 실험

본 논문에서 사용된 실험 영상은 실세계에 존재하는 도시 컬러 영상을 이용하였다. 비디오 카메라를 이용하여 촬영한 동영상을 320×240으로 프레임 단위로 캡처하여 사용하였다. 이 영상을 Pentium IV 1.4 Ghz 프로



(a) 검출 결과 그래프



(b) Frame 5

그림 12. 보통 보행자 검출 결과  
Fig. 12. The result of regular pedestrian detection.

세서의 윈도우즈2000 프로페셔널 버전 운영체제에서 Visual C++ 6.0을 이용하여 구현하였다.

실험은 보행자 검출과 보행자 추적의 2단계로 진행하였다. 보행자 검출 실험에 사용한 영상의 프레임 수는 총200여개로 프레임 당 보행자 수는 2~4명이었고 보통 보행자, 느린 보행자, 빠른 보행자, 뛰는 보행자와 같은 4종류의 영상을 가지고 실험하였다. 그리고 보행자 검출 알고리즘의 성능을 평가하기 위해 식 (8), (9), (10)과 같은 검출 성공률, 검출 오류률, 검출 실패율을 사용하였다.

$$\sum_{\text{전체프레임}} \frac{\text{검출한 보행자수}}{\text{프레임 당 보행자수}} \times 100\% \quad (8)$$

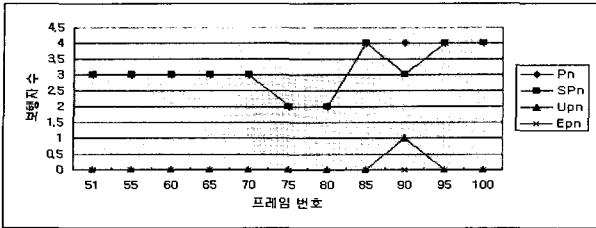
$$\sum_{\text{전체프레임}} \frac{\text{검출 못한 보행자수}}{\text{프레임 당 보행자수}} \times 100\% \quad (9)$$

$$\sum_{\text{전체프레임}} \frac{\text{잘못 검출한 보행자수}}{\text{프레임 당 보행자수}} \times 100\% \quad (10)$$

실험에서 보행자가 서로 겹쳐진 경우 보행자의 머리를 구분할 수 있는 경우에만 두 명의 보행자로 판단하고 그렇지 않은 경우는 한 명의 보행자로 판단한다.

제안한 방법으로 실험한 결과 검출 성공률은 97%이고 검출 오류률은 0%이며, 검출 실패율은 3%로 아주 우수한 성능을 보였다. 검출에 실패한 3개의 프레임은 두 명의 보행자가 너무 인접해 있어 한 명의 보행자로 검출했기 때문이다.

그림 12에서 그림 15까지는 보행자 검출 결과를 나타낸 것이다. 여기서 Pn은 보행자수, SPn은 검출 성공한 보행자 수, UPn은 검출하지 못한 보행자 수, Epn은 잘

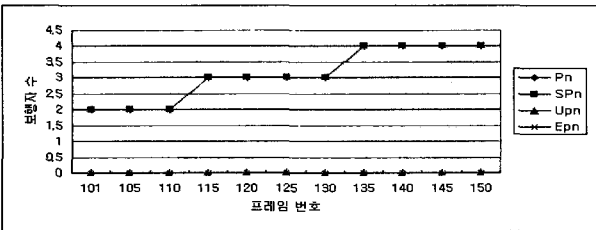


(a) 검출 결과 그래프



(b) Frame 65

그림 13. 느린 보행자 검출 결과  
Fig. 13. The result of slow pedestrian detection.



(a) 검출 결과 그래프



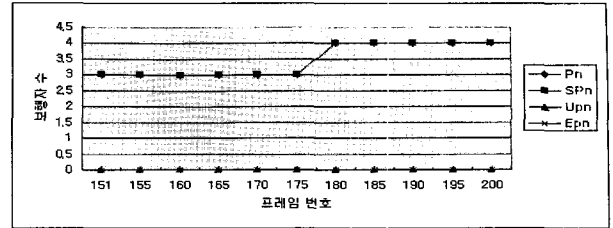
(b) Frame 105

그림 14. 빠른 보행자 검출 결과  
Fig. 14. The result of fast pedestrian detection.

못 검출한 보행자 수를 나타낸다.

보행자 추적 실험에 사용한 영상은 초당 15FPS의 프레임율로 분할한 200여개의 프레임을 사용하여 실험한 결과 프레임당 처리시간이 평균 65.2ms였다. 그리고 추적 정확도는 완전 탐색 기법에 의하여 추적한 위치를 최적의 위치로 가정하고 제안한 방법으로 추적하였을 때의 위치 사이의 오차  $e$ 를 계산하여 평가하였다. 오차  $e$ 는 완전 탐색 기법을 적용하여 구한 보행자 바운딩 박스의 왼쪽 위 모서리의 좌표를  $(x,y)$ 라 하고 제안한 방법에 의하여 구한 대응하는 좌표를  $(x',y')$ 라고 하면 다음 식 (11)과 같이 정의한다.

$$e = \sqrt{(x-x')^2 + (y-y')^2} \quad (11)$$



(a) 검출 결과 그래프



(b) Frame 195

그림 15. 뛰는 보행자 검출 결과  
Fig. 15. The result of running pedestrian detection.

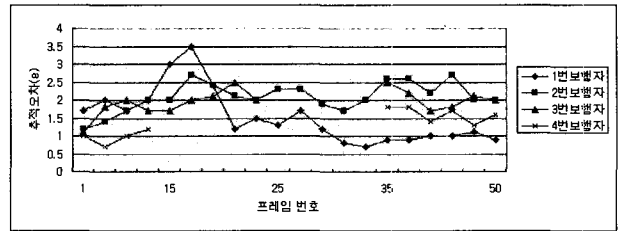


그림 16. 보통 보행자 추적 오차 그래프(평균오차 1.66)  
Fig. 16. The tracking error graph of regular pedestrian.

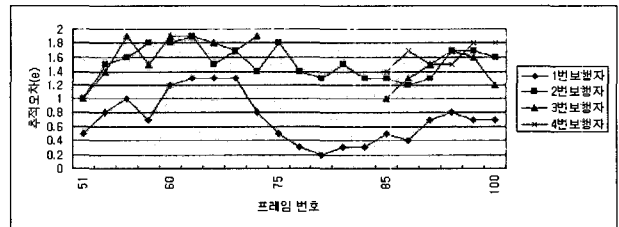


그림 17. 느린 보행자 추적 오차 그래프(평균오차 1.27)  
Fig. 17. The tracking error graph of slow pedestrian.

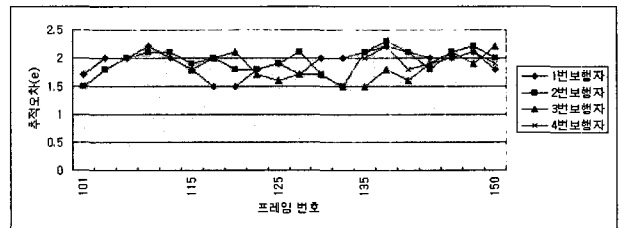


그림 18. 빠른 보행자 추적 오차 그래프(평균오차 1.90)  
Fig. 18. The tracking error graph of fast pedestrian.

그리고 기존의 보행자 추적 방법의 하나인 Hausdorff정합방법<sup>[19]</sup>과 비교를 통하여 제안한 방법의 우수성을 입증하였다.

그림 16에서 그림19까지는 각각 보통 보행자, 느린 보행자, 빠른 보행자, 뛰는 보행자 추적 결과를 그래프



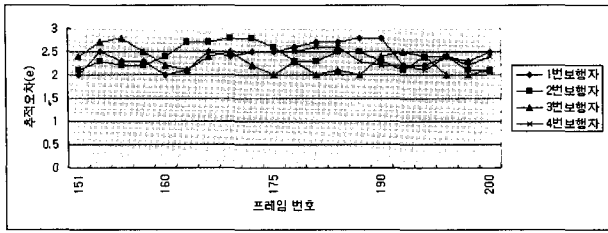


그림 19. 뛰는 보행자 추적 오차 그래프(평균오차 2.36)  
 Fig. 19. The tracking error graph of running pedestrian.

표 1. 추적 방법 비교

Table 1. The comparison of tracking method.

	프레임수	프레임율	처리시간	추적오차
Hausdorff방법	100	28fps	35.7ms	1.66
제안한 방법	200	15fps	65.2ms	1.79

르 나타낸 것이다. 그림 16에서 오차 값이 크게 나타난 곳은 카메라 이동이 크게 발생한 프레임을 나타낸다.

표 1은 기존의 보행자를 추적하는 방법과 제안한 방법을 비교한 결과 제안한 방법이 우수함을 보여주었다.

#### IV. 결 론

본 논문에서는 이동 카메라에서 획득한 컬러 영상에서 다수의 보행자를 검출하고 추적하는 RGB 컬러 기반 알고리즘을 제안하였다. 제안한 알고리즘은 보행자들이 서로 인접해있거나 카메라 이동에 의한 배경 화면이 변해도 보행자를 검출하고 다양한 형태의 보행자를 강건하게 추적하였다. 실험은 움직임이 있는 카메라로부터 획득한 컬러 영상을 입력으로 하여 보행자 검출과 보행자 추적 2단계로 실시하였다. 보행자 검출은 보통 보행자, 느린 보행자, 빠른 보행자, 뛰는 보행자 등 4종류의 영상 200여 개의 프레임을 가지고 실험하여 검출 성공률 97%, 검출 오류율 0%, 검출 실패율 3%의 우수한 성능을 보였다. 그리고 보행자 추적 또한 전체 평균 오차 1.79의 매우 높은 추적 정확도를 보였다.

#### 참 고 문 헌

[1] C. Wohler, J. K. Aulaf and T. Portner, U. Franke, "A Time Delay Neural Network Algorithm for Real-time Pedestrian Detection," *Proc. of the IEEE Intelligent Vehicles Symposium'98, (stuttgart, Germany)*, pp. 247-251, Oct. 1998.  
 [2] M. yachida, M. Asada and S. Tsuji, "Automatic Analysis of Moving Image," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-3, No. 1, pp. 12-20, 1981.

[3] J. Agbinya and D. Rees, "Multi-object Tracking in Video," *Real-Time Imaging* 5, pp. 295-304, 1999.  
 [4] H. Inoue, T. Tachikawa and M. Inaba, "Robot Vision System with a Correlation Chip for Real-time Tracking, Optical flow and Depth Map Generation," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 1621-1626, 1992.  
 [5] S. Yamamoto, Y. Mae and Y. Shirai "Real-time Multiple Object Tracking based on Optical Flows," *Proc. of the Robotics and Automation*, Vol. 3, pp. 2328-2333, 1995.  
 [6] A. Broggi, M. Bertozzi and A. Fascioli, "Shape-based Pedestrian Detection," *Proc. of the IEEE Intelligent Vehicles Symposium 2000*, pp. 215-220, 2000.  
 [7] H. Mori, N. M. Charkari and T. Matsushita, "On-Line Vehicle and Pedestrian Detection based on Sign Pattern," *IEEE Trans. on Industrial Electronics*, Vol. 41, No. 4, pp.384-391, 1994.  
 [8] S. A. Niyogi and E. H. Adelson, "Analyzing and Recognizing Walking Figures in xyt," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 469-474, 1994.  
 [9] S. A. Niyogi and E. H. Adelson, "Analyzing Gait with Spatiotemporal Surfaces," *IEEE Workshop on Motion of Non-Rigid and Articulated Objects*, pp. 64-69, 1994.  
 [10] J. S. Lim and W. H. Kim, "Multiple Pedestrians Tracking Using Difference Image and Projection Histogram," *Proc. of the International Conf. on Imaging Science, Systems and Technology (CISST'02)*, Vol. 1, pp. 329-334, 2002.  
 [11] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving Target Classification and Tracking from Real-Time Video," *Workshop on Application of Computer Vision*, Princeton, 1998.  
 [12] T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," *International Journal of Computer Vision*, Vol. 37, No. 2, pp. 175-185, 2000.  
 [13] 이규원, 김학수, 전준근, 박규태, "특징 기반 움직임 플로우를 이용한 이동 물체의 검출 및 추적," *한국통신학회논문지*, 제23권 제8호, 1998-2009쪽, 1998년 8월.  
 [14] K. Rohr, "Towards Model-based Recognition of Human Movements in Image Sequences," *CVGIP:Image Understanding*, Vol. 59, No. 1, pp. 94-115, 1994.  
 [15] 장대식, 김계영, 최형일, "모델에 기반한 이동 물

체의 추적,” *정보과학회논문지(B)*, 제23권 제12호, 1298-1308쪽, 1996년 12월.

[16] N. Paragios and R. Deriche, "Geodesic Active Contours and Level Sets for the Detection and Tracking of Moving Objects," *IEEE Trans. on PAMI*, Vol. 22, No. 3, pp. 266-280, 2000.

[17] H. R. Myler and A. R. Weeks, "The Pocket Handbook of Image Processing Algorithms in C," *Prentice Hall PTR*, 1993.

[18] 이호석, 김준기, "알기쉬운 MPEG-2," *홍릉과학출판사*, 2001.

[19] 전춘, 이주신, "이동 카메라 영상에서 이동물체의 실시간 추적," *정보처리학회논문지B*, 제9-B권 제4호, 465-470쪽, 2002년 8월.

저 자 소 개



**임 종 석**(정회원)  
 1991년 계명대학교 물리학과 학사 졸업.  
 1998년 대구가톨릭대학교 전산통계학과 석사 졸업.  
 2004년 영남대학교 컴퓨터공학과 박사 졸업.

<주관심분야: 영상처리, 패턴인식, 컴퓨터비전>



**곽 현 옥**(학생회원)  
 2000년 영남대학교 컴퓨터공학과 학사 졸업.  
 2002년 영남대학교 컴퓨터공학과 석사 졸업.  
 2004년 영남대학교 컴퓨터공학과 박사 수료.

<주관심분야: 시각정보처리, 패턴인식, 컴퓨터비전>



**김 옥 현**(정회원)  
 1981년 경북대학교 전자공학과 학사 졸업.  
 1983년 경북대학교 컴퓨터공학과 석사 졸업.  
 1993년 일본 쓰쿠바 대학 공학 연구과 박사 졸업.

1983~1993년 한국전자통신 연구원 선임연구원  
 1994년~현재 영남대학교 전자정보공학부 교수  
 <주관심분야: 시각정보처리, 패턴인식, 화상처리>