한국지능시스템학회 논문지 2010, Vol. 20, No. 5, pp. 631-637

교통신호제어를 위한 CUDA기반 보행자 행동판단

Pedestrians Action Interpretation based on CUDA for Traffic Signal Control

이홍창 · 이상용 · 김영백

Hong-Chang Lee, Sang-Yong Rhee and Young-Baek Kim

경남대학교 컴퓨터공학부

요 약

본 연구에서는 교통 신호를 능동적으로 제어하기 위하여 횡단보도영역에서 보행자의 행동을 판단하는 방법을 제안한다. 코드북기법을 이용하여 보행자 객체를 검출하고, 외곽선을 정보를 획득한다. 신속한 객체 검출을 위하여 CUDA(Compute Unified Device Architecture)기반 병렬화 처리한다. 해당 객체의 형상정보에 왜곡을 일으키는 투영 음영을 제거한 후, 보행자 객체가 보행자인지 혹은 차량, 동물인지를 식별하기 위해 힐버트 스캔 거리값(Hilbert Scan Distance)을 이용한 형판정합 기법을 수행한다. 정합 후에는 보행자 객체의 움직임, 얼굴영역의 특징, 대기 시간의 분석을 통하여 보행자의 횡단보도 이용 의지를 판단하고 교통신호를 제어한다.

키워드 : 행동판단, 보행자 추출, 힐버트 스캔, CUDA, 교통신호제어.

Abstract

In this paper, We propose a method of motion interpretation of pedestrian for active traffic signal control. We detect pedestrian object in a movie of crosswalk area by using the code book method and acquire contour information. To do this stage fast, we use parallel processing based on CUDA (Compute Unified Device Architecture). And we remove shadow which causes shape distortion of objects. Shadow removed object is judged by using the hilbert scan distance whether to human or noise. If the objects are judged as a human, we analyze pedestrian objects' motion, face area feature, waiting time to decide that they have intetion to across a crosswalk for pdestrians. Traffic signal can be controlled after judgement.

Key Words : Action Interpretation, Pedestrian Detection, Hilbert Distance, CUDA, Traffic Signal Control

1. 서 론

우리나라의 현 교통 신호등 시스템은 중앙통제실에서 적절한 시간간격을 주어 신호등의 신호를 변환시키는 것을 기본으로 한다. 이처럼 시간간격을 정해서 신호를 변환시키는 시스템은 교통량과 보행자의 통행량이 많은 대도시의 경우에는 효율적이다. 그러나 통행량이 적은 일반국도나 소도시의 일반도로의 경우에는 보행자가 없음에도 불구하고 현 신호등 시스템에 따라 불필요한 차량 대기시간으로 인해 차량 에너지소모의 증가, 운전자의 불법운행, 보행자의 무단횡단 등 경제적, 안정적인 측면에서 비효율적인 면이 발생한다. 따라서 현 신호등 시스템에 컴퓨터비전기술을 도입하여 비효율적인 점들을 개선하고 더 효율적으로 보행자의 안전과 원활한 차량흐름을 유도하기위해 교통신호를 상황에 맞게 제어할 필요성이 있다.

현행 교통 신호제어가 고정된 신호 주기에 따라 변하도록 되어 있다는 단점을 보완하기 위해 윤재홍 등[1]은 카메라 영상에서 모폴로지 연산을 통해 객체를 추출하고 추출된 객체의 위치에 따라 객체를 구분하여 적응적으로 교통신호를 제어하는 방법을 제안하였다. 하지만 검출된 객체가 단순히 노이즈에 의한 것인지 보행자인지 차량인지에 대한 검증이 이루어지지 않았다는 단점이 있다. 그리고 김철기 등[2]은 횡단보도 신호제어를 위해 차영상을 이용하여 객체를 검출하고 ART2 알고리즘을 적용하여 검출된 객체를 추적함으로써 보다 효율적으로 횡단보도 신호를 제어하고자 하였다. 하지만 이 역시도 검출된 객체가 보행자가 아닌 동물이나 자동차인 경우에 대비하지 못했으며, 배경 이미지에 대한 갱신 시간이 짧은 경우 추적율이 현저히 떨어졌다.

본 논문에서는 카메라 영상에서 객체를 검출하기 위해 코드북 알고리즘[3,4,5]을 이용하여 배경 이미지를 만듬으로써 객체 검출 시 배경 이미지에 따른 영향을 최소화 하였고, 힐버트 스캔 거리값을 이용한 템플릿 매칭 기법을 이용하여 검출 된 객체에 대한 보행자 검증을 수행함으로써 동물이나 자동차에 의한 오작동을 막는다. 또한 보행자가 단순히 지나가는 보행자인지 횡단보도를 건너고자 하는 보행자인지를 분석하여 보다 효율적인 신호 제어가 가능하도록 한다. 하지만 이러한 기법들은 많은 연산량을 요구하여 실시간 처리를 어렵게 만든다. 하지만 최근 GPU는 다수의 프로세서

접수일자 : 2010년 4월 3일
완료일자 : 2010년 10월 1일
+ 교신저자
*본 연구는 2009년도 경남대학교 학술연구장려금 지원으로 이루어졌음

를 이용한 병렬처리를 수행함으로써 영상처리와 같이 연산량은 많으나 종속도가 낮은 경우 월등히 빠른 처리속도를 보여주므로 제안하는 방법을 실시간으로 처리 할 수 있다. 따라서 기존의 CPU가 아닌 GPU를 이용한 CUDA 프로그래밍 방법을 사용하여 실시간 처리가 가능하도록 한다.

2. 본론

2.1 시스템 개요

제안하는 방법은 먼저, 횡단보도 영역에서 촬영된 동영상으로부터 영상을 입력받고 일정 프레임을 입력받는 동안 코드북 형태로 배경을 모델링한다. 그 후 입력된 프레임과 모델링된 코드북의 비교를 통하여 보행자 객체를 검출한다. 이 과정에서는 모델링된 코드북과 입력된 프레임의 전체 정보에 접근해야 하므로 효율적인 처리속도를 위해 CUDA를 이용하여 병렬처리로 수행한다.

검출된 객체들은 태양과 같은 조명의 위치에 따라 투영 음영이 나타나는 경우가 발생한다. 이렇게 투영 음영이 포함되어 있으면 객체 검출 시 외형적인 왜곡을 일으키기 때문에 투영 음영을 제거한다. 그리고 검출된 객체가 반드시 보행자라고 판단하는 것은 합리적이지 않기 때문에, 보행자 객체를 식별하기 위해 힐버트 스캔 거리값을 이용하여 검증한다.

이러한 검증과정을 통해 보행자로 식별된 객체가 존재한다면 얼굴영역을 검출하여 특징을 분석하고, 움직임 여부, 대기시간을 통하여 보행자의 횡단보도 이용의지를 판단한다. 마지막으로 보행자의 행동판단 결과에 따라 교통신호등을 제어한다. 그림1은 제안하는 시스템의 흐름도를 도식화한 것이다.

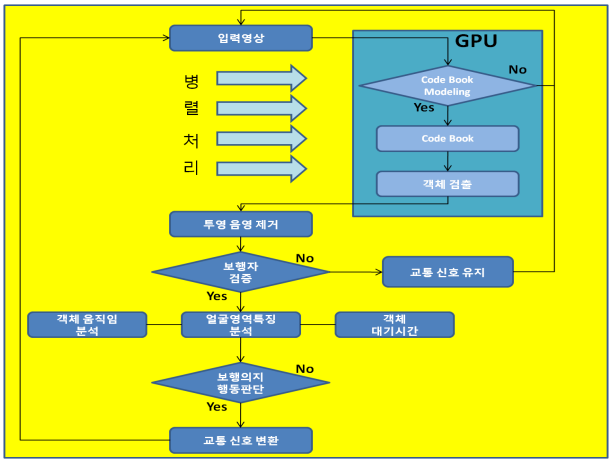


그림 1. 시스템 흐름도.
Fig. 1. System Flowchart.

2.2 보행자 객체 검출

동영상으로부터 입력된 3채널 프레임에서 보행자 객체를 분리하기 위한 단계를 제일 먼저 수행한다. 일정한 프레임을 입력받고 코드북을 통해 배경을 모델링한다.

모델링된 코드북은 입력된 프레임의 전체 픽셀에 대응되는 배경의 상태 정보를 저장하고 있는다. 그 이후로 입력된 프레임의 픽셀값과 대응되는 코드북의 상태를 비교하여 배경은 0, 보행자 객체는 1의 값을 가지는 이진영상으로 얻는다. 이진영상에서는 잡음이 존재하기 때문에 모폴로지 연산을 수행하여 잡음을 제거한다.

보행자 객체를 분리하는 과정은 입력된 프레임의 전체픽셀과 그에 대응되는 코드북을 이용하기 때문에 많은 시간이 소요된다.

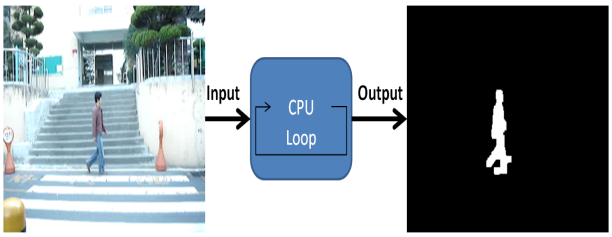


그림 2. CPU 처리 방법.
Fig. 2. CPU processing method.

그림2와 같이 기존의 CPU 프로세서의 처리방식은 순차적으로 루프를 돌면서 이미지 정보에 접근하기 때문에 정보의 수량에 따라 처리시간이 증가할 수밖에 없다. CPU의 순차적인 접근 방식보다는 정보에 동시접근이 가능한 병렬처리방식을 사용한다면 더 효율적인 처리가 가능하다.

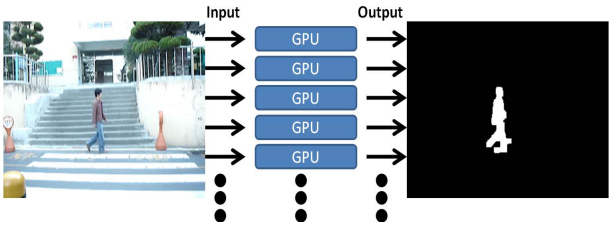


그림 3. GPU 처리 방법.
Fig. 3. GPU processing method.

그림3과 같이 GPU의 다중 코어를 이용하면 이미지 정보에 병렬적으로 동시 접근하여 정보를 처리할 수 있다. GPU 프로세서 코어는 CPU 프로세서 코어보다 단순하고 덜 강력하지만 수 백개가 들어있기 때문에 동시에 수 백 개의 작업을 처리할 수 있는 이점이 있다. 그러므로 실시간환경에 적합한 처리속도를 위해 GPU를 이용하여 코드북 모델링 및 객체검출 과정을 병렬화하여 처리한다.

2.3 투영 음영 제거

(a) (b)
그림 4. 투영 음영이 포함된 객체들.
Fig. 4. Objects included shadow.

실외에서 획득한 영상의 경우에는 태양광의 위치가 수시

로 변하기 때문에 태양의 고도가 높은 점심에는 투영 음영의 길이가 짧고, 태양의 고도가 낮은 아침이나 저녁에는 긴 형태의 투영 음영이 발생한다. 그림 4의 (a)는 태양광에 의해 보행자 객체에서 투영 음영이 발생한 영상이고 (b)는 보행자 객체를 검출한 이진 영상이다.

이렇게 투영 음영이 포함되면 정확한 객체 검출이 되기 않기 때문에 형상정보와 형판정합 기법을 이용한 보행자 검증에 있어서 문제가 발생한다. 즉, 투영 음영이 존재하지 않는 모델들의 형상정보와 투영 음영이 존재하는 객체의 형상정보를 비교하면 전혀 다른 결과가 나오는 것이다. 따라서 정확한 보행자 검증을 위해 투영 음영을 제거하여 검증 과정을 수행한다.

투영 음영부분은 경계선 검출 시, 다른 경계선 값보다 완만한 값을 가지는 특징이 있다. 이것은 객체영상의 전체 경계선 값의 평균보다 낮은 값을 가지는 것이다.

경계선 검출을 하기위해 라플라시안 마스크(Laplacian Mask)를 사용하였으며 마스크는 그림5와 같다.

-1	-1	-1
-1	8	-1
-1	-1	-1

그림 5. 라플라시안 마스크.
Fig. 5. Laplacian mask.

라플라시안으로 구한 객체영상의 평균 경계선 값은 식(1)에 의해서 구해지고, 또한 평균 경계값은 임계치(th)로 설정된다.

$$th = \sum_{x,y} Object(x,y) / (width \times height) \quad (1)$$

식(1)에 의해서 설정된 임계치를 기준으로 경계선 영상의 값을 수식(2)와 같이 비교하여 밝기값이 완만한 투영 음영 부분이 제거된 이진영상을 만든다.

$$Output(x,y) = \begin{pmatrix} 1 & \text{if } Object(x,y) > th \\ 0 & \text{if } Object(x,y) < th \end{pmatrix} \quad (2)$$

(a) (b) (c)

그림 6. 투영 음영 제거.
Fig. 6. Result after removing shadow.

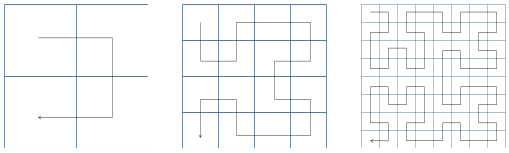
그림6의 (a)는 추출된 객체의 회색조 영상이고, (b)는 수식(1),(2)에 의해서 만들어진 이진영상이며, (c)는 투영 음영 부분을 제거한 후에 위치정보를 수정하여 새롭게 추출된 객체이다.

2.4 보행자 검증

검출된 객체가 보행자인지 아니면 차량이나 동물등과 같은 객체인지를 판별하기 위하여 보행자 검증을 수행한다. 보행자 검증은 형상 정보를 이용하여 형판정합기법을 사용한다.

본 논문에서는 힐버트 패스(Hilbert Path)를 이용하여 1차원에서 유사도 측정이 가능한 힐버트 스캔 거리값(Hilbert Scan Distance)[6,7]을 사용한 형판정합 기법을 이용하여 보행자를 검증한다.

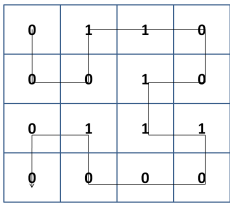
힐버트 패스는 1891년 David Hilbert에 의해 고안되었으며, 영상의 모든 영역을 포함하는 단방향의 패스를 나타낸다. 힐버트 패스는 2의 거듭 제곱근으로만 표현이 가능하고, 차수가 d일 경우 힐버트 패스의 크기는 $2^d \times 2^d$ 가 된다. 그림 7은 힐버트 차수에 따른 패스 경로이다.



(a) d = 1 (b) d = 2 (c) d = 3

그림 7. 힐버트 차수에 따른 패스.
Fig. 7. Hilbert path according to the order.

그림 8은 힐버트 패스를 이용하여 2차원의 영상을 1차원으로 변환한 것이다. (a)에서 1은 외곽선이 추출된 픽셀을 의미하여 (b)는 힐버트 패스에 따라 1차원 배열로 변환된 것이다.



(a) 2차원의 외곽선 영상

0	0	0	1	1	0	0	1	1	1	0	0	0	1	0

(b) 1차원 배열로 변환한 결과

그림 8. 힐버트 패스를 이용한 1차원 배열 변환.
Fig. 8. Transformation image into 1D array using Hilbert path.

힐버트 스캔 거리값을 이용하여 추출된 객체와 모델과의 유사성을 판별하는 단계는 그림 9과 같다.

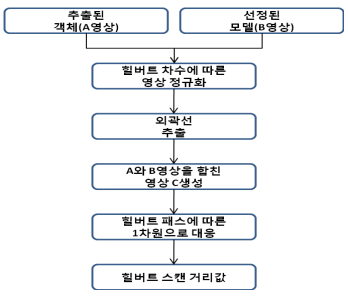


그림 9. 힐버트 스캔 거리값을 이용한 검증 흐름도
Fig. 9. Flowchart of verification process using Hilbert scan distance.

힐버트 스캔 거리값을 구하기 위한 과정은 총 7단계로 이루어지며 그 과정은 다음과 같다.

1) 추출된 객체 영상(A영상)과 미리 선정되어 있는 모델 영상(B영상)을 세로 및 가로비를 고려하여 힐버트 차수에 맞게 정규화 한다.
2) 정규환된 A영상과 B영상의 외곽선을 추출한다. 단 여기서 A영상의 외곽선 값은 '1'로 정의하고 B영상의 값은 '2'로 정의한다.
3) A영상과 B영상을 합한 새로운 C영상을 생성한다.
4) C영상에 대하여 힐버트 패스에 따른 1차원으로 대응시킨다.
5) 1차원으로 대응된 C에서 값이 '2'이면 다음 외곽선 픽셀로 이동하고, '1'이면 가장 가까운 '2' 또는 '3'을 찾아서 거리값을 구한다. 만약 값이 '3'이면 해당 픽셀은 겹쳐지기 때문에 거리값은 '0'이다.
6) 5)에 의해서 구해진 거리값의 평균을 구한다. 또한 영상의 방향성을 고려하여 5)의 값을 바꿔서 동일하게 수행하여 거리값의 평균을 구한다.
7) 두 개의 평균 거리값 중 큰 값을 힐버트 스캔 거리값으로 선택한다.

검출된 객체는 템플릿 매칭을 이용하여 보행자, 자동차, 동물에 대해 각각 위의 절차에 따라 수행하여 힐버트 스캔 거리값을 구한다. 구해진 3가지의 거리값 중에서 가장 낮은 거리값을 가지는 모델과 흡사하다고 판단한다.

그림 10은 보행자 검증에 사용된 대표모델이다. 대표모델을 선택하는 것은 템플릿 매칭에서 중요한 요소로 작용하기에 형상 정보가 일반적인 것들을 선택하는 것이 중요하다. 대표모델의 선택은 횡단보도 지역에서 가장 흔히 나타날 수 있는 (a)사람, (b)자동차, (c)동물, 세가지를 대표모델로 선택하였다.

(a) 사람

(b) 자동차 (c) 동물

그림 10. 검증에 사용된 모델.

Fig. 10. Representative models.

2.5 객체 움직임 분석

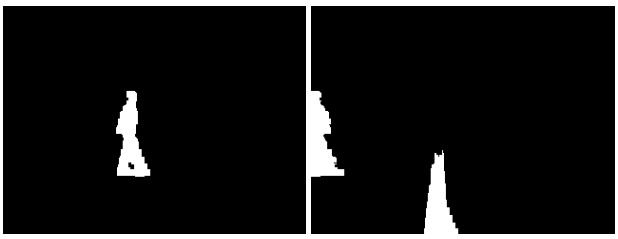
보행자로 식별되었다면 움직임을 분석한다. 그러기 위해서 보행자 객체 검출에서 획득한 외곽선 위치정보를 이용한다. 최대, 최소 폭과 높이의 중심값을 구한다. 만약 검출된 보행자가 이동 중이라면 다음 입력 프레임에서 검출된 보행자 객체의 중심값은 이전 중심값과 차이가 있고, 대기 중이라면 차이가 없다. 이 중심값의 비교에 따라 차이가 나타난다면 이동 중이라고 판단하고, 미리 정해진 시간동안 차이가 나타나지 않는다면 대기 중으로 판단한다.

2.6 얼굴영역 특징 분석

신호 대기 중인 사람임을 검증하는 다른 방법은 얼굴이 정면 즉 건너 편에 있는 카메라를 보고 있는지 혹은 옆을 쳐다 보는 지를 알아보는 것이다.

사람의 정면모습과 측면모습일 때의 얼굴비율은 비율적인 차이를 가지는 특징이 있다. 건널목을 이용하려는 보행자의 특징은 건널목 신호등을 확인하기 위해서 정면을 주시하기 때문에 정면의 모습일 것이고, 일반적으로 지나가는 보행자는 건널목 신호등을 확인할 필요가 없으므로 측면의 모습이다.

이러한 특징을 분석하는 것은 추출된 얼굴영역의 비율로써 판단하는데, 그 기준은 검출된 얼굴영역에서 머리비율이 얼굴 비율보다 크다면 측면 모습으로 판단하고 그와 반대의 경우라면 정면 모습으로 판단한다. 이런 특징을 분석하기 위해서 보행자 객체에서 얼굴영역을 추출해야한다. 얼굴영역을 검출하기 위해 이진화 영상에서 투상 영상을 생성한다.



(a) (b)

그림 11. 투상 영상.

Fig. 11. Projection image.

그림 11의 (a)는 보행자 객체 검출에 의한 이진영상이고, (b)는 (a)영상에 따른 수평, 수직 히스토그램 영상이다. 투상 히스토그램에서 사람의 체형 정보를 바탕으로 하여 얼굴영역을 검출한다. 사람의 체형정보에 의한 얼굴영역은 보행자 객체를 상단부와 하단부로 나누었 때 상단부에 존재하고, 몸 전체에 비해 적은 정보로 이루어진다 그림 12는 히스토그램의 체형 정보를 이용하여 얼굴 영역을 검출한 결과이다.

그림 12. 얼굴영역 추출.

Fig. 12. Extraction of Face area.

2.7 보행자 행동 판단 및 신호제어

보행자의 행동을 판단하여 신호를 제어 하기 위해서는 앞 단계에서 수행된 결과에 따라 판단한다. 보행자의 초기 판단은 얼굴특징을 이용한다. 검출된 객체가 이동중이고 측면 모습으로 판단되었다면 이 보행자는 보행으로 추정하고 교통신호를 기존과 동일하게 유지한다. 만약 객체가 이동중이고 정면 모습으로 판단되었다 하더라도 보행으로 추정하고 신호는 그대로 유지한다.

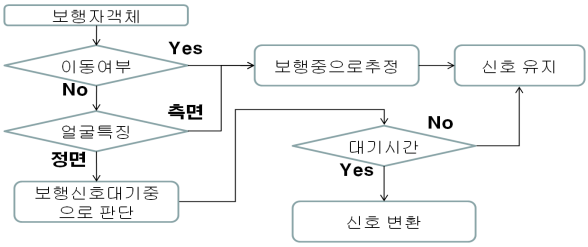


그림 13. 행동 판단 흐름도
Fig 13. Motion Interpretation Flowchart.

그러나 검출된 객체가 어느 한 지점에서 대기 중이고 정면의 모습으로 판단되었으면 보행신호대기후보자로 판단하고 대기시간을 측정한다. 만약 대시기간이 일정한 값에 도달하면 이 보행자는 보행신호대기로 판단하고 교통신호를 차량신호등 적색, 보행신호등 청색으로 제어한다. 또한 변환된 신호는 고정된 시간이 지나면 초기 상태인 차량신호등 청색, 보행신호등 적색으로 변환되어 유지된다.

3. 실험 및 결과

3.1 프로그램 구현

본 논문에서 제안한 CUDA기반의 보행자 행동판단방법을 평가하기 위하여 코어2듀오 2.2GHz 프로세서와 메모리가 2GB인 PC에서 구현하였다. 도구로는 Visual C++과 간단한 영상처리를 위해 OpenCV 라이브러리를 사용하였다. 또한 일부 병렬처리과정을 위해 CUDA SDK 2.3과 NVIDA사의 GeForce GTX 260 그래픽카드를 이용하였다. GTX 260의 사양은 표 1과 같다.

표 1. GTX 260 사양.
Tabel 1. GTX 260 Specifications.

GPU 사양	
CUDA 프로세서 코어	192
그래픽 클럭(MHz)	576 MHz
프로세서 클럭(MHz)	1242 MHz
메모리 환경	896MB

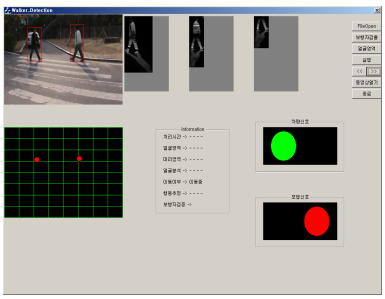


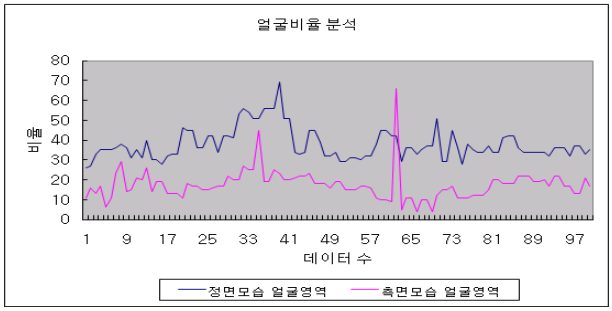
그림 14. 구현된 프로그램.
Fig 14. Implemented program.

실험 데이터는 횡단보도가 존재하는 영역을 대상으로 실험하였으며, 횡단보도 양방향이 아닌 한 방향에서만 카메라가 존재한다는 제약을 두고 영상을 획득하였다. 교통신호등과 같은 하드웨어적인 부분은 실제 제작이 어려운 관계로 프로그램 상 단순 이미지로 구현하였다. 그림 14는 구현된 프로그램의 모습이다. 본 실험에 사용된 영상은 320 × 240 사이즈의 컬러영상이며, 프레임 속도는 25 frames/sec 이다.

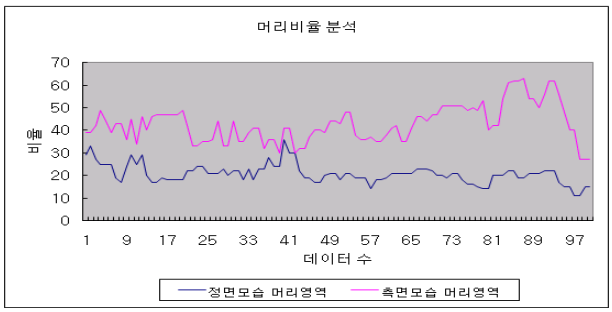
3.2 실험 결과

우리는 신호를 기다리고 있는 사람인지 지나가는 사람인지를 판단하기 위해 얼굴비율을 이용한다. 얼굴비율을 이용한 행동판단에 대한 성능을 분석하기 위해 횡단보도지역의 촬영한 동영상에서 신호 대기 중인 사람의 영상과 지나가는 사람의 영상을 각각 100장씩 분석하였다.

그림 15는 얼굴특징분석 비율에 관한 데이터들이며, (a) 그래프는 정면과 측면 모습의 얼굴영역의 비율이다. 남색부분이 정면모습의 얼굴영역, 본홍색 부분이 측면모습의 얼굴영역이다. 정면모습의 얼굴영역비율이 측면모습의 얼굴영역비율 보다 높은 것을 확인할 수 있다. (b)그래프는 정면과 측면 모습의 머리영역의 비율로 (a) 그래프와는 반대의 결과를 확인할 수 있었다. 그림 15의 데이터를 사용해 얼굴영역 특징 분석에 관한 정확도는 표2와 같다.



(a) 얼굴비율



(b) 머리비율

그림 15. 얼굴영역 비율 분석 결과.
Fig. 15. Face ratio analysis result.

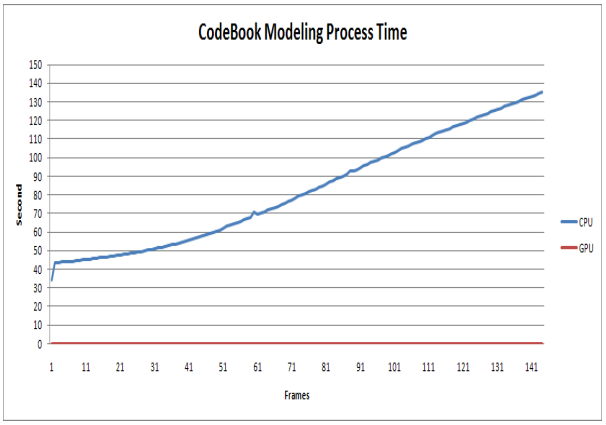
표 2. 얼굴영역특징분석 결과.
Tabel 2. Face feature analysis result.

구분	정면으로 판단(%)	측면으로 판단(%)	합계(%)
정면 모습일때	96	4	100
측면 모습일때	11	89	100

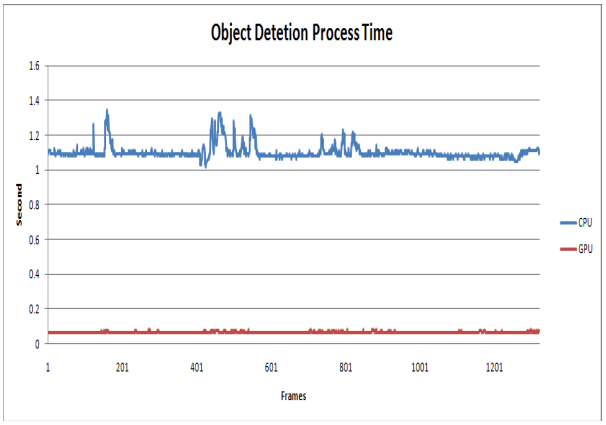
얼굴영역특징분석의 정확도는 보행상태일 때 측면모습으로 판단한 경우가 96%이며, 정면모습으로 오판단한 경우가 4%로 다소 높은 정확도를 보였다. 그러나 대기상태일 경우 정면모습으로 판단한 경우가 89%이며, 옆모습으로 결정한 경우가 11%로 다소 낮은 정확도를 보였다.

정면모습을 측면으로 오판단한 4%와 측면모습을 정면으로 오판단한 11%는 얼굴영역 검출과정에서 발생한 문제였다. 얼굴영역 검출시 머리 부분과 목부분이 같이 영역화 되었기 때문에 잘못 판단을 한 것으로 판단된다.

그림 16의 그래프는 코드북 모델링과 객체 검출 처리시간의 CPU와 CUDA방식의 비교결과이다.



(a) Code Book Modeling Process Time



(b) Object Detection Process Time

그림 16. 처리시간 분석 그래프.
Fig. 16. Processing time analysis graph.

그림 16의 (a),(b)는 구현된 알고리즘을 CPU와 GPU에서 한 프레임당 100번씩 반복하여 실험한 결과이며, 청색부분이 CPU의 처리시간이고 적색부분이 GPU의 병렬처리 속도를 보여주고 있다. 코드북 모델링 부분에서는 CPU는 처리시간이 증가하는 현상이 나타났고, GPU는 일정하고 빠른 처리속도를 보였다.

물체 탐지 부분에서도 마찬가지로 CPU의 처리속도보다 GPU의 병렬처리가 약 18배 빠른 결과를 보였다. 또한 그림 15의 그래프를 확인해보면 CUDA의 처리속도의 패턴은 안정적이고 유동이 없는 반면에 CPU의 순차적인 처리에서는 시간이 지남에 따라 점점 처리속도가 비효율적인 결과를 확인할 수 있었다. CUDA방식의 병렬처리가 CPU의 순차적인 처리보다 더 효율적이고 실시간 시스템에 적합하다고 할 수 있다. 표3,4은 코드북 배경모델링과 객체 검출과정을 CPU와 GPU의 최소, 최대 ,평균 처리시간을 정리한 결과이다.

표 3. 배경모델링 처리 시간.
Tabel 3. Background modeling time.

CodeBook Modeling	최저	최고	평균
CPU	34.03	135.17	82.22
GPU	0.0	0.01	0.004

표 4. 객체 검출 처리 시간
Tabel 4. Object detection processing time.

Object Detection	최저	최고	평균
CPU	1.01	1.34	1.10
GPU	0.06	0.07	0.06

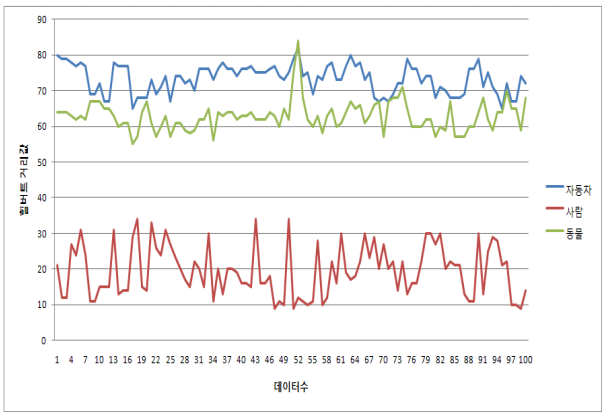


그림 17. 힐버트 스캔 거리값 결과.
Fig. 17. Hilbert scan distance result.

그림 17의 그래프는 힐버트 스캔 거리값을 이용하여 검출된 보행자 객체와 대표모델간의 유사도를 측정한 결과이다. 힐버트 스캔 거리값의 성능을 측정하기 위해 실제로 검출된 보행자, 자동차, 동물에 대한 영상을 100장씩 분석하였다.

힐버트 패스의 5차수 정규화를 통하여 얻은 결과이며, 단순 외곽선 정보만으로도 보행자 객체의 검증이 가능하였다. 또한 그래프를 보면 보행자와 자동차, 동물간의 힐버트 스캔 거리값 차이가 확연히 나는 결과를 확인할 수 있는데 이것은 사람과 객체사이의 분류 정확도가 높다고 할 수 있다. 표5는 보행자 검증의 정확도를 나타낸 것이다.

표 5. 보행자 검증 결과
Tabel 5. Pedestrian verification result.

구분	사람 모델(%)	자동차 모델(%)	동물 모델(%)	합계 (%)
보행자	100	0	0	100
자동차	0	49	51	100
동물	0	43	57	100

표 5의 결과를 보면 보행자의 검증은 100%이며 정확한 식별이 가능하였지만, 자동차와 동물의 식별 정확도는 많이 부족하였다. 그러나 본 논문에서는 보행자의 행동을 판단하기 위해 보행자와 다른 객체를 정확히 판별하는 것이 중요한 관심사이기 때문에 보행자와 다른 객체간의 정확한 분류가 가능하다면 자동차와 동물간의 식별 정확도가 부족하더라도 문제 되지 않는다고 판단하였다.

4. 결 론

본 논문에서는 교통신호를 능동적으로 제어하기 위하여 CUDA를 기반으로 보행자의 행동판단 방법을 제안하였다. 횡단보도영역에서 촬영된 동영상에서 코드북의 비교를 통하여 객체를 검출하였다. 투영 음영이 존재할 경우에는 이를 제거하고 외곽선 정보와 힐버트 스캔 거리값을 이용한 형판정합 기법으로 보행자인지 잡음(차량,동물)인지를 식별하였다.

보행자로 식별되었을 때는 이동여부를 판단하고 수평, 수직 투상 히스토그램과 사람의 체형 정보를 바탕으로 얼굴영역을 검출한다. 마지막으로 얼굴특징, 움직임, 대기시간 분석을 통하여 보행자의 건널목 이용 의지 행동을 판단하고 능동적으로 교통신호를 제어한다.

CUDA를 기반하여 프레임 전체 픽셀을 처리하는 부분을 병렬화 하여 실시간 환경에 적합한 결과를 보였고, 정확한 보행자 행동을 판단 할 수 있었다. 또한 힐버트 스캔을 이용한 보행자 객체의 분류의 정확도가 높게 나타났다. 그러나 얼굴비율분석의 정확도면에서 다소 부족한 점을 보였다.

향후 연구방향은 줌기능의 카메라를 사용하여 얼굴을 관찰함으로써 횡단보도를 건너려는 의지를 세밀히 확인하는 방법과, 처리속도의 향상을 위하여 전체적인 알고리즘을 병렬화하는 연구를 진행하고자 한다.

참 고 문 헌

[1] 윤재홍, 지윤강 “영상에서 객체 추출을 통한 적응형 통행 우선순위 교통신호 제어 시뮬레이션,” *한국멀티미디어학회 논문지*, 제11권, 8호, pp. 1051-1058, 2008.

[2] 김철기, 강이철, 차의영, “횡단보도 신호제어를 위한 보행자 추적,” *한국멀티미디어학회 춘계학술발표논문집*, 제2권, 1호, pp.391-396, 1999.

[3] K. Kim, T. H. Chalidabhonge, D. Harwood, L. Davis. “Real-time foreground-background segmentation unsing codebook model,” *Elsevier Real-Time Imaging,* vol. 11, pp. 172-185, 2005.

[4] Yongbin Li, Feng Chen, Wenli Xu, and Youtian Du, “Gaussian-Based Codebook Model for Video Background Subtraction,” *Lecture Notes in Computer Science*, pp. 762-765, September 2006.

[5] Mohamad Hoseyn Sigari, and Mahmood Fathy, “Real-time Background Modeling Subtraction using Two-Layer Codebook Model,” *International MultiConfernece of Engineers and Computer Scientists*, vol. 1, March 2008.

[6] Li TIAN, et al, “A Fast and Accurate Algorithm for Matching Images Using Hilbert Scanning Distance with Threshold Elimination Function,” *IEICE INF.&SYST*, vol.89, pp. 209-297, 2006.

[7] Bongki Moon, H.V. Jagadish, Christos Faloutsos, etal, “Analysis of the Clustering Properties of the Hilbert Space-Filling Curve,” *IEEE Transactions on Knowledge and Data Engineering*, vol.13, no.1, pp. 124-141, Jan/Fed 2001.

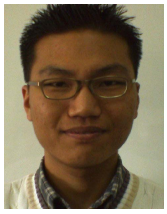
저 자 소 개

이홍창(Hong-Chang Lee)
2009년 : 경남대 컴퓨터공학부 졸업
2009년～현재 : 동 대학원 첨단공학과 석사과정

관심분야 : 영상 처리, 컴퓨터 비전, 모션 분석
Phone : 011 - 9314 - 5122
E-mail : square@hanma.kr

이상용(Sang-Yong Rhee)
1982년 : 고려대 산업공학과 졸업.
1984년 : 고려대 대학원 산업공학과 (공학석사)
1992년 : 포항공대 대학원 산업공학과 (공학박사)
1992년～현재: 경남대학교 컴퓨터공학부 교수

관심분야 : 컴퓨터 비젼, 증강현실, 뉴로-퍼지, 지능 로봇
Phone : 055 - 249 - 2706
E-mail : syrhee@kyungnam.ac.kr



김영백(Young-Baek Kim)
2005년 : 경남대학교 컴퓨터 공학부 졸업.
2007년 : 경남대 대학원 컴퓨터공학과 (공학석사)
2007～현재 : 경남대 대학원 컴퓨터공학과 박사과정

관심분야 : 영상처리, 컴퓨터 비젼, 증강현실
Phone : 010 - 5779 - 4783
E-mail : baroaleum@gmail.com