

다중 프레임에서의 보행자 검출 및 삭제 알고리즘

(Automatic Pedestrian Removal Algorithm Using Multiple Frames)

김창성*, 이동석**, 박동선***

(ChangSeong Kim, DongSuk Lee, Dong Sun Park)

요약

본 논문은 영상에서 효과적으로 보행자를 삭제하는 자동 삭제 시스템을 제안한다. 첫 번째로 Histogram of Oriented Gradient(HOG) / Linear-Support Vector Machine(L-SVM)분류기를 이용하여 보행자를 찾고, 참조영상으로부터 적절한 배경을 습득하여 삭제될 보행자를 대체한다. 배경은 참조영상 내에서 검색하며 변경된 feather blender 연산은 대체 영역의 경계를 자연스럽게 만든다. 기존에 존재하던 대부분의 시스템이 수동인 것에 반해 제안된 시스템은 자동으로 객체를 검출하고 자연스러운 배경을 생성한다. 실험결과 대체된 영역의 PSNR 평균은 19.246으로 측정되었다.

■ 중심어 : 보행자 검출; 객체삭제; 자동화

Abstract

In this paper, we propose an efficient automatic pedestrian removal system from a frame in a video sequence. It firstly finds pedestrians from the frame using a Histogram of Oriented Gradient(HOG) / Linear-Support Vector Machine(L-SVM) classifier, searches for proper background patches, and then the patches are used to replace the deleted pedestrians. Background patches are retrieved from the reference video sequence and a modified feather blender algorithm is applied to make boundaries of replaced blocks look naturally. The proposed system, is designed to automatically detect object and generate natural-looking patches, while most existing systems provide search operation in manual. In the experiment, the average PSNR of the replaced blocks is 19.246

■ keywords : Pedestrian Detection; Object Removal; Automatic

I. 서론

현 사회는 다양한 영상기기의 발달과 보급으로 인하여 방대한 양의 영상 데이터가 쏟아지고 있다. 디지털 영상 사용자 중 가장 많은 비중을 차지하는 집단은 영상기기를 이용하여 영상을 수집할 뿐, 편집능력이 부족한 일반 사용자이다. 이들의 겪는 문제 중 하나는 원하지 않는 보행자가 영상에 나타나는 것이며 이러한 문제를 원본영상을 훼손하지 않으면서 쉽게 해결하기 원한다. 본 논문 또한 이러한 사회적 흐름에 맞춰 영상에서

사용자가 원치 않는 보행자 정보를 자동으로 삭제해주고 자연스럽게 배경과 대체하는 보행자 삭제 시스템을 제안 하고자 한다.

기존 영상에서 특정 영역 또는 객체를 삭제하고자 하는 다양한 시도가 있었고 시중에 다양한 제품이 출시되어 있다.[1][2][3][4] 하지만 기존제품들은 프로그램이 자동으로 객체를 검출하여 삭제하는 것이 아닌 사용자가 원하는 영역을 직접 삭제 하여야 했다. 이러한 방법은 사용자의 삭제 프로그램 숙련도에 따라 삭제 결과가 확연히 차이가 나며 작업에 소요되는 시간 또한 오래 걸린다. 본 논문에서는 영상내의 대상객체를 식별하고 객체

* 학생회원, 전북대학교 컴퓨터공학과

** 학생회원, 전북대학교 전자공학과

*** 정회원, 전북대학교 IT융합연구센터

이 논문은 2013년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2013R1A1A2013778). 이 논문은 한국연구재단 BK21 플러스 사업의 지원을 받았다.

가 식별된 객체영역을 자동으로 삭제해주는 프로그램을 제안하고자 한다.

보행자 검출은 보행자가 갖는 특징을 추출하고 분류기를 이용 추출한 특징을 분리하는 학습과정을 거친다. 보행자 검출에는 가장 일반적인 HOG(Histogram of Oriented Gradient)특징을 사용하였다. HOG란 그라디언트의 크기와 각도를 히스토그램으로 쌓아 보행자의 그라디언트 분포를 보여주는 특징이다. HOG특징은 보행자에 대한 구별 능력이 뛰어나고 노이즈에 강한 특성을 보이기 때문에 보행자 인식을 위해 가장 많이 사용되는 특징이다.[5][6]

SVM은 학습 데이터인 보행자 샘플(positive sample)과 비 보행자 샘플(negative sample)간의 거리를 구하고 그 가운데에서 마진(margin)이 최대가 되는 최적의 초평면(hyperplane)을 산출하여 판단하는 알고리즘이다.[7] 학습에 사용된 분류기는 선형 SVM(Linear - Support Vector Machine)으로서 학습된 데이터를 기준으로 판별하고자 하는 객체가 어떤 분류 집단에 속하는지를 판별한다.

이후 학습한 분류결과를 기준으로 삭제를 원하는 시점 프레임에서 삭제 대상객체인 보행자를 검출한다. 검출된 보행자를 덮어씌워 삭제할 배경을 획득하기 위해 보행자가 검출된 위치를 주변 프레임에서 탐색한다. 탐색은 선택 프레임으로부터 시간차가 적은 위치부터 선행되며 이는 시간변화에 따른 배경의 이질감을 최소화하기 위함이다. 프레임 탐색에 앞서 선택 프레임과 탐색 프레임간의 차 영상을 통하여 보행자의 움직임 정보를 파악하고 보행자의 움직임이 없는 프레임은 보행자가 윈도우내에 남아있다고 판단하여 탐색대상 프레임에서 제외하였다. 보행자가 검출되었던 위치와 주변영역에 보행자 객체가 없을 경우 이를 배경으로 획득하고 획득한 배경을 삭제 대상 영역에 덮어씌움으로서 객체 삭제가 수행된다. 대상 프레임과 배경 사이의 이질감 발생을 최소화하기 위하여 배경에 변경된 페더 블렌더를 적용한다. 프로그램이 특정객체를 탐색하고 이를 삭제하는 기능은 기존의 연구에서 찾아볼 수 없기 때문에 그 중요성이 매우 높다고 할 수 있다.

본 논문의 구성은 다음과 같다. 2장은 HOG와 SVM을 이용한 학습 방법에 대하여 소개하고, 블러링상을 제거하기 위해 사용하고자 하였던 Feather Blender에 대하여 알아본다. 3장은 본 논문에서 제안하는 삭제가 알고리즘과 변경된 페더 블렌더를 설명하고, 4장은 각 과정에서의 연산 결과 확인과 제안한 보행자 객체 삭제기

성능평가 등 실험을 수행 및 분석한다. 마지막으로 5장은 결론 및 향후 연구 방향에 대해 이야기 한다.

II. 관련 연구

2.1 Histogram Of Gradients(HOG)

HOG특징은 Dalal가 보행자 검출을 목적으로 제안한 특징이다.[8] 컴퓨터비전과 영상처리에 많이 쓰이고 있으며, HOG 특징은 입력 영상의 모든 블록의 히스토그램들을 모아 정규화 한 벡터이다. HOG는 그라디언트와 외각 방향의 분포를 특징으로 하는 히스토그램을 의미하며 윈도우내에서 그라디언트 크기와 방향을 이용하여 셀 단위로 히스토그램을 만들고, 블록 단위로 히스토그램을 정규화 한 후 이어 붙여, 이들의 집합을 특징 벡터로 사용하는 것을 말한다.

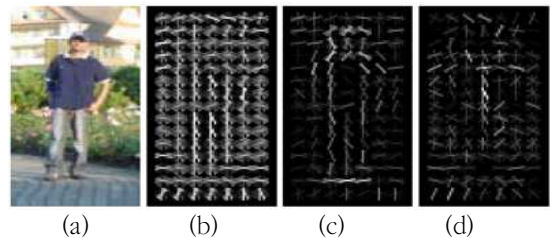


그림 1 HOG를 사용한 특징 추출

- (a) 테스트 이미지
- (b) HOG 기술자 이미지
- (c) 보행자 샘플 SVM가중치 적용 이미지
- (d) 비 보행자 샘플 SVM가중치 적용 이미지

그림1의 (a)는 테스트 이미지이며 (b)는 계산된 HOG 기술자를 나타낸 모습이고 (c)는 SVM가중치가 적용된 보행자 HOG기술자의 모습 (d)는 SVM가중치가 적용된 비 보행자 HOG기술자의 모습이다.

2.2 Support Vector Machine(SVM)

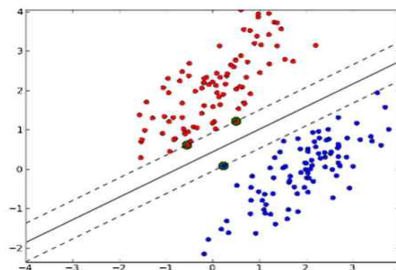


그림 2 추출된 특징을 나누는 분류기 SVM

SVM은 Support Vector(그림2의 초록색 점)간의 마진이 가장 큰 분류 선을 그어줌으로서 일반화 능력을 극대화 시킨 분류기이다. SVM의 다른 특징으로는 차원증가를 통한 선형분류가 가능하다는 점이 있다. 판별하고자 하는 대상의 특징이 들어오면 분류기준을 바탕으로 어느 객체 집단에 속하는지를 분류해 준다.

2.3 Feather Blender

그림3은 흰 원 가장자리 영역에 Feathering Mask를 적용시킨 모습이다. 가장자리 영역이 점차 벌어지는 형태를 보이는 것을 확인할 수 있다. Feather Blender는 덧붙일 배경의 가장자리 영역과 기존 배경의 영역이 자연스럽게 섞여서 결과 영상에서 이질감이 생기는 현상을 방지하는 알고리즘이다. 그러나 기존 알고리즘은 적용 대상 객체의 크기가 아주 작은 경우에는 적용 후 변화가 거의 없게 되며 이질감 또한 해결하지 못하는 단점을 가지고 있다.



그림 3 Feather Blender를 적용한 타원

Ⅲ. 제안하는 알고리즘

본 논문에서 제안하는 시스템은 그림4와 같이 보행자 검출기, 배경탐색기,삭제기로 구성된다. 보행자 검출기(Pedestrian Detector)는 사용자가 선택한 프레임에서 SVM을 사용하여 분류된 HOG특징을 기준으로 보행자를 검출하는 기능을 수행한다. 배경탐색기(Background Searcher)는 프레임이 이동하며 보행자 삭제에 이용할 배경을 습득한다. 삭제기(Eraser)는 삭제영역과 배경을 전달받아 보행자를 삭제한다.

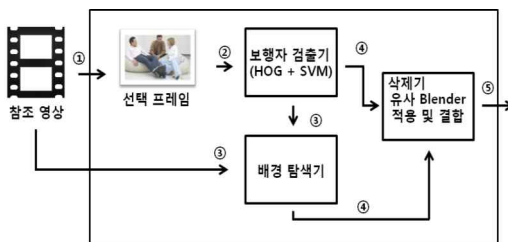


그림 4 삭제 시스템 구조

시스템의 전반적인 흐름은 다음과 같다. ①참조 영상으로부터 삭제를 수행할 프레임을 선택한다. ②보행자 검출기는 사용자가 지정한 프레임 내에서 보행자를 검출한다. ③보행자 검출기는 보행자가 검출된 위치정보를 배경탐색기에 전달한다. 탐색할 이웃 프레임은 참조 영상에서 전달받는다. ④배경 탐색기는 이웃 프레임을 이동하며 전달받은 위치를 탐색하게 된다. 탐색결과 해당위치에 보행자가 없다면, 이를 배경으로 습득하여 삭제기에 전달한다. ⑤ 삭제기는 전달받은 배경을 선택프레임에 덮어씌우는 방식으로 삭제를 진행하게 되는데 습득 배경과 선택 프레임간의 이질감을 최소화하기 위하여 Feather Blender와 유사 기능을 갖는 변경된 페더블랜더를 적용한 후 삭제 연산을 진행한다.

3.1보행자 검출기



그림 5 학습 처리과정

학습은 그림 5와 같이 전처리 → HOG특징 추출 → SVM분류 순서로 진행된다. 보행자 검출 또한 탐색중인 윈도우 이미지의 HOG특징을 추출하고 추출된 특징을 사전에 학습된 SVM 분류기를 사용하여 윈도우내의 이미지가 보행자 집단에 속하는지 판별하게 된다. 위 과정을 통해 그림6과 같이 삭제를 진행할 프레임에서 보행자를 검출하게 된다. 검출된 보행자의 위치 정보는 배경 탐색기에 전달되게 된다.



그림 6 보행자 검출 결과

3.2 배경 탐색기

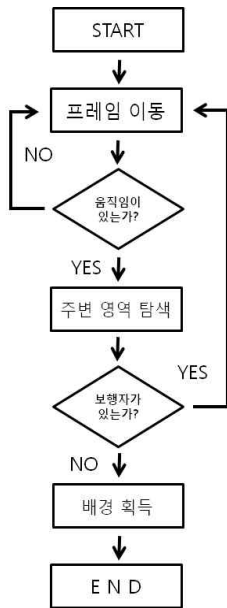


그림 7 배경 탐색기 흐름도

배경 탐색기의 흐름은 그림7과 같으며 각 과정의 세부 내용은 다음과 같다.

3.2.1 프레임 탐색 순서



그림 8 주변 프레임 탐색

삭제에 사용될 배경을 획득하기 위하여 선택 프레임에서 시간의 변화가 작은 프레임을 우선순위를 높게 두고 선택프레임(t프레임)의 전후 양방향(t+n 프레임, t-n 프레임)으로 프레임을 번갈아 이동하며 탐색을 진행하게 된다.

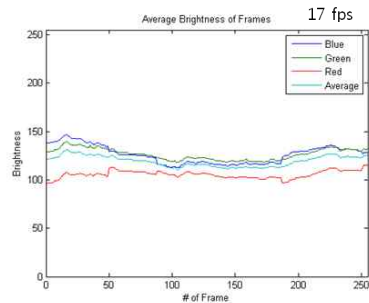


그림 9 시간에 따른 밝기변화 (영상1)

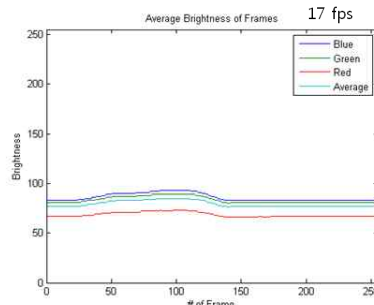
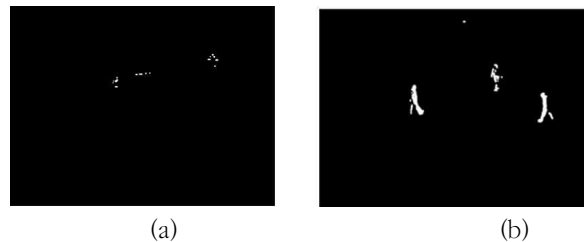


그림 10 시간에 따른 밝기변화 (영상2)

시간의 변화가 작은 이웃 프레임일수록 명도 등 환경 변화가 적어, 배경을 획득하기 적합하다고 가정하였고 이를 확인하기 위하여 영상에서 움직이는 객체가 등장하지 않는 구간의 시간에 따른 밝기변화를 측정했다. 그림9, 그림10의 그래프를 분석한 결과 시간변화에 따른 밝기 변화가 존재하며 특정 프레임으로부터 시간의 변화와 무관한 프레임보다 이전 이후 시간대의 밝기 값이 더 유사한 특성을 보임을 알 수 있다.

3.2.2 보행자 움직임 정보



(a) (b)
그림 11 객체 움직임 정보

- (a) 객체 움직임이 미미한 경우
- (b) 객체 움직임이 활발한 경우

연산이 상대적으로 많은 HOG의 특성상 모든 이웃 프레

임을 검사한다면 시간이 많이 소요된다. 따라서 검색 대상 프레임을 한정 시킨다. 이를 위해서 객체의 움직임 정보를 사용한다. 움직임 정보는 시간변화에 따른 프레임 간 변화를 이용하며, 움직임이 있는 프레임만을 배경탐색의 대상으로 하였다. 그림 11은 시간차가 있는 프레임간 차연산을 통하여 움직임이 있는 영역을 나타낸 그림이다. 삭제 대상인 보행자가 검출된 위치 주변에서 움직임이 감지되지 않는다면 배경을 습득할 위치에 보행자가 존재한다고 판단하여 해당 프레임은 배경획득 대상에서 제외한다.

3.2.3 주변영역 검사



그림 12 배경 획득 시 주변영역 검사

배경 탐색은 검출기로부터 전달받은 보행자 검출 위치의 주변 영역까지 포함하여 탐색이 이뤄진다. 학습된 데이터는 보행자의 전신을 바탕으로 하기 때문에 신체의 일부분이 배경에 포함되더라도 이를 배경으로 판단하여 배경으로 획득 할 수 있기 때문이다. 그림 12는 삭제를 수행할 프레임에서 보행자가 검출된 영역을 전달받아 이웃프레임에서 배경을 획득하기 전 주변영역을 검사하는 모습이다. 중앙 영역과 8방향의 주변 영역을 검사한 결과 모든 방향에서 보행자가 검출되지 않는다면 중앙영역에 신체의 일부가 포함될 우려가 없어 이를 배경으로 습득한다.



그림 13 배경을 활용한 삭제 결과

그림 13은 배경을 획득하여 삭제 대상을 덮어씌움으로서 삭제를 수행한 결과이다. 삭제는 잘 이뤄 졌지만 원본 이미지에서 보행자 객체가 갖는 그림자와 명도의 변화로 인한 이질감이 느껴진다.

3.3 변경된 페더 블렌더(Modified Feather Blender)

앞서 언급한 이질감 해결하기 위하여 Feather Blender를 사용하고자 하였으나 사용한 결과 적용하고자 하는 대상의 이미지 크기가 너무 작아 원하는 효과를 얻기 어려웠다. 따라서 작은 이미지에서도 Feather Blender와 유사한 효과를 내는 변경된 페더 블렌더를 제시한다.

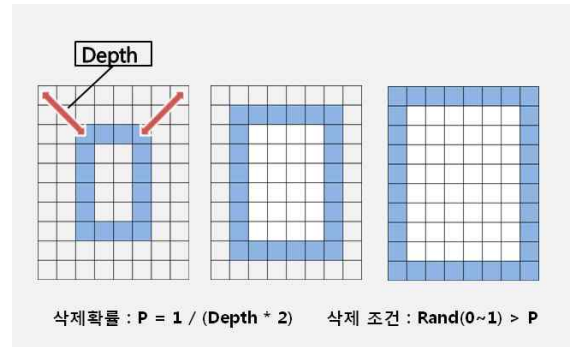


그림 14 변경된 페더 블렌더 연산과정

그림 14는 논문에서 제시하는 변경된 페더 블렌더의 연산과정으로 이미지의 일정 비율을 깊이(Depth) 값으로 하여 해당 깊이 값으로부터 내부에서 외부로 이동하며 픽셀의 삭제 확률을 늘려가는 방식이다.



그림 15 변경된 페더 블렌더 적용 결과

그림15는 변경된 페더 블렌더 연산을 적용한 결과이다. 그림16은 변경된 블렌더가 적용된 배경영역을 사용한 삭제 수행결과이다. 배경과 원본 영상의 경계면이 섞여 자연스럽게 보이도록 하였다.



그림 16 최종 연산 결과

IV. 실험 및 분석

실험에서는 본 논문에서 사용한 검출법과 다양한 검출법의 검출성능을 비교하여 보행자 검출에 있어 적합한 검출 방법인지에 대하여 알아본다. 그리고 삭제기의 각 과정의 산출 결과를 확인해 봄으로서, 각 처리기가 잘 수행되고 있는지 확인한다. 마지막으로 처리결과와 보행자가 없는 해당 영역간의 PSNR을 알아보려고 한다.

4.1 학습에 사용된 데이터

보행자 검출은 HOG특징과 Linear-SVM 분류기를 사용하여 학습하였다. 학습에 사용된 이미지 데이터의 크기는 64×128 이며 Positive(보행자) 2416장 Negative(보행자가 아닌 객체 또는 배경) 12190장을 사용했다. Positive 이미지는 INRIA Data Set을 사용했으며 Negative 이미지는 보행자가 아닌 무작위 이미지를 사용하였다. Positive 이미지에 비하여 Negative 이미지의 수가 상대적으로 많기 때문에 발생할 수 있는 학습의 편중을 막고, 긍정오류를 줄이기 위해 4회에 걸쳐 재학습을 진행하였다. 그림 17은 학습에 사용된 이미지 예시와 학습정보가 나타나있다.

POSITIVE : INRIA X 2416 SIZE : 64×128

Retraining
- 데이터양의 균형
- 검출 향상

NEGATIVE : RANDOM IMAGE X 12190 SIZE : 64×128

First set
Positive 2416 Images
Negative 2416 Images
Retraining x 3
3300 Images
3300 Images
3174 Images

그림 17 학습에 사용한 데이터

4.2 검출기 성능비교

본 논문에서는 기존에 많이 사용되었고 성능이 검증된 기술자인 HOG를 사용하여 실험 하였고 대부분의 경우 보행자 객체를 잘 찾아낸 것을 확인할 수 있었다. 하지만 HOG가 발표된 후 많은 시간이 지났고 HOG보다 향상된 성능을 보이는 기술자들이 발표되었다. 제안하는 삭제 알고리즘에 앞서 객체 검출이 선행되어야하기 때문에 이러한 기술자의 성능에 대한 비교실험이 필요하였고 기존연구가 많이 진행되어 있기 때문에 생략한다.[9]

4.3 Blender 적용

그림 18의 블렌더 적용 전후의 모습을 보여주고 있다. 삭제 대상영역 비교적 밝고 부드러운 질감일 경우 Blender 미적용 시 이질감이 느껴지는 모습을 볼 수 있었으며, 반면에 어둡고 거친 질감의 영상에선 이질감이 나타나지 않았다. 전자의 경우 Blender를 사용 시 이질감이 없어진 부드러운 결과영상을 획득 하였지만 후자의 경우 Blender 사용이 불필요 하였다.



(a)Blender 적용 전 (b)Blender 적용 후

그림 18 Blender 성능 비교

4.4 과정별 산출 결과

그림 4의 삭제기 구성도의 흐름에 따라 단계별로 의도한 연산 결과가 나오는지 확인해 보고자 한다. 실험을 위해 다양한 크기의 보행자 객체가 등장하는 프레임을 선택하였고 모든 보행자 객체를 하고 있다.

보행자가 아니지만 보행자라고 잘못 찾은 경우와 동일한 보행자를 중복해서 찾은 경우가 있다. 하지만 이러한 문제는 최종연산 결과에 영향을 끼치지 않는다.

그림 19는 학습된 정보를 바탕으로 보행자를 검출한 모습이다.



그림 19 검출기의 검출결과

배경탐색은 보행자가 검색된 위치의 주변 8방향에서 수행되어야 하며 그림 20은 이를 위하여 각 영역을 잘라온 모습이다.



그림 20 주변 영역 검색

획득한 배경을 바로 삭제에 사용하면 보행자 객체의 그림자 또는 명도 변화로 인하여 경계에서 이질감이 발생할 수 있다. 그림 21은 이를 해결하기 위해 제시한 변경된 페더 블렌더를 적용 시킨 모습이다.

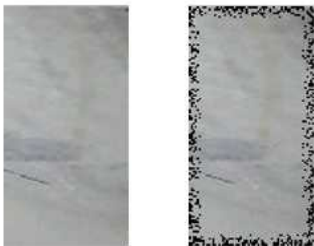


그림 21 획득한 배경과 변경된 Blender 적용

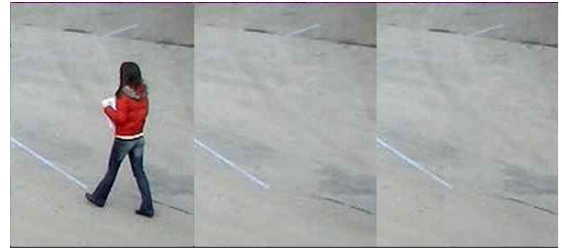


그림 22 최종 적용 결과

마지막으로 그림 22는 획득한 배경을 선택프레임에

적용시켜 객체를 삭제한 모습이다.

4.5 처리결과 PSNR



(a)삭제전 (b)획득배경 (c)삭제결과

그림 23 PSNR 실험 이미지

PSNR 수치는 그림 23의 삭제기를 사용하여 삭제한 결과(c)와 배경습득에 사용한 프레임에서의 동일영역(b)를 사용하여 산출하였다. 실험은 50개의 삭제결과를 바탕으로 하였다. (b)와 (c)의 PSNR값을 구하였고 이를 통하여 삭제결과가 객체가 없는 상황과 얼마나 유사한지를 알아보려고 하였다. PSNR값의 평균은 약 19.246로 측정되었다.

4.6 분석

실험결과 보행자 검출율과 속도 향상을 위한 기술자의 변화가 필요했으며, 본 논문에서 제시한 Blender는 명도가 낮고 구조가 복잡한 배경에서는 그 효과가 미미하여, 모든 삭제연산에 사용할 필요는 없었다. 하지만 이질감이 발생한 경우 사용하면 확실히 개선된 결과를 보여주고 있다. 검출기는 설계한 과정에 맞게 각 단계 별로 알맞은 수행결과를 보여줬으며 최종 처리결과 또한 일반 사람이 구별하기 어려운 정도로 우수하게 나타났다.

V. 결론 및 향후 연구방향

본 논문에서는 시중 제품의 수동적이고 숙련이 필요한 삭제과정을 자동화하기 위해서 스스로 보행자를 검출하고 이를 삭제하고 자연적 배경으로 대체하는 삭제기를 제안 하였다.

제안하는 방법은 학습된 결과를 바탕으로 원하는 영상내의 보행자를 검출하고 검출된 영역의 위치를 중심으로 시간 변화가 적은 프레임에서 배경을 습득한다.

습득한 배경을 제시한 블렌더로 처리하여 초기 검출영상에 덮어씌우는 방식으로 보행자 객체를 삭제한다. 구현된 삭제기를 이용하여 다양한 영상에서 특별한 작업과정 없이 보행자 삭제가 가능하였으며 질적인 면에서 수동 프로그램 이상의 결과물을 얻을 수 있었다.

전체 삭제기의 정확도는 기본적으로 객체 검출기의 성능에 직접적으로 영향을 받게 되며, 사용된 학습기와 학습데이터에 따라 결과가 달라진다. 또한 수동프로그램은 단일 영상으로 편집 가능하지만 제안하는 알고리즘은 시간변화에 따른 다중영상이 필요하다는 단점이 있다. 실험을 통하여 나타남바와 같이 Blender를 적용시킬 경우에 대한 판단기준이 필요하며, 학습정보에 따라 보행자 뿐 아닌 다양한 객체가 삭제 대상이 될 수 있기 때문에 이와 관련된 연구가 진행되어야 할 것이다.

REFERENCES

[1] ADOBE PHOTOSHOP 기능 설명서
 [2] ADOBE AfterEffects 기능 설명서
 [3] Pixlr Editor 기능 설명서
 [4] dotPDN LLC paint.net 기능 설명서
 [5] 김수진, 조정순, "HOG 특징 연산에 적용하기 위한 효율적인 기울기 방향 bin 및 가중치 연산 회로 설계", 『전자공학회논문지 제51권 11호』, 2014. pp66-72.
 [6] 황대규, "에지 강화 필터를 사용한 해상도 변화에 적응적인 보행자 검출 시스템", 2014
 [7] 박원재, 김대환, 여준기, 노태문, 고성재, "선택적 블록기반의 HOG-SVM을 이용한 향상된 속도의 보행자 검출", 『2011년도 대한전자공학회 추계종합학술대회』, 2011. pp608-611.
 [8] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005.
 [9] Rodrigo Benenson, Mohamed Omran, Bernt Schiele. "Ten Years of Pedestrian Detection, What Have We Learned?", 2014

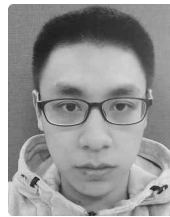
저 자 소 개



김창성(학생회원)

2009년 ~ 현재 : 전북대학교 컴퓨터 공학과 재학 중.

〈주관심분야 : 영상처리〉



이동석(학생회원)

2012년 : 전북대학교 전자공학과 학사 졸업.

2012년 ~ 현재 : 전북대학교 전자공학과 석사 재학 중.

〈주관심분야 : 컴퓨터비전, 인공지능〉



박동선(정회원)

1979년 : 고려대학교 전자공학과학사 졸업

1984년 : 미국Missouri대학 공학석사

1991년 : 미국Missouri대학 공학박사

1991년 ~ 현재 : 전북대학교 전자공학부 교수

〈주관심분야 : 신경망, 패턴인식, 영상 처리〉