

개선된 DOF 방식의 스마트 감시 카메라 설계 및 구현

이재구, 김병관, 고영웅
한림대학교 컴퓨터공학과
{dlworn8842, kwani, yuko}@hallym.ac.kr

Design and Implementation of Smart Camera Using Enhanced DOF Scheme

Jea-Gu Lee, Byung-kwan Kim, Young Woong Ko
Dept. of Computer Engineering, Hallym University

요 약

본 논문은 일반적인 사람들의 일상 생활에서 발생하는 상태 변화(넘어짐)를 조금 더 빠르게 대처 하기 위해 제안하는 스마트 감시 카메라를 이용한다. 제안하는 알고리즘은 기존의 DOF 알고리즘이 가지는 빛의 영향 및 노이즈로 인한 문제점을 보완하고자 이미지 블록화 방식을 적용한다. 개선된 DOF 방식을 적용하는 스마트 감시 카메라는 안전 및 보안용으로 사람의 움직임 및 넘어짐을 체크하고 그 결과를 관리자에게 전송하여 좀 더 빠른 이상 징후 탐지와 대처에 도움을 줄 수 있다.

1. 서론

최근 기술의 발달로 이미지 변화에 대한 다양한 탐지 및 분석 기술[1]이 존재한다. 이미지 자동 탐지와 분석작업[2, 3]은 매우 중요하며 다양한 산업 분야에서 지속적인 도전 과제이다. 이미지 자동 탐지와 분석은 모든 분야에서 사람의 수작업을 줄이는 것을 기본 목적으로 한다. Graph cuts 알고리즘은 반복적인 에너지 소비를 최소화하는 알고리즘[4]을 사용하여 이미지 분할 문제를 해결한다. 하지만, Graph cuts 알고리즘이 좋은 결과를 나타내는 반면, 복잡한 연산으로 성능 저하를 초래하기도 한다.

이진화 기법(image thresholding)[5]은 색을 반전 시킴으로 흑 또는 백의 색만을 가지게 되는 것을 말한다. 이전 이미지와 현재 이미지를 각각 이진화 처리한 다음 두 이미지에서 값이 다른 픽셀 들을 찾아서 변화 값을 구하는 방법이다. 그러나 이는 조명 및 색의 영향을 많이 받기 때문에 사람이 넘어지는 순간 같은 세밀한 부분에 적용 하기에는 부적합하다. 또한 연속되는 사진 촬영에 반복적인 이진화를 진행하여 픽셀 비교를 한다면 성능 저하를 초래 할 수도 있다.

헤드 트래킹(head tracking)[6]은 사람의 머리를 정확하고 빠르게 추적하면서 값을 추출해 내는 기술이다. 그러나 출력 시 노이즈로 인한 떨림과 같은 문제가 수반된다. 부드러운 출력을 위해서 노이즈 및 떨림을 제거하는 등의 과정을 거치게 되면 데이터 처리

와 영상측정 반응속도에서 느린 성능을 보이는 문제점이 있다.

DOF(Different of Frame)는 사용자가 설정한 영상 프레임을 현재 추출된 영상 프레임과 비교하여 변경된 부분의 픽셀을 나타내는 알고리즘[7, 8, 9]이다. DOF 는 빛의 변화 및 노이즈 등에 민감하여 불필요한 데이터 값을 추출하는 결과를 초래 하기도 한다.

본 연구에서는 앞에서 제시한 Graph cuts 의 복잡한 연산으로 인한 성능저하 문제, 이진화 기법의 미세한 색의 영향 그리고 헤드 트래킹의 노이즈 및 떨림을 제거할 때 나타나는 데이터 처리시간 및 반응속도 저하 등의 단점들을 해결하고자 기존에 DOF 알고리즘에 이미지 블록화를 적용 하는 방식을 제안한다. 제안하는 방식은 기존의 방식들에 비해 데이터 처리 과정도 매우 단순하고 빠르며 노이즈 및 미세한 색의 변화에도 영향을 적게 받아서 정확한 결과 값을 측정할 수 있다. 실험을 통하여 모든 방향에서 사람의 움직임 및 넘어짐을 확인하였고 빠른 데이터 처리 후 결과값을 분석하여 관리자에게 경고 메시지가 전송되어 제안하는 개선된 DOF 알고리즘이 유용하게 동작이 됨을 확인 하였다.

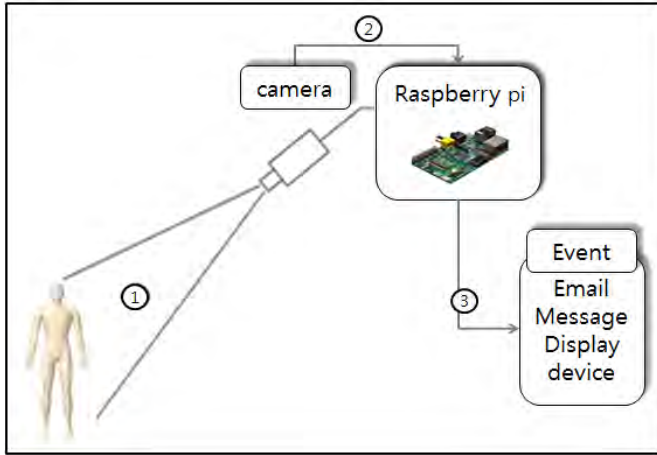
2. 스마트 감시 카메라 시스템

최근 보안용으로 감시 카메라의 사용이 많아지고 있으나 관리자가 계속 모니터링을 해야 한다거나 오직 감시용으로만 사용하는 등의 한가지 목적 및 기능으로만 카메라를 사용하는 경우가 많다. 제안하는 방식은 카메라를 이용하여 단순히 보안용이 아닌 안전 및 관리를 위한 스마트 감시 카메라 시스템을 구현하는 것이다. 스마트 감시 카메라 시스템은 소형 임베디드 보드와 일반 카메라를 사용하여 다른 제품 보다

이 논문은 2014 년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2014R1A2A1A11054160)

값이 저렴하고 설치하기 쉬운 아이템으로 병원, 요양 병원 및 특정 공공시설 또는 혼자 근무해야 하는 시설에서 환자 또는 일반 사람들이 넘어져서 일어나지 못하는 경우에 스마트 카메라가 모니터링을 하지 않는 관리자에게도 데이터를 전송하는 것을 목적으로 한다.

2.1 시스템 동작 과정



(그림 1) 시스템 동작 순서

① 카메라가 특정 장소를 계속 촬영한다. ② 촬영된 데이터는 소형 임베디드 보드로 전송된다. ③ 임베디드 보드에서 이전 사진과 현재 사진을 제안하는 개선된 DOF 알고리즘을 통해 비교해서 사람의 움직임, 넘어짐을 판단하여 사람이 넘어졌다면 Email 로 경고 알림을 전송한다.

```

Input: CompareFunction, CheckFallFunction
Input: initialImage, currentImage, checkImage
Output: WarningMessage
begin
    initialImage ← currentImage
    while CheckImage do
        // If Statement for Compare Image
        if isSameImage(CompareFunction) then
            Delete initialImage
            initialImage ← checkImage
        end
        else
            // Check Fall or Not
            FunctionCall CheckFallFunction
            return WarningMessage
        end
    end
end
    
```

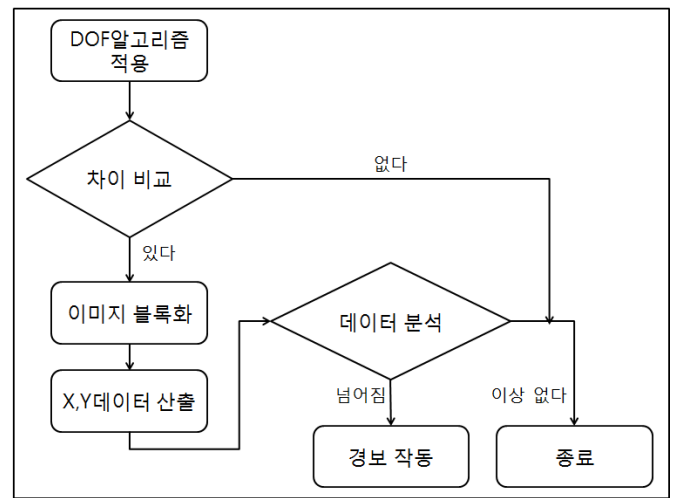
(그림 2) 시스템 동작 알고리즘

그림 2 에서 initialImage 값을 현재 이미지 값으로 초기화되어 지속적으로 발생하는 checkImage 값과 비교하여 이미지의 변화 여부를 확인하기 위해

CompareFunction 을 호출하여 판단한다. 비교를 통해 변화가 없을 경우에는 initialImage 값을 삭제하고 checkImage 값으로 대체한다. CompareFunction 에서 변화가 탐지되면 CheckFallFunction 을 호출하여 이미지 블록화를 적용한 다음 사람이 넘어졌는지를 판단하고 사람이 넘어졌다면 해당 이미지를 관리자에게 WarningMessage 로 전송하여 이상 징후를 알리게 된다.

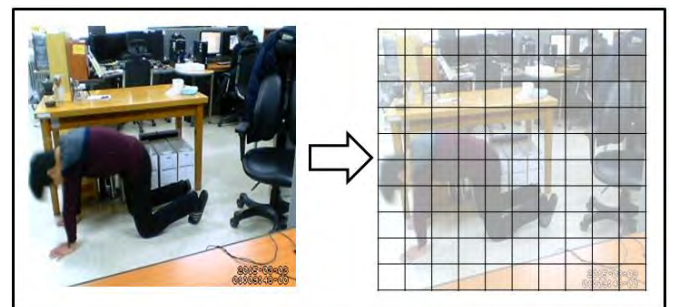
2.2 이미지 블록화

이미지 블록화는 DOF 알고리즘을 사용하여 이전 이미지와 현재 이미지에서 25%이상의 오차가 발생할 경우 이미지 블록화를 적용해서 블록들의 X, Y 값을 산출하여 그림 2 의 CheckFallFunction 을 연산할 때 사용된다.



(그림 3) 이미지 블록화 순서도

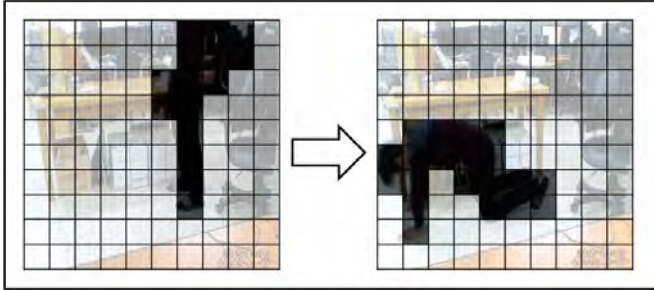
그림 3 은 DOF 알고리즘을 사용하여 두 개의 사진에서 데이터 차이가 있나 확인 한 후 이미지 블록화를 수행하여 인접한 블록들 중에서 가장 긴 블록집합의 최대 X, Y 값을 측정 및 연산하여 사람의 넘어짐을 확인하는 작업의 순서도이다.



(그림 4) 이미지 블록화

그림 4 는 DOF 알고리즘을 적용 한 다음에 이미지 블록화를 실시한 그림이다. 단순한 픽셀 비교로는 넘어짐 현상을 판단하기 어렵기 때문에 10X10 의 이미지 블록화를 적용한다. 이전 사진과 현재 사진 모두 블록화를 수행한 후 두 사진의 차이가 나는 부분의

블록들만 선택 한다. 선택하는 방식은 각 블록에 포함된 전체 픽셀 중 50% 이상의 오차가 발생한 블록만을 선택한다. 그렇게 선택된 블록들 중에서 모든 블록의 길이를 측정하는 것이 아니라 서로 인접한 블록들만 측정을 한다.



(그림 5) 인접한 블록의 최대 X와 Y 값 추출

그림 5 은 이전 사진과 현재 사진에 이미지 블록화를 수행한 후 인접한 블록들만 선택한 상태이다. 서로 인접한 블록들 중에서 연결길이 값이 가장 높은 그룹의 총 X, Y 값을 구하여 사람이 넘어졌는지를 분석할 수 있다. X, Y 데이터를 추출할 때 서로 인접한 블록들만 측정하며 가장 값이 높은 데이터만 연산하는 이유는 주변 환경의 간섭을 차단하고 정확한 결과 값을 연산하기 위함이다. 예상하지 못한 빛의 간섭이나 사물의 움직임이 있다면 이미지 블록화를 적용했을 시 넘어진 사람 보다 변화량이 작아서 사람을 기준으로 데이터를 정확히 추출할 수 있다. 또한 이미지 블록화를 적용 함으로써 어느 방향으로 넘어져도 판별이 가능하다.

3. 실험 환경 및 결과

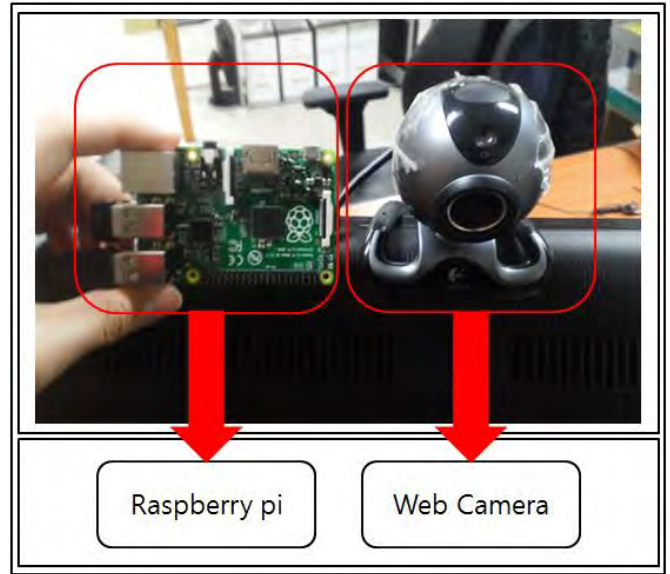
3.1 실험 환경

제안하는 실험은 다수의 인원이 공동으로 작업하는 연구실에서 다양한 상황 및 환경을 조성하여 직접 실험을 하였다.

<표 1> 실험 환경

하드웨어	Raspberry pi model B(보드) Logitech Quickcam Pro 5000(웹캠)
운영체제	Rasbian 3.12.28
그래픽카드	Broadcom VideoCoreIV
CPU	700MHz ARM11
Memory	512MB
개발언어	C, opencv

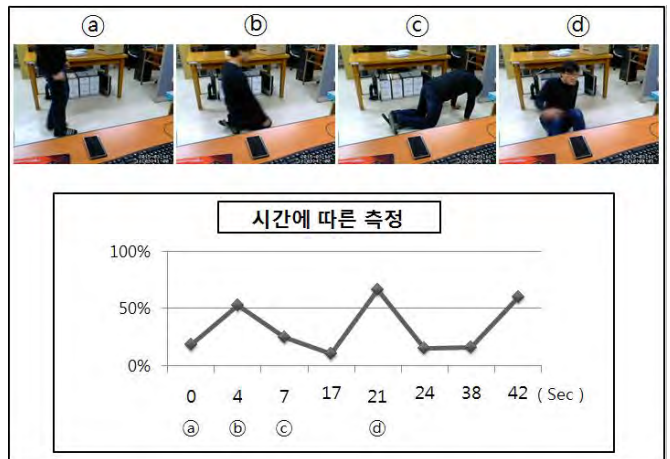
표 1 은 본 논문에서 제안한 방법을 구현한 실험 환경이다. 라즈베리 파이 모델 B 를 이용하여 C 언어로 구현하였고 이미지 픽셀 비교를 위한 연산은 opencv 라이브러리를 사용하였다. 이미지 촬영은 일반 웹 카메라를 사용하였다.



(그림 6) 실제 실험 장비

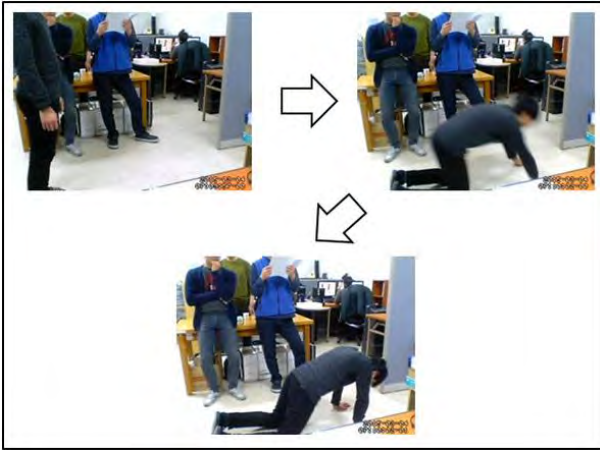
그림 6 은 실제로 실험에 사용된 장비를 나타낸다. 왼쪽은 Raspberry pi 라는 소형 임베디드 보드이고 오른쪽은 일반 웹 카메라이다. 실험 에서 사용되는 웹 카메라뿐만 아니라 라즈베리 파이에 호환이 되는 모든 종류의 카메라를 사용한다면 동일한 결과 값이 나온다.

3.2 실험 결과



(그림 7) 넘어짐 체크 실험 결과

그림 7 은 실제로 사람이 다양한 행동을 반복하면서 움직인 실험 결과이다. 데이터①은 이전 이미지와 현재 이미지와의 오차 값이 25%이하로 이미지 블록화를 하지 않는 것을 나타낸다. 25%이상의 데이터 ②, ③, ④을 보면 이미지 블록화가 적용되어 사람의 넘어짐 체크가 정상으로 작동된 것을 볼 수 있다. 또한 데이터①의 다음 이벤트인 7 초 때를 확인해 보면 27%의 결과 값을 가지고 있는 데이터③을 볼 수 있다. 이미지 블록화를 적용하면 데이터③과 같이 낮은 결과 값을 가진 데이터도 정확하게 넘어짐을 체크 할 수 있다.



(그림 8) 다수의 실험자가 있는 환경에서 실험

그림 8 은 방 안에 다수의 사람이 있는 상황을 가정하여 실험한 모습을 보인다. 제안하는 기술은 움직이는 사람이 1 명일 경우에는 정확한 측정이 가능하지만 2 명 이상일 경우에는 측정 값이 정확하게 나오지 않는다.

4. 결 론

본 논문에서는 일반적으로 많이 쓰이는 카메라를 이용하여 안전 및 관리용 스마트 감시 카메라 시스템을 구현하였다. 기존의 DOF 알고리즘이 가지는 단점을 보완하고자 이미지 블러킹 기법을 제안하고 구현하였다. 이를 통해 요양 병원 및 일반 병원, 혼자 근무 해야 하는 장소, 사람의 인적이 드문 골목길 등에서 사람들이 낙상을 당하는 경우 경고 메시지를 전송하여 관리자가 빠르게 인식하는 것이 가능하게 되었다. 하지만 특정 공간에 혼자 있는 사람을 대상으로 구현하였기에 다수의 사람이 움직이고 있는 경우는 확인에 어려움이 있다. 향후 실험 및 보완을 통해 다수의 사람이 활동하고 있는 환경에서도 정확한 결과를 나타내도록 할 것이다.

참고문헌

[1] Radke, Richard J., et al. "Image change detection algorithms: a systematic survey." *Image Processing, IEEE Transactions on* 14.3: 294-307, 2005.

[2] Herbon, Christopher, Klaus Tönnies, and Bernd Stock. "Detection and segmentation of clustered objects by using iterative classification, segmentation, and Gaussian mixture models and application to wood log detection." *Pattern Recognition*. Springer International Publishing, 354-364, 2014.

[3] 이한성, et al. "HoG 특징 추출과 SRC 분류기를 이용한사람탐지." *한국정보과학회학술발표논문집*, 39.2 B: 222-224, 2012.

[4] 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.

[5] İlsever, Murat, and Cem Ünsalan. "Pixel-based change detection methods." *Two-Dimensional Change Detection Methods*. Springer London, 7-21, 2012.

[6] Rougier, Caroline, et al. "Monocular 3D head tracking to detect falls of elderly people." *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*. IEEE, 2006.

[7] Vezhnevets, Vladimir, Vassili Sazonov, and Alla Andreeva. "A survey on pixel-based skin color detection techniques." *Proc. Graphicon*. Vol. 3. 2003.

[8] Huerta, Ivan, et al. "Detection and removal of chromatic moving shadows in surveillance scenarios." *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009.

[9] 서영철, 이병수, "보안 카메라를 이용한 반려 동물 분포도 및 행동 분석", *한국 소프트웨어공학회지* 제 16 권 1 호, KCSE, 2014.