

영상정보만을 이용한 사람과 로봇간 실시간 상대위치 추정 알고리즘

이정욱* · 선주영** · 원문철***†

* 삼성전자, ** 충남대학교 메카트로닉스공학과

Real-Time Algorithm for Relative Position Estimation Between Person and Robot Using a Monocular Camera

Jung Uk Lee*, Ju Young Sun** and Mooncheol Won***†

* Samsung Electroics, ** Dept. of Mechatronics Engineering, Chungnam Nat'l Univ.

(Received September 24 2012 ; Revised October 29, 2013 ; Accepted November 7, 2013)

Key Words: Robot Vision(로봇 비전), Histogram of Oriented Gradient(기울기 히스토그램), Support Vector Machine(서포트 벡터 머신), CUDA(쿠다)

초록: 본 논문에서는 단안 카메라를 이용하여 사람과 로봇(카메라)간의 상대위치를 실시간으로 추정하는 알고리즘을 제안한다. HOG(기울기 히스토그램) 특징벡터와 SVM(서포트 벡터 머신) 분류기를 이용하여 사람의 두부 및 어깨영역을 검출한다. 검출된 영역의 크기와 위치를 이용하여 사람과 로봇(카메라)간의 상대 위치 및 각도를 계산한다. 또한 알고리즘 수행속도를 향상시키기 위하여 본 논문에서는 NVIDIA의 GPU와 CUDA 라이브러리를 사용하였다. 그 결과 알고리즘 수행속도는 초당 15 프레임의 영상데이터를 처리할 수 있다. 알고리즘의 정확도 비교를 위해서 SICK 레이저 스캐너 출력과 비교하였다.

Abstract: In this paper, we propose a real-time algorithm for estimating the relative position of a person with respect to a robot (camera) using a monocular camera. The algorithm detects the head and shoulder regions of a person using HOG (Histogram of Oriented Gradient) feature vectors and an SVM (Support Vector Machine) classifier. The size and location of the detected area are used for calculating the relative distance and angle between the person and the camera on a robot. To increase the speed of the algorithm, we use a GPU and NVIDIA's CUDA library; the resulting algorithm speed is ~ 15 Hz. The accuracy of the algorithm is compared with the output of a SICK laser scanner.

1. 서론

최근 카메라를 이용한 사람 검출 기술은 자동 감시 시스템, 서비스 로봇 분야, 인간과 로봇간의 상호작용 등 많은 응용시스템에서 적용되어 중요성이 더욱 증가하고 있다. 사람 검출을 위한 영상 획득 방법에는 카메라 한 대를 이용한 단일 영상획득 방법과 두 개 이상의 카메라를 이용하여 얻은 스테레오 영상으로 영상의 깊이 차를 이용하는 방법이 있다.⁽¹⁾ 스테레오 카메라는 카메라와 사람의 깊이(Depth) 정보를 제공해준다는 장점이 있지만 단안 카메라에 비해 상대적으로 고가

이며 깊이 정보도 노이즈에 의해 정확한 값의 계산이 어렵다. 실제 사람도 한쪽 눈의 정보만으로도 근거리에서 앞 사람을 일정한 거리를 두고 쫓아가는데 문제가 없다. 본 논문에서는 단안 카메라를 이용한 사람과 로봇(카메라)간의 상대 거리 및 각도를 추정하는 방법을 제안하고자 한다.

또한 본 논문에서는 사람 검출 시 기존의 사람 몸 전체를 검출하는 방법⁽²⁾ 대신 사람의 두부 및 어깨만을 검출하는 방법을 사용한다. 이 방법을 쓸 경우 Fig. 1과 같이 사람 몸 전체를 검출하기 위해서는 카메라 시야에 사람 몸 전체가 나와야 하지만 사람 두부 및 어깨 검출 시에는 카메라 시야에 사람의 두부와 어깨만 나와도 되기에 사람 몸 전체를 검출할 때보다 근거리에서도

† Corresponding Author, mcwon@cnu.ac.kr

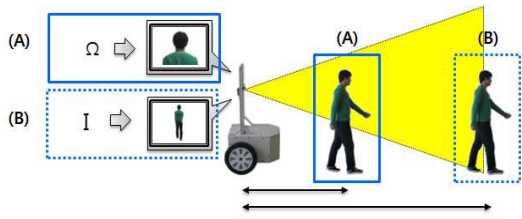
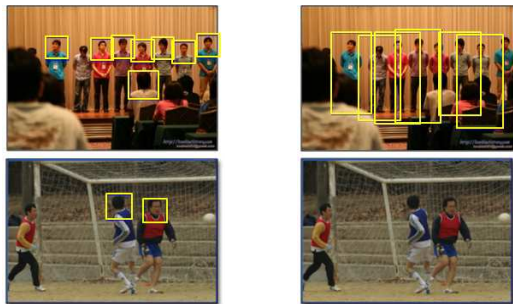


Fig. 1 Advantage of head and shoulder area detection



(a) Examples of head and shoulder detection (b) An failed examples of whole body detection

Fig. 2 Comparison of head and shoulder detection and whole body detection

(0.5m에서 1m정도) 검출할 수 있다는 장점을 가지고 있다. 또한 I자 형태를 가진 사람 몸 전체를 검출할 때에는 사람의 동작(손/발)에 대한 영향을 많이 받지만, Ω자 형태를 가진 사람 두부 및 어깨 검출에는 사람의 손발 동작의 영향을 적게 받는다는 장점을 갖는다(Fig. 2).

현재 컴퓨터 비전에서 보행자 검출 방법으로는 보행자 외형 및 움직임을 이용하는 외형 기반 방법⁽³⁾과 특징을 추출하여 보행자를 검출하는 특징 기반 방법^(4,5)이 있다. 외형 기반 방법은 보행자의 움직임, 의복, 및 자세와 배경 등 다양한 외부 조건에 영향을 많이 받는다.⁽⁶⁾ 따라서 본 논문에서는 입력 영상에서 특징을 추출하여 실시간으로 보행자를 추출하는 방법을 사용하였다. 특징 추출 방법으로 Haar Wavelet 기반 방법,⁽⁷⁾ Gabor Filter 방법,⁽⁸⁾ LRF(Local Receptive Field) 방법, HOG(Histogram of Oriented Gradient) 방법 등이 있고, 분류기에는 AdaBoost,⁽⁹⁾ 신경망, 트리 분류기, SVM(Support Vector Machine) 등이 많이 사용하고 있다. 본 논문에서는 특징 기반 방식 중 HOG를 이용하여 특징 벡터를 추출하고, 분류기로는 SVM 방법을 사용하였다.⁽¹⁰⁾ HOG 특징 벡터 추출 방법은 연산량이 많아 CPU 기반의 프로그래밍으로는 실시간 연산처리가 불가능하므로, 병

렬 프로그래밍을 지원하는 NVIDIA사의 CUDA를 사용하여 초당 15 프레임 내외의 고속 연산 프로그램을 하였다.

사람과 로봇(카메라)간의 상대 거리 및 각도를 추정하는 알고리즘 정확도 비교를 위해서 SICK 레이저 스캐너 출력과 비교하였다.

본 논문의 2장에서는 HOG와 SVM을 이용한 두부/어깨 검출 알고리즘의 전체 개요를 설명한다. 3장에서는 2장에서 설명한 두부/어깨 검출 알고리즘을 이용한 SVM 학습에 대한 구체적 내용과 영상에서 실시간으로 사람을 검출한 결과에 대해서 설명한다. 4장에서는 카메라와 사람간 거리 및 각도 추정 방법에 대해서 서술한다. 5장에서는 4장에 설명한 거리 및 각도 추정 방법에 대한 실험결과를 나타내고 있다.

2. 두부/어깨 검출 알고리즘 개요

본 연구에서는 Dalal과 Triggs가 보행자 전신 검출을 위하여 제안한 HOG 알고리즘⁽²⁾을 응용하여 사람의 두부/어깨 영역을 검출하였다. HOG 은 여러 개의 Positive image와 Negative image로부터 추출된 HOG 특징 벡터를 가지고 SVM 파라미터를 학습하는 단계(Learning Phase)와 입력 영상내의 다양한 크기 및 위치를 가지는 부분 이미지의 HOG 특징 벡터와 학습된 SVM을 이용하여 객체를 검출하는 검출 단계(Detection Phase)로 이루어져 있다(Fig. 3). 본 연구에서도 기존 알고리즘과 동일한 학습/검출 단계를 이용한다. 단 차이점으로는 Positive image로는 사람 전신 이미지가 아닌 두부/어깨 영역 이미지를 사용하고, Negative image는 두부/어깨 영역이 아닌 일반이미지(Negative image)를 사용한 점이다.

검출 단계에서 입력 영상내의 다양한 크기 및 위치의 부분 이미지에 대해서 HOG 특징 벡터를 추출하여 사람의 두부 및 어깨를 검출하기 때문에 Fig. 4(a)와 같이 다양한 크기와 위치를 가지는 검출 윈도우가 생성된다. 하지만 우리가 최종적으로 원하는 결과는 Fig. 4(b)처럼 사람 두부/어깨 영역 근처의 여러 개의 중첩된 검출 윈도우가 아닌 각 객체(사람의 두부 및 어깨)를 대표하는 단일 검출 윈도우이다. 그래서 사람의 두부/어깨 영역 검출 과정의 마지막 과정으로 KDE(Kernel Density Estimation)을 응용한 Non-Maximum Suppression(NMS) 알고리즘을 사용하여 최종 검출 윈도우를 검출⁽²⁾하였다.

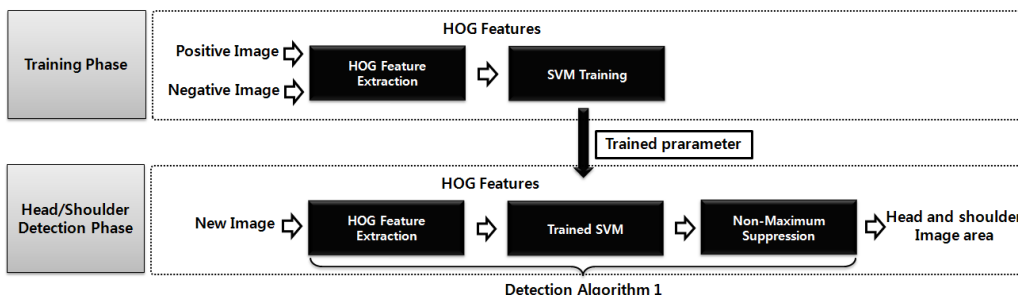


Fig. 3 Overview of the HOG detection algorithm



(a) Before implementing NMS (b) After implementing NMS

Fig. 4 Before and after implementing Non-Maximum Suppression(NMS)

2.1 제안하는 두부/어깨 검출 알고리즘

본 연구에서는 두부/어깨 영역 인식률을 높이기 위해 기존⁽²⁾의 HOG detection algorithm 수행 이후 Trained SVM-2로 한 번 더 검출하는 Dual-SVM detection algorithm(D-SVM)을 제안한다.

D-SVM은 항상 Trained SVM-2을 가지고 다시 한번 검출하지 않는다. 이전에 두부/어깨 검출 영역에 제한하여 이전 검출된 영역의 1.5배 영역에 대해서 Trained SVM-2을 이용하여 다시 한 번 검출한다. D-SVM을 수행하기 위해서 Trained SVM-1으로 검출된 윈도우 위치와 크기 값을 Buffer에 저장한다. 이 때 Buffer는 바로 이전 5 프레임에 대해서만 저장한다. SVM-1으로 두부/어깨를 검출하지 못하면 Queue-buffer에 저장되어 있는 데이터를 확인한다. 이때 이전에 검출된 영역이 있으면 이전 검출된 윈도우영역의 1.5배 영역에 대하여 Trained SVM-2을 이용하여 다시 한 번 검출한다. 이 때 Trained SVM-1과 Trained SVM-2의 차이점은 Table 1과 같이 객체 검출 조건과 탐색 영역 조건이 다르다. Trained SVM-1에서 Positive Image와 Negative Image을 판별하는 SVM Score 임계값이 1이라고 하면 Trained SVM-2에서는 SVM Score 임계값이 0으로 설정되어 있다. Trained SVM-2에서 SVM Score 값이 0 과 1사이의 값을 갖는 거짓 참 이미지가 검출된

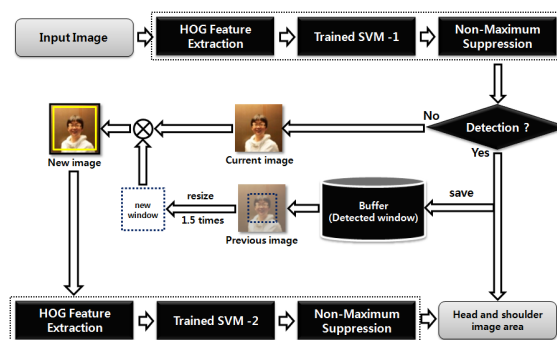


Fig. 5 Overview of Dual HOG detection method

확률을 가지고 있지만 탐색 영역 조건을 전 프레임에서 검출된 영역 크기의 1.5배로 정해 거짓 참 이미지 인식률을 감소시켰다.

3. 두부/어깨 검출 알고리즘 적용 실험

본 논문에서 실험을 위한 프로그램 및 알고리즘은 Window XP 환경에서 MFC를 이용하여 UI 프로그램을 구현하였다. 카메라에서 영상 획득 부분은 OpenCV를 이용하고, 고속 연산처리를 위하여 NVIDIA사에서 지원하는 CUDA을 이용하였다. CUDA⁽¹¹⁾는 그래픽카드를 이용한 GPGPU (General Purpose Graphics Processing Unit)⁽¹²⁾의 통합 개발 환경 제공을 목적으로 한다. 이전의 그래픽 카드는 3D 처리를 위해 다양한 기능이 추가되었지만 모두 그래픽 처리를 위한 것으로 범용적인 용도로 쓰이는 것들이 아니었다. 이러한 문제를 해결하기 위해 NVIDIA사는 기존의 Geforce 시리즈를 그래픽 전용처리를 위한 구조에서 유연성을 향상시켜 프로그램을 처리할 수 있도록 변경하였다. 또한 행렬 연산은 Blite++ 라이브러리⁽¹³⁾을 사용하여 NMS 알고리즘 코드를 구현하였다. 알고리즘이 수행하는 하드웨어 사양은 CPU는 Intel(R) Core(TM)2 Quad 2.33GHz를 사용하고, RAM 메모

Table 1 Differences between trained SVM-1 and trained SVM-2

Difference	Trained SVM - 1	Trained SVM - 2
Object detection condition	SVM Score is more than 1	SVM Score is more than 0
Search domain condition	All domain of input image	1.5-scaled image of previously detected frame



(a) Positive Example Image (b) Negative Example Image
Fig. 6 Examples of data Set (Image)

리는 2GB이다. 특히 고속 연산을 위하여 사용한 그래픽 카드는 Geforce GTX 560 ti를 사용했다.

3.1 SVM 파라미터 학습

본 논문에서 HOG를 사용하여 생성된 특징 벡터를 SVM 분류기를 사용하여 객체와 비객체를 구분하기 위해서는 미리 SVM 분류기를 학습해야 한다. SVM 분류기 학습을 하기 위해서는 Data Set인 긍정 예 이미지(Positive example image)와 부정 예 이미지(Negative example image)가 필요하다. 이 때 사용한 학습 이미지는 INRIA에서 지원하는 이미지와 자체 실험실에서 촬영한 이미지를 사용하였다. 먼저 긍정 예 이미지 생성과정은 64x64크기의 사람 두부와 어깨가 들어 있는 이미지를 추출한다. 이 때 사람의 몸은 좌우로 대칭되어 있는 점을 이용하여 추출한 이미지 한장을 좌우 대칭하여 두 개의 긍정 예 이미지를 생성한다. 그래서 본 논문에서는 2940개의 긍정 예 이미지를 제작하여 사용했다. 다음으로 부정 예 이미지 생성과정은 먼저 사람이 없는 이미지와 사람 두부 및 어깨 부분만 없는 이미지를 가지고 윈도우 크기와 중심 좌표값을 무작위로 생성하여 부정 예 이미지를 생성한다. 하나의 이미지 당 64x64 크기를 가지는 10개의 부정 예 이미지를 생성한다. 따라서 본 논문에서는 이미지 1562개를 가지고 총 15620개의 부정 예 이미지를 제작하여 사용하였다. Fig. 6은 사용된 긍정 및 부정 이미지들의 예를 나타낸다.

3.2 두부/어깨 부위 인식 결과

Fig. 7은 3절에서 제안한 사람 두부 및 어깨 검



Fig. 7 Head and shoulder recognition result of proposed algorithm

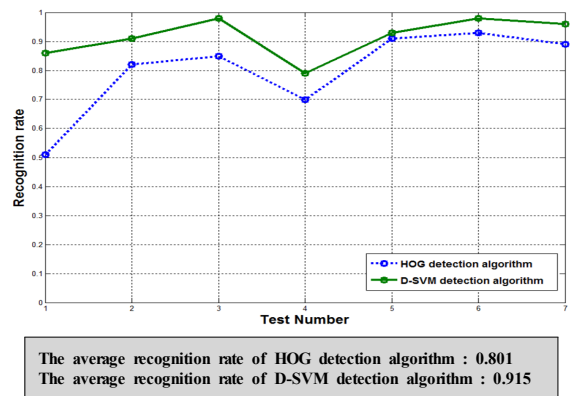


Fig. 8 Comparison of HOG detection algorithm to D-SVM detection algorithm

출 방법인 D-SVM Detection algorithm으로 HOG와 SVM 분류기를 사용하여 사람 두부 및 어깨를 인식한 결과이다. Fig. 7(a)의 참 긍정 이미지를 보면 여러 명의 사람이 있어도 다중 검출이 되는 것을 볼 수 있고, 특히 사람의 손/발 동작에 영향을 적게 받는 것을 볼 수 있다. 하지만 Fig. 7(b)의 거짓 긍정 이미지처럼 사람 두부 및 어깨만 나와 있는 광고판에 대해서는 잘못된 검출이 발생한다.

또한 Ω모양을 가진 객체에서도 잘못된 검출이 발생하는 것을 볼 수 있다. 마지막으로 Fig. 7(c)는 거짓 부정 이미지로 모자를 쓴 사람이나, 헤어스타일, 사람의 목이 보이지 않는 영상에 대하여 인식률이 떨어지는 것을 볼 수 있다.

3.2.1 HOG 알고리즘과 D-SVM 알고리즘의 결과 비교

Fig. 8은 실험실과 자연채광이 되는 지하 로비에서 촬영한 7개의 동영상을 가지고 3절에서 설



(a) HOG detection algorithm (b) D-SVM detection algorithm

Fig. 9 Examples of HOG detection algorithm and D-SVM detection algorithm

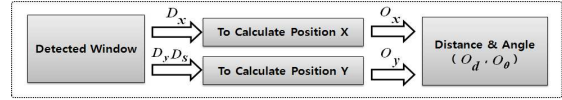
명한 HOG 알고리즘과 D-SVM 알고리즘에 대하여 인식률을 비교한 그래프이다. 특히 D-SVM 알고리즘이 HOG 알고리즘보다 우수한 점은 Fig. 9와 같이 초반에 Trained SVM-1로 사람의 두부 및 어깨를 검출하면 그 이후 사람 두부 및 어깨 영역에 손동작 같은 다른 물체가 들어와도 HOG 알고리즘보다 사람의 두부 및 어깨를 잘 검출되는 장점을 가지고 있다.

4. 카메라와 사람 간의 거리 및 각도 추정 방법

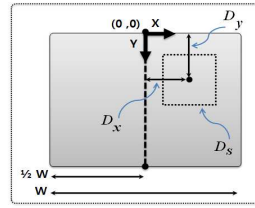
D-SVM 알고리즘으로 얻어진 영상내의 윈도우(사람 두부/어깨 영역) 정보로부터 카메라(로봇)와 사람간 거리 및 각도를 추정하는 방법을 설명한다. 본 논문에서 카메라와 사람간 거리 및 각도를 추정하는 방법은 Fig. 10(a)와 같다. 우리가 구하고자 하는 카메라와 사람 간의 거리(O_d) 및 각도(O_θ)은 카메라와 사람 간의 X축 거리 (O_x)와 카메라와 사람 간의 Y축 거리(O_y)을 가지고 구한다(Fig. 10(c)). 이때 O_x 값은 카메라 이미지 평면상에 검출된 윈도우의 중심 X축 거리(D_x)를 가지고 구하고, O_y 은 검출된 윈도우의 Y축 거리(D_y)과 윈도우 크기 비율(D_s)를 가지고 구한다.

4.1 카메라와 사람 간의 Y축 거리 추정함수

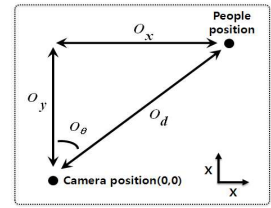
카메라와 사람 간의 Y축 거리(O_y)와 검출된 윈도우에서 얻은 D_y 와 D_s 간의 상관 관계를 알아보기 위해 키가 171/172/174/178 cm인 사람들을 대상으로 실험해 본 결과 Fig. 11의 그래프가 도출되었고 실험에 사용한 O_y 와 D_y 의 평균값을 최소



(A) System flow of estimated the distance and angle via detected window



(B) Image plane



(C) Camera coordinate systems

Fig. 10 Estimation Method of the distance and angle between camera and person via detected window

제곱법을 이용하여 식 (1)과 같은 4차 추정 함수를 구한다.

$$O_{estY}(D_y) = a_1(D_y)^4 + b_1(D_y)^3 + c_1(D_y)^2 + d_1D_y + e_1 \quad (1)$$

또한 O_y 와 D_s 도 실험을 통해 Fig.12의 그래프와 같이 나왔고, 식 (1)을 구한 동일한 방법을 이용하여 식 (2)와 같은 4차 추정 함수를 구한다.

$$O_{estS}(D_s) = a_2(D_s)^4 + b_2(D_s)^3 + c_2(D_s)^2 + d_2D_s + e_2 \quad (2)$$

카메라와 사람 간의 Y축 거리(O_y)를 윈도우 중심 Y축 거리(D_y) 또는 윈도우 크기 비율(D_s)만을 사용하여 구할 수도 있지만, 같은 키를 가진 사람이라도 두부와 어깨의 크기가 다르므로 D_y 와 D_s 관계를 같이 사용하는 식 (3)의 추정함수를 구한다.

$$O_y(D_y, D_s) = (1-\alpha)O_{estY}(D_y) + \alpha O_{estS}(D_s) \quad (3)$$

식 (3)의 추정함수 가중치 값(α)을 0부터 1까지 0.1간격으로 바꾸면서 카메라와 사람의 Y 좌표값 (O_y)의 에러값을 구한 결과 Fig 13의 그래프와 같다. D_y 범위가 120~190일 때는 α 가 0.3, 0.4, 0.5 순서로 가장 작은 에러 값을 가지는 것을 볼 수 있다. 따라서 식 (3)을 사용할 때 가중치 값(α)을 상수 값으로 사용하지 않고, D_y 의 크기에 따

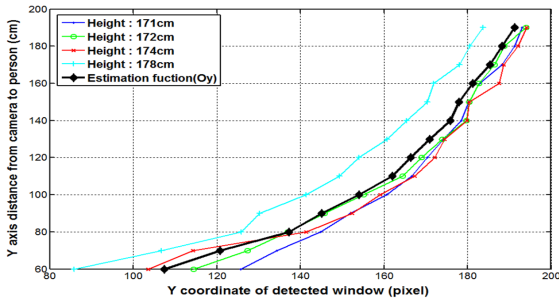


Fig. 11 Relation between Y coordinate of the detected window and distance from camera to person

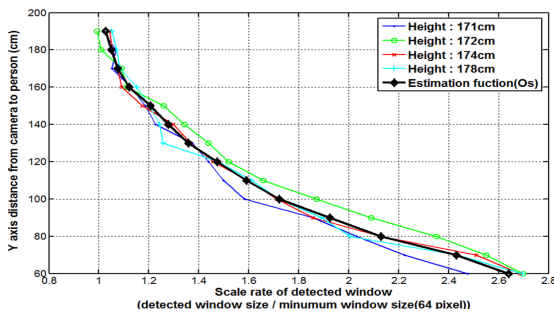


Fig. 12 Relation between scale rate of the detected window and distance from camera to person

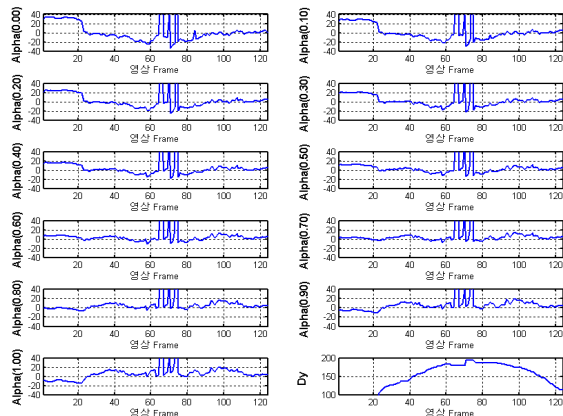


Fig. 13 Relation between-error of equation (4) and weight value

라 α 를 바꿀 수 있는 식 (4)를 사용한다.

$$\alpha(D_y) = (D_y - 100) / 255 + 0.2 \quad (4)$$

4.2 카메라와 사람 간의 X축 거리 추정함수

카메라와 사람 간의 X축 거리(O_x)와 검출된 윈도우에서 얻은 D_x 간의 상호관계를 알아보기 위해 키가 171/172/174/178cm인 사람들을 대상으로 실험해 본 결과 Fig. 14의 그래프가 도출되었다. Fig. 14의 그래프는 카메라와 사람 간의 Y축

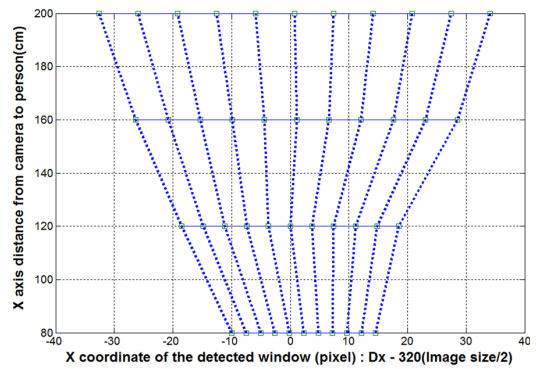


Fig. 14 Relations between X coordinate of the detected window and X axis distance from camera to person

거리(D_x)가 80, 120, 160, 200cm으로 유지한 상태에 카메라와 사람간의 X축 거리를 왼쪽 50cm(-50)부터 오른쪽 50cm(+50)사이를 10cm 간격으로 사람이 서 있는 상태에 D-SVM 알고리즘으로 검출된 윈도우 중심 X축 거리(O_x)를 나타낸 것이다.

Fig. 14의 그래프는 보면 카메라와 사람 간의 Y축 거리(O_y)는 같고 카메라와 사람의 X축 거리(O_x)을 카메라의 왼쪽에서 오른쪽으로 일정한 간격(10cm)로 이동시 검출된 윈도우 중심 X축 거리(D_x)도 왼쪽에서 오른쪽으로 일정한 간격으로 이동하므로 식 (5)의 상관 관계를 구할 수 있다.

$$O_x \propto D_x \quad (5)$$

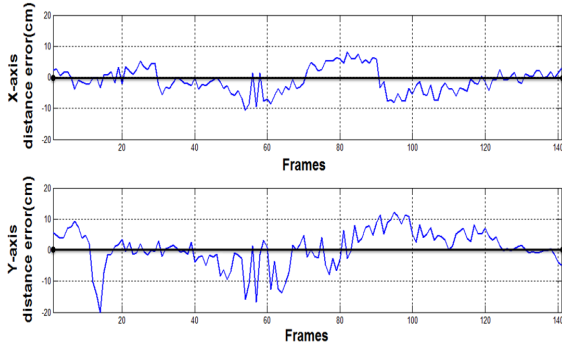
그리고 카메라와 사람 간의 Y축 거리(O_y)가 멀어질수록 검출된 윈도우의 중심 X축 거리 변화율(ΔD_x)은 커지므로 식 (6)의 상관 관계를 구할 수 있다.

$$\Delta D_x \propto O_y \quad (6)$$

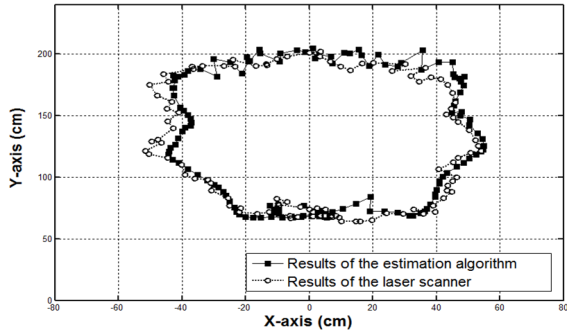
카메라와 사람 X축 거리(O_x)는 검출된 윈도우 중심 X축 거리(D_x)와 검출된 윈도우 중심 X축 거리 변화율(ΔD_x)을 가지고 식 (7)의 추정 함수를 구할 수 있다.

$$O_x(D_x) = \Delta D_x(O_y) \times D_x \quad (7)$$

최종적으로 4.1절과 4.2절의 설명으로 구한 카메라와 사람 간의 X축 거리(O_x)와 카메라와 사람 Y축 거리(O_y)을 이용하여 카메라와 사람 간의 거리(O_d)은 식 (8)을 통해 구하고, 카메라와



(a) X/Y axis distance error between laser scanner and estimation algorithm



(b) Routing garph of laser scanner and estimation algorithm

Fig. 15 Experiment result on distance and angle from camera to person

사람 간의 각도(O_θ)는 식 (9)를 통해 구한다.

$$O_d(O_x, O_y) = \sqrt{(O_x)^2 + (O_y)^2} \quad (8)$$

$$O_\theta(O_x, O_y) = atan(O_y/O_x) \quad (9)$$

5. 카메라와 사람간 거리 및 각도 추정 실험결과

Fig. 15는 사람이 실험실에서 원을 그리면서 이동하는 경우 4장에서 설명한 카메라와 사람간의 거리 및 각도 추정 알고리즘을 검증한 결과를 나타낸다. 참값으로 설정한 레이저 스캐너의 측정 데이터와 추정 알고리즘의 출력 X/Y축 위치에 대한 오차 값(a)과 이동경로를 출력한 그래프(b)이다. 전체적으로 X축 위치에 대한 오차 값은 10cm 이내이지만, Y축 위치에 대한 오차 값은 20cm 이내인 것을 볼 수 있다. 전반적으로 중간정도 거리에서는 오차 값이 작지만, 카메라와 사람의 거리가 멀거나 거리가 아주 가까운 경우, 오차 값이 커지는 것을 볼 수 있다.



Image Size	CPU based	GPU based
320x240	0.5s	0.031s
640x480	2.3s	0.068s

Fig. 16 Comparison of HOG feature vector running times between CPU based and GPU based

6. CUDA을 이용한 연산 속도 향상 결과

HOG를 이용하여 특징 벡터를 추출할 때는 많은 연산량이 필요하다. 본 논문에서 사용한 입력 영상 크기는 640x480, 윈도우 크기는 64x64, 이미지 피라미드 단계는 64단계로 했을 때 한 프레임 영상마다 추출해야 하는 특징 벡터 개수는 총 234,084개이다. 이는 CPU 기반으로 HOG 특징 벡터를 계산하기에는 연산량이 너무 길어 실시간 알고리즘 수행이 불가능하다. 그래서 본 논문에서는 NVIDIA에서 지원하는 그래픽 카드와 CUDA 라이브러리를 이용하여 고속 연산이 가능한 병렬 처리 연산을 수행했다. 수행한 결과는 아래 Fig. 16와 같다. 프레임 해상도 640x480에서 GPU기반 연산처리가 CPU 기반 연산처리보다 약 34배 빠른 것을 알 수 있다.

7. 결론

본 논문에서는 50cm에서 70cm 내외의 근거리에서 영상정보 만을 이용하여 사람을 판단하고 사람과의 상대거리 및 방향을 추정하는 알고리즘을 제시하였다. HOG특징벡터와 SVM 분류기를 이용하여 사람의 두부 및 어깨를 검출하고, 검출된 윈도우의 위치와 크기를 이용하여 최종적으로 사람과 카메라간의 상대 거리 및 각도를 구하는 알고리즘을 구현하였다. 또한 알고리즘 검증을 위해 실제 객체가 존재하는 환경에서의 사람 검출 및 상대 거리를 계산 한 결과와 레이저 스

캐너의 거리 출력과 비교하였다. 수행 결과 특징 벡터의 추출이 알고리즘 성능에 큰 영향을 끼칠 수 있었다. 모자를 쓴 사람이나, 헤어스타일이 특이한 경우 및 사람의 목이 보이지 않는 영상에 대하여 인식률이 좀 떨어지지만 향후 다양한 모양/자세에 대한 긍정 예 이미지를 추가하여 SVM 분류기를 학습 시키면 거짓 부정 이미지에 인식률은 낮아지고 참 긍정 이미지에 대한 인식률은 높아질 것이다. 본 알고리즘에서는 레이저 스캐너의 상대 거리 및 각도 추정치와 크게 다르지 않음(오차 10cm정도)을 확인할 수 있었다. 또한 CPU와 GPU의 사용과 비교를 통해 CPU에 비해 수행 속도가 약 34배 빠름에 따라 GPU환경에서의 실시간 객체 검출이 가능함을 확인하였다. 따라서 향후 연구 방향은 본 알고리즘을 통해 계산된 상대 거리 및 각도를 활용하여 실제로 로봇이 사람을 추적 및 추종하는 연구가 될 것이다.

후 기

이 연구는 2010년도 충남대학교 자체연구비에 의해 지원되었습니다.

참고문헌

- (1) Enzweiler, M. and Gavrilu, D., 2009, "Monocular Pedestrian Detection : Survey and Experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 13, pp. 2179~2195.
- (2) Dalal, N. and Triggs, B., 2005, "Histogram of Oriented Gradients for Human Detection," *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*.
- (3) Broggi, A., Bertozzi, M. and Fascioli, A., 2000, "Shape Based Pedestrian Detection," *In Proc. of IEEE Intelligence Vehicles Symposium 2000*, pp.215~220.
- (4) Fang, Y., Yamada, K., Nunomiya, Y., B.H.P., H. and Masaki, I., 2004, "A Shape-Independent Method for Pedestrian Detection with Far-Infrared-Images," 43(5)
- (5) Broggi, A., Fascioli, A., Grisleri, P., Graf, T. and Meinecke, M., 2005, "Model Based Balidation Approaches and Matching Techique for Automative Vision Based Pedestrian Detection," *In Intl. IEEE Wks. on Object Tracking and Classification in and Beyond the Visible Spectrum*, San Diego, USA, page in press.
- (6) Mählich, M., Oberländeer, M., Lählein, O., Gavrilu, D. and Ritter, W., 2005, "A Multiple Detector Approach to Low-Resolution for Pedestrian Recognition," *In Proc. IEEE Intelligence Vehicles Symposium 2005*, pp.23~28.
- (7) Papageorgiou, C. and Poggio, T., 2000, "A Trainable System for Object Detection," *International Journal of Computer Vision*, 38(1):15~33.
- (8) Daugman, J. G., 1985, "Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimized by Two-Dimensional Visual Cortical Filter," *Journal of the Optical Society of America*, 2(7):1160~1169.
- (9) Schapire, R. E., 2002, "The Boosting Approach to Machine Learning, an Overview," *In MSRI Workshop on Nonlinear Estimation and Classification*.
- (10) Burges, C. J. C., 1998, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, Vol. 2, pp.121~167.
- (11) CUDA, http://www.nvidia.com/object/cuda_home.html.
- (12) GPGPU, <http://gpgpu.org>.
- (13) Object Oriented Scientific Computing, <http://www.oonumerics.org/blitz/>.