

자율주행차량의 장애물 인식을 위한 물체형상 및 움직임 포착에 관한 연구

이진우^o · 이영진 · 조현철 · 손주한 · 이권순
 동아대학교 전기공학과

A Study on Detection of Object Shape and Movement for Obstacle Recognition of Autonomous Vehicle

Jin Woo Lee · Young Jin Lee · Ju Han Son · Hyun Cheol Cho · Kwon Soon Lee
 Dong-A University

Abstract - It is important to detect objects movement for obstacle recognition and path searching of autonomous robots and vehicles with vision sensor. This paper shows the method to draw out objects and to trace the trajectory of the moving object using a CCD camera and it describes the method to recognize the shape of objects.

1. 서론

최근 산업 발달이 점차 가속화되면서 물류 자동화 및 생산성 향상을 위하여 무인반송차량(autonomous guided vehicle) 및 자율이동로봇(autonomous mobile robot) 등이 많이 이용되고 있다[1]. 특히, 무인 운행시 발생하는 장애물의 출현을 인식하고 이로 인한 충돌을 회피하는 것은 산업현장의 안정적 측면에서 대단히 중요하며, 이를 해결하기 위한 방안으로 근접 센서나 영상(image) 정보를 이용한 비전시스템(vision system)이 많이 응용되고 있다. 특히 고속컴퓨터의 개발에 따른 영상정보처리 속도가 빨라지고 영상정보가 다른 센서에 비해 보다 많은 정보를 얻을 수 있다는 이점 때문에 비전시스템의 이용이 늘어나고 있는 추세이다. 또한, 영상처리기술은 무인 감시장치나 효율적인 차량이동을 위한 도로의 교통량의 제어 등과 같이 다양한 분야에 응용될 수 있어 그 활용가치가 더욱 높다고 할 수 있다.

비전시스템을 이용한 영상정보 처리 기법으로는 한 대의 카메라를 이용한 단안(single vision)기법과 두 대의 카메라를 이용한 양안(stereo vision)기법으로 나눌 수 있다. 일반적으로 단안기법은 대상물체를 인식하고 단지 물체의 상대적 움직임을 알 수 있지만, 물체의 실질적인 이동과 카메라가 부착된 차량과 물체와의 절대적 거리를 검출하기 위해서는 두 대의 카메라를 이용하여야 한다[2]. 실제로 무인으로 운행되는 차량의 전방에 장애물이 돌발적으로 나타났을 때 물체의 움직임만으로는 물체와의 충돌을 회피할 수 없으며, 물체와의 실제거리와 물체의 실질적인 움직임을 통한 이동속도를 모두 알아야만 충돌을 회피하거나 차량의 속도를 줄여 보다 안전하고 효율적인 운행이 될 수 있을 것이다.

따라서, 본 논문에서는 한 대의 카메라를 이용하여 물체의 형상과 움직임 및 차량과의 실질적인 거리를 동시에 인식할 수 있는 알고리즘을 개발하고자 한다. 제안된 알고리즘에 대한 검증은 위하여 움직이는 몇 가지의 물체를 여러 환경에서 CCD 카메라를 이용하여 영상을 획득하는 가상의 실험을 실시하였으며, 실제 상황과 인식된 결과와의 차이를 비교분석하였다.

2. 본론

일반적으로 물체의 움직임을 분석하기 위해서는 정지된 배경에서 움직인 물체만을 차분영상에 의해 해석하는 방식을 사용하고 있다[2]. 그러나, 배경이 움직일 경우

영상내의 모든 부분이 움직이므로 물체를 추출하기 어려워진다. 본 연구에서는 물체만의 움직임뿐만 아니라 차량의 움직임에 의한 변위도 동시에 검출하여 물체를 검출할 수 있도록 하고자 한다. 먼저, 비전센서(vision sensor)를 이용하여 얻어진 영상을 전처리과정(preprocessing)을 거쳐 색상비율(color ratio)을 이용한 필터를 통해 물체와 배경을 분리하고, 레이블링 알고리즘(labeling algorithm)을 사용하여 경계치(threshold)내의 요소군을 가지는 물체들만을 추출할 수 있도록 한다. 다음단계로서 전처리과정을 통하여 추출된 물체의 특징정보들을 신경회로망 학습법을 이용한 패턴인식기(pattern recognizer)를 통하여 미리 정의된 형상으로 인식하게 된다[3]. 이때, 프레임(frame)간의 추출된 특징정보에 근거하여 그 물체들의 움직임도 함께 검출하고자 한다. 이렇게 인식된 물체들은 실좌표에서 시뮬레이터상의 가상의 좌표로 옮겨져 위치를 파악하게 되고, 그 다음 프레임의 정보들과의 차분값에 의해 움직임을 결정하게 된다. 그림 1은 본 논문에서 제안한 알고리즘에 대한 전체적인 처리과정을 나타내고 있다.

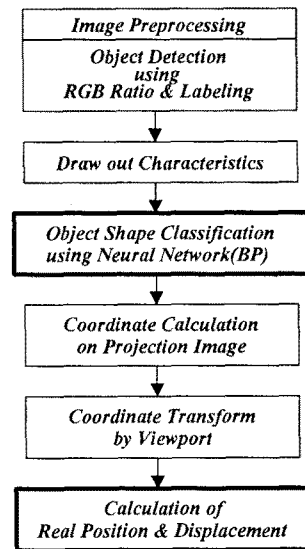


그림 1. 장애물 인식을 위한 영상처리 블록다이어그램

2.1 전처리 및 특징 추출

획득된 영상으로부터 물체의 형상 및 움직임을 인식하기 위하여 전처리과정을 통하여 색상비율을 이용한 필터와 레이블링 알고리즘을 사용하여 대상 물체들만을 검출하도록 하였다. 또한 형상인식을 위하여 신경회로망 패턴인식기를 이용하였다.

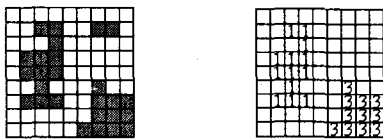
2.1.1 색상 비율을 이용한 전처리

CCD 카메라에서 입력되는 컬러 영상은 각각의 256가지의 RGB(red-green-blue)의 삼색으로 조합되어 있다. 컬러 영상의 색상비율을 이용하여 물체와 배경을 분리하기 위하여 입력되는 영상의 전체 색상비율을 구하여 우성의 색상을 중심으로 각 픽셀의 색상비율을 다시 조사하여 그 색상과 다른 두 색상과의 비가 설정된 경계구간 밖에 있는 픽셀들은 제거하게 된다.

2.1.2 레이블링(Labeling)

비전시스템에 있어서 가장 일반적인 조작 중 하나는 영상에서 연결된 요소(component)를 찾는 것이다. 연결된 요소의 점들은 나타난 물체에 대한 후보 영역을 형성한다. 획득된 영상에서 거의 모든 물체들은 면(面)을 가진다. 각각의 면에 속한 점들은 공간적으로 단려진 점들로 투영된다. 만약 영상에서 단 하나의 물체만 있다면, 연결된 요소들을 찾을 필요는 없지만, 영상에서 많은 물체의 특성과 위치를 찾아야 할 필요가 있을 때에는 연결되는 요소들을 결정해야만 한다[2].

요소 레이블링 알고리즘은 영상에서 모든 연결된 요소들을 찾고, 같은 요소 안의 모든 점들을 자기만의 레이블로 할당한다. 대상 물체만을 검출하기 위해서는 레이블링과정을 처리한 후 연결된 요소들의 크기에 경계값을 두어 그 크기에 벗어나는 요소들을 제거하는 필터링 작업을 해야 한다. 처리속도를 증가시키기 위하여 레이블링과정 안에 요소의 수가 너무 크거나 잡음일 수 있는 부분을 바로 제거하는 루틴(routine)을 삽입하였다. 그림 2는 원 영상과 그것의 레이블링 연결 요소들을 보여주고 있다. 그림 2(a)의 우측 상단 부분의 잡음을 레이블링하면서 요소수가 물체로 검출될 범위 안에 들지 않으므로 물체 후보에서 제거하게 된다. 그림 2(b)는 원 영상(original image)에서 레이블링되고 잡음은 제거된 결과를 보여주고 있는데, 여기서 2번의 요소군이 제거되고 바로 3번으로 레이블링되었음을 알 수 있다.



(a) 검출된 입력영상 (b) 레이블링된 영상
그림 2. 레이블링 필터

많은 응용분야에서 레이블링기법은 크기와 위치, 방위와 같은 특성들을 계산하기 위해서는 적절하다. 레이블링에는 순환적 알고리즘(recursive algorithm)과 순차적 알고리즘(sequential algorithm)의 두 가지 방법이 있다[2]. 본 논문에서는 순차적 레이블링을 사용하여 물체들을 검출하고, 그 물체들의 고유한 레이블을 부여하였다.

2.1.3 특징 추출 및 히스토리 생성

레이블링된 물체는 그 특징을 가지고 있다. 이후로 검출된 물체의 움직임을 포착하기 위해서는 그 히스토리(history)를 만들어야 한다. 물체의 히스토리에는 추출된 물체의 전방거리, 높이, 너비 및 형상패턴 등의 정보들이 포함되게 된다. 이러한 물체의 히스토리는 레이블링된 물체의 움직임을 파악하는데 중요한 정보로 이용된다. 물체의 위치를 알기 위해서는 그 물체의 중심점을 설정하여 CCD 카메라와의 거리를 파악하여야 한다.

본 논문에서는 평면 바닥에 근거한 단안 기법을 이용하여 거리를 측정하기 때문에 그 물체의 중심점을 최하단부의 중심 픽셀로 선정하였다. 그림 3(a)에서 보는 바와 같이 수직·수평으로 최외각에서 추출되는 점으로부터 대상 물체의 너비와 높이를 구할 수 있다. 그림 3(b)

는 물체 형상을 패턴 인식기로 입력될 데이터를 구하기 위하여 잘려진 물체의 부분영상을 정의된 $M \times M$ 의 행렬로 변환시킨 모습이다.



(a) 물체의 너비와 높이 (b) 정규화된 영상
그림 3. 물체 크기와 정규화된 물체 형상

2.1.4 형상 분류

물체의 형상 분류를 위하여 그림 3(b)의 정규화된 영상을 이용하였으며, 형상 패턴의 인식을 위하여 신경회로망 학습알고리즘을 이용하였다. 추출된 부분 영상이라 학습된 도형의 패턴으로 인식되기 위하여 신경회로망의 입력으로 하였다. 학습에 이용된 형상패턴은 구, 직육면체, 타원 및 원뿔의 네 가지로 분류하였으며, 학습에 이용된 신경회로망은 개선된 BP법을 사용하였다. 개선된 BP법은 지역해에 빠지지 않기 위해 학습시간이 이전의 BP법보다 길었지만, 인식단계에서 걸리는 시간은 같으면서도 인식률이 높은 장점이 있다[3]. 형태에 대한 정보는 이전 프레임에서 인식된 물체를 현재 프레임에서도 그 물체란 것을 인식하기 위해 필요하게 된다. 즉, 물체의 히스토리 안에는 픽셀정보에 의한 위치좌표와 물체의 크기와 형태 정보를 내포하고 있으므로 다시 다음 프레임에 의해 생성된 히스토리의 정보가 신경회로망에 입력된다.

2.2 물체 위치 및 움직임 포착

물체의 위치와 움직임을 포착하기 위하여 특징추출된 영상의 히스토리 정보를 이용하여 viewport에 의한 실제좌표변환을 통하여 검출하도록 하였다.

2.2.1 Viewport에 의한 실좌표 변환

본 논문에서는 한 대의 카메라를 이용한 단안 기법으로부터 바닥면에 근거하여 이동 물체를 검출하고, 검출된 신호를 이용하여 이동물체의 좌표를 계산함으로써 물체의 움직임을 포착할 수 있도록 하였다.

먼저, 물체의 위치와 움직임을 포착하기 위해서는 대상 물체가 포함된 영상좌표계를 3차원적 실좌표계로 변환하여야 한다[4]. 그림 4는 영상정보를 이용하여 실좌표계에서 물체의 위치를 알아낼 수 있도록 CCD 카메라의 viewport를 설정한 것이다. 중앙의 기울어진 평면에 투영되는 상이 실제 얻어진 영상화면과 일치하게 된다.

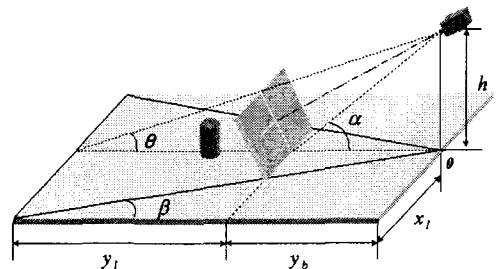


그림 4. Viewport

좌표계는 카메라의 높이 h 와 viewport 안의 실제 수직·수평 거리 x_1 , y_1 카메라와 최하단 view point와의 거리 y_2 에 의해 변환될 수 있다. 이 값들에 대한 상관관계는 식(1)과 같다. 여기서, $(\alpha - \theta)$ 는 카메라의 수직화각과 같게 된다. 그림 4의 중앙에 위치한 평면의 중앙점에 대한 수직벡터(normal vector)는 설치된 카메라

렌즈의 중앙부분과 평면의 중앙점을 지나는 벡터와 일치하게 된다.

$$\alpha = \tan^{-1} \left(\frac{h}{y_b} \right)$$

$$\theta = \tan^{-1} \left(\frac{h}{y_b + y_l} \right) \quad (1)$$

$$\beta = \tan^{-1} \left(\frac{y_b + y_l}{x_l} \right)$$

이로부터 영상에서 투영된 좌표를 이용하여 실제좌표를 구할 수 있다. 투영된 2차원영상의 좌표 픽셀 P_x 와 P_y 로 실좌표 x , y 의 관계를 나타내면 식(2)와 식(3)과 같다.

$$y = h \times \tan[(90^\circ - \alpha) + \left(\frac{P_y}{S_y}\right)(\alpha - \theta)] \quad (2)$$

$$x = y \times \left(\frac{P_x}{S_x}\right) \tan \beta \quad (3)$$

여기서 S_x 와 S_y 는 각각 수평·수직의 총 픽셀 개수를 나타낸다. 식(2)와 식(3)을 이용하여 픽셀 좌표점에서 실좌표계의 물체의 위치와 크기를 알 수 있다.

실좌표를 알아내기 위해 영상에서의 2차원 좌표를 다시 구성해야 한다. 그림 5는 전처리된 영상을 다시 실좌표로 변환하기 위해 원영상을 다시 원점을 설정하여 나타낸 것이다. 우측 viewport는 양의 부호를 가지고, 좌측은 음의 부호를 가진다. 이 좌표에 근거하여 검출된 물체들은 각각의 히스토리를 가질 것이다. 또한 이는 그림 4에 표시된 중앙부분의 평면상에 투영된 영상과 일치하게 된다.

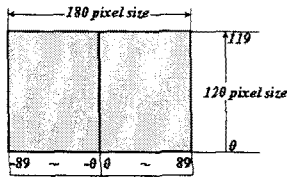


그림 5. 실좌표 변환을 위한 영상좌표계

새로운 영상좌표계에 의한 픽셀의 좌표점을 이용하여 viewport에 의한 실좌표로 변환되면 검출된 물체의 중심점에 의해 실좌표상의 x 와 y 를 구할 수 있다. 2차원 영상에서 검출된 물체의 너비와 높이는 실제의 크기로 해석하기 위해 실좌표상의 위치에서 다시 계산되어진다. 이 값이 그 물체의 실제 크기가 된다.

2.2.2 물체의 움직임 포착

현재의 영상에서 대상 물체들을 검출하여 생성된 히스토리 안에는 이전과 현재 프레임에 대한 그 물체 고유의 크기(너비, 높이)와 실좌표계로 변환된 좌표점이 기록되어 있다. 그림 6은 각 프레임 영상으로부터 구해진 히스토리를 나타낸다.

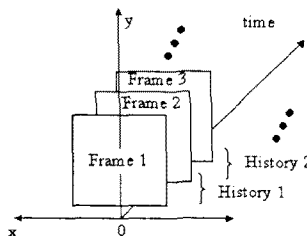


그림 6. 각 프레임에서의 히스토리

히스토리에에는 이전의 정보와 현재의 정보가 저장되어 있으며, 현재의 정보는 다시 다음 프레임에 대한 히스토리의 이전정보로 입력된다. 또한 전처리과정으로부터 검출된 각 물체들은 고유 레이블을 가지고 있고 전후 프레임들의 레이블은 일치하여야 하며, 동일한 레이블을 갖는 물체는 히스토리 안에 현재의 위치와 이전의 위치로 변위를 계산하여 움직임을 알아 낼 수 있다. 그림 7에서 보는 바와 같이 히스토리 내부의 정보는 연계되어 있다. 현재 프레임에서 물체의 변위는 이전 프레임의 위치와 현재 위치의 차분값으로 알아낼 수 있다.

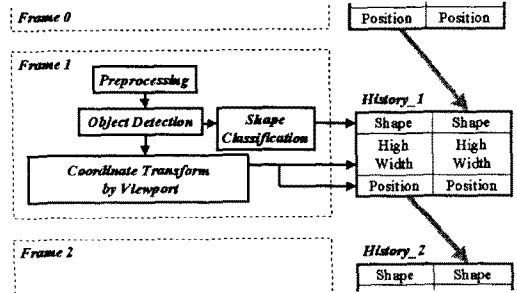


그림 7. 히스토리를 이용한 물체 움직임 포착

각 물체에 대한 변위를 구하게 되면, 상대적으로 물체가 차량에 대해 어떻게 움직이고 있다는 것을 알 수 있게 된다. 차량에 CCD 카메라를 부착하여 적용한다면 여러 방향으로 움직이고 있는 물체 중에서 급격히 다가오는 물체에 대해서는 장애물로 인식되어 충돌의 위험을 방지할 수 있을 것이다.

3. 실험 결과 및 고찰

본 실험에 사용된 카메라는 수평 48° , 수직 36° 의 화각을 가지며, 해상도는 처리속도의 신속성과 계산량을 줄이면서도 물체를 검출함에 있어 무리가 없도록 하기 위하여 영상의 사이즈는 $180(H) \times 120(V)$ 의 픽셀수를 갖도록 설정하였다. PC는 Windows 98의 OS환경 하에 CPU는 Pentium II, MMX 266MHz를 사용하였다.

실험에서 설정된 값은 $\alpha = 44^\circ$, $\beta = 66^\circ$, $\theta = 8^\circ$, $h = 47[\text{cm}]$, $y_b = 49[\text{cm}]$, $y_l = 285[\text{cm}]$, $x_l = 149[\text{cm}]$ 이며, 실험실 내에 viewport를 설치하였다. 실험방법은 정지된 직각육면체 1개와 같은 크기의 원통형 물체 2개를 움직이면서 영상을 획득하고 전처리과정을 거친 후 viewport에 의한 좌표변환을 하였다.

그림 8(a)는 입의의 3개 프레임에 대한 원영상이며, 그림 8(b)는 원영상을 프레임별로 전처리과정을 거친 영상이다. 잡음은 제거되고, 고유의 색으로 레이블링되어 있다.

frame 1 frame 2 frame 3



(a) 물체의 움직임에 대한 원영상



(b) 전처리과정을 거쳐 검출된 영상
그림 8. 원영상과 전처리된 영상

4. 결 론

전처리과정을 거친 영상은 각각의 검출된 물체들에 대해 CCD 카메라와 가장 가까운 픽셀을 찾게된다. 이 픽셀을 이용하여 viewport에 의한 좌표변환을 하면 실좌표상의 위치를 얻게 된다. 본 실험에서는 CCD 카메라로부터의 물체에 대한 실제위치를 각 프레임에 대하여 측정하고, 획득된 영상정보를 이용하여 위치를 계산하였다. 표 1은 실제 측정된 물체의 위치를 나타내고, 표 2는 비전센서에 의해 구해진 물체의 위치를 나타낸다. 좌측은 음의 부호로 나타내었다.

표 1. 각 물체의 실공간좌표 (단위:cm)

Frame	물체 1	물체 2	물체 3
1	(-13.0, 78.0)	(0.0, 65.0)	(35.0, 123.0)
2	(-13.0, 64.0)	(0.0, 65.0)	(22.0, 138.0)
3	(-8.0, 49.0)	(0.0, 65.0)	(8.0, 107.0)

표 2. 영상처리에 의한 좌표 (단위:cm)

Frame	물체 1	물체 2	물체 3
1	(-13.700, 78.247)	(-0.977, 65.091)	(35.510, 122.386)
2	(-12.998, 64.372)	(-0.966, 64.371)	(21.987, 137.348)
3	(-7.958, 49.710)	(0.666, 66.560)	(7.954, 106.911)

물체 1은 좌측에서 접근하고 있는 원통체이고, 물체 2는 정지하고 있는 직각육면체이며, 물체 3은 우측에서 접근하는 물체 1과 같은 크기의 원통체다. 그림 9는 각각 세 물체의 측정된 위치와 변환된 실좌표상의 위치의 움직인 궤적을 평면상에 나타내었다. 점선은 측정된 값이고, 실선은 계산된 값이다. 그림 9에서 알 수 있듯이 물체 1과 3은 실제이동을 잘 추정함을 알 수 있으며, 움직이지 않는 물체 2는 다소 오차가 발생하였으나, 전처리과정에서 발생된 것으로 추정되며, 추후 보완이 요구된다.

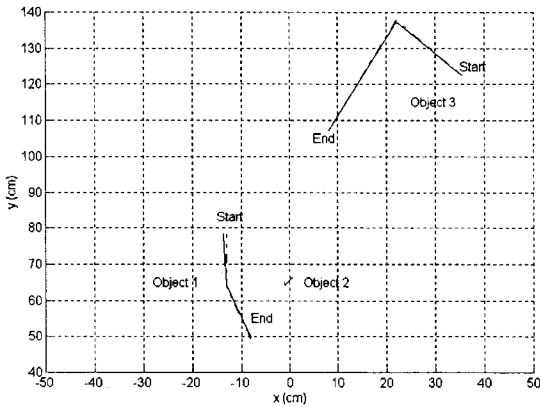


그림 9. 물체의 측정값과 계산값에 대한 궤적

시뮬레이션 결과, CCD 카메라로부터 멀리 있는 물체일수록 픽셀당 계산되는 거리의 범위가 커지므로 오차가 커질 것으로 예상했지만, 오히려 가까이 있는 물체가 검출과정에서 부분적인 손실로 인해 오차가 더 크게 나왔다.

자율이동 로봇 및 차량에서 전방 또는 측·후방에 존재하는 물체가 현재 상대적으로 어떻게 움직이고 있는가를 알아내는 것은 중요하다. 따라서, 본 논문에서는 단안 기법을 이용하여 3차원 공간상에 있는 물체의 위치를 알아내어 실측된 데이터와 비교하였다. 그 결과 움직이는 물체 1과 3은 실제좌표와 거의 일치하여 검출됨을 알 수 있었으며, 움직이지 않는 물체 2도 거의 일치하였지만, 전처리과정에서 검출된 영상의 아래 부분이 손실되어 위치를 찾는 중심점이 이동되어 다소 오차가 발생되었다.

시뮬레이션 결과 나타난 문제점으로는 전처리과정에서 배경 색상비율과 유사한 색상의 물체는 검출할 때에는 배경이 가지는 색상비율의 경계치에 속하는 경우 그 물체를 배경으로 처리되어 검출되지 않는 경우도 발생하였다. 따라서, 이 점을 극복할 수 있도록 전처리과정에 대한 추가적인 연구가 있어야 할 것으로 사료된다.

[참 고 문 헌]

- [1] G.Cheng and A. Zelinsky, "Real-time visual behaviours for navigating a mobile robot," Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 2, pp. 973-980, 1996.
- [2] R. Jain, R. Kasturi, and B. G. Schunch, *Machine Vision*, McGraw-Hill, 1995.
- [3] Weiyang, "Verification of the nonparametric characteristics of backpropagation neural networks for image classification," IEEE Transactions on Geoscience & Remote Sensing , Vol. 37, No. 2, pp. 771-779, 1999.
- [4] D. Hearn and M. P. Baker, *Computer Graphics C Version*, Prentice Hall, pp. 431-468, 1997.
- [5] J. A. S. Centeno, and V. Haertel, "An adaptive image enhancement algorithm," Pattern Recognition, Vol. 30, No. 7, pp. 1183-1189, 1997.
- [6] M. Umasuthan and A. M. Wallace, "Model indexing and object recognition using 3D viewpoint invariance," Pattern Recognition. Vol. 30, No. 9, pp. 1415-1434, 1997.
- [7] Y. Ye, "Sensor planning for 3D object search," Computer Vision and Image Understanding, Vol. 73, No. 2, pp. 145-168, 1999.