

<논 문>

스캔라인 연속영상을 이용한 실시간 장애물 인식에 관한 연구

정 성 엽* · 오 준 호*

(1996년 5월 10일 접수)

A Study on the Real Time Obstacle Recognition by Scanned Line Image

Sheung Youb Cheung and Jun Ho Oh

Key Words : Computer Vision(컴퓨터비전), Stereo Image(스테레오이미지), Scanned Line Image(스캔라인 연속영상), Obstacle Detection(장애물 인식), Real Time Image Processing(실시간 영상처리), Multiple Matching(다중 매칭), Kalman Filter(칼만필터)

Abstract

This study is devoted to the detection of the 3-dimensional point obstacles on the plane by using accumulated scan line images. The proposed accumulating only one scan line allow to process image at real time. And the change of motion of the feature in image is small because of the short time between image frames, so it does not take much time to track features. To obtain recursive optimal obstacles position and robot motion along to the motion of camera, Kalman filter algorithm is used. After using Kalman filter in case of the fixed environment, 3-dimensional obstacles point map is obtained. The position and motion of moving obstacles can also be obtained by pre-segmentation. Finally, to solve the stereo ambiguity problem from multiple matches, the camera motion is actively used to discard mis-matched features. To get relative distance of obstacles from camera, parallel stereo camera setup is used. In order to evaluate the proposed algorithm, experiments are carried out by a small test vehicle.

1. 서 론

로봇이 이동하기 위해서 주변 환경에 대한 정보를 가지고 있어야 한다. 주변 환경을 인식하는 방법으로 초음파센서, 레이저센서 등 능동센서를 이용하는 방법과 시각센서와 같은 수동센서를 이용하는 방법이 있다.⁽¹⁴⁾ 능동센서는 한번 하드웨어로 구성하게 되면 적용 대상을 확장시키거나 변경할 수 없는 반면에 수동센서는 기능이 소프트웨어에 의하여 결정되는 경우가 많다. 시각센서로 시스템

을 구축하면 기능을 추가 또는 변경하기가 용이하게 된다. 따라서, 시각센서는 이동로봇에 적합한 센서라고 할 수 있다.^(1~3) 카메라로부터 받아들인 영상으로부터 물체까지의 상대거리를 구하는 방법으로는 상대적으로 고정된 두 대의 카메라로부터 얻은 영상과 카메라의 상대위치로부터 삼각법에 의해 거리는 구하는 방법인 스테레오 매칭과 한 대의 카메라가 움직이며 연속적으로 받아들인 영상으로부터 각 점의 변위를 구하여 변위에 반비례하는 거리를 구하는 모션 스테레오의 방법이 있다.⁽⁹⁾ 스테레오 매칭에서 가장 중요하고 어려운 문제는 한 영상에서의 한 점이 다른 영상에서 어디에 해당하는

*회원, 한국과학기술원 기계공학과

를 찾아내는 대응문제이다.⁽⁸⁾ 대응문제가 해결되면 간단한 기하학적인 방법으로 거리를 알아낼 수 있다. 시각센서를 이용하는 경우 정보의 양이 많기 때문에 유용한 정보를 얻기 위해서는 많은 시간이 소요된다. 따라서 시간단축을 위해 경험적인 방법을 이용한 연구^(2,4) 축소영상을 이용한 연구⁽³⁾ 등이 있다. 연속적인 에피폴러 라인상의 이미지를 가지고 모션 스테레오를 이용하여 고정된 에피폴러 라인상의 물체의 위치를 알아내는 연구⁽¹⁰⁾가 있었다. 이 방법은 자신의 모션을 정확히 알고 있는 경우 고정된 에피폴러 라인에서 물체의 에지를 트래킹하기 쉬운 장점을 가진다. 그러나 에피폴러 라인상에 걸리는 물체에 대해서만 거리가 구해지는 단점이 있다. 스캔라인에서 이미지를 스캔하는 경우는 시간이 지남에 따라 에피폴러 라인의 위치가 계속적으로 변하기 때문에 시간이 지남에 따라 전체의 영상을 얻게 되는 효과를 가진다. 따라서 본 연구는 스캔라인 연속영상의 장점을 이용하여 실시간으로 로봇이 진행함에 따라 변화하는 환경에 대한 물체의 위치를 알아내는 것을 목적으로 한다.

2장은 스캔라인을 적용하기 위한 기본가정 및 스캔라인의 의미와 스테레오 매칭, 로봇의 운동에 대하여 설명하였고, 3장은 2장에서 구한 물체의 거리와 영상간의 관계를 이용한 최적운동과 최적위치를 구하는 칼만필터에 대하여 설명하였다. 4장은 실험 및 고찰에 대하여, 5장은 결론을 서술하였다.

2. 스캔라인 연속영상

2.1 기본 가정

본 논문에서 사용된 가정은 다음과 같다.

- (1) 물체는 바닥으로부터 지지되어 있다.
- (2) 물체는 특징으로 구별될 수 있다.
- (3) 두 개의 카메라로부터 나오는 영상의 밝기는 같다.
- (4) 이웃하는 두 영상의 시간간격은 충분히 짧다.
- (5) 이동로봇의 운동과 물체의 상대운동은 바닥상의 2차원 평면운동이다.

이러한 가정으로부터 물체의 경계는 스캔라인 상의 에지(edge)라는 특징으로 구분이 가능하여 물체의 위치는 바닥 평면상의 Z 로 나타낼 수 있다. Fig. 1에서 알 수 있듯이 핀 효율 모델을 이용하면

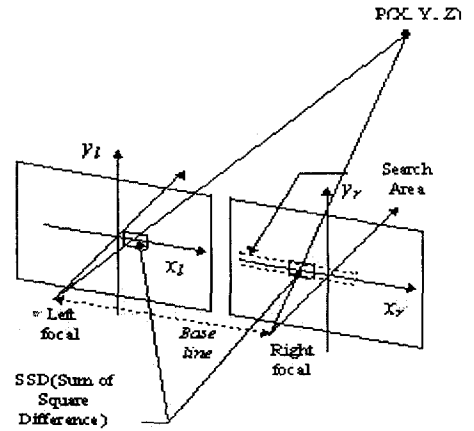


Fig. 1 Geometrical model of stereo matching

카메라로부터 물체의 위치는 식 (1)와 같이 구할 수 있다.⁽⁹⁾

$$X = \frac{B \cdot \left(\frac{x_l + x_r - C_x}{2} \right)}{2 \cdot Z}, \quad Z = \frac{B \cdot S_x}{x_l - x_r} \quad (1)$$

2.2 스캔라인 연속영상

로봇이 이동함에 따라 카메라로부터 얻게 되는 영상은 Fig. 2(a), (b)에서 $T=1$ 에서부터 $T=n$ 까지 모두 n 개의 영상이 될 것이다. 시간축으로 이웃한 $T=k$ 와 $T=k-1$ 두 영상을 생각하면 영상에 담긴 정보는 거의 일치한다. 또한 이미지상의 한 라인을 연속적으로 적층시키면 Fig 2(c)와 같은 영상을 얻을 수 있다. 즉, 외부의 정보를 한번에 모두 볼 수 있기도 하지만 n 시간 동안 이미지 상의 한 라인을 연속시키도 얻을 수 있다. 이렇게 얻은 영상을 스캔라인 연속영상이라 정의한다. 스캔라인 연속영상과 전체 이미지는 Fig 2(d)와 같은 관계를 가진다. 이렇게 n 시간 동안 얻은 스캔라인 연속영상에 대하여 거리를 구하는 경우는 매 순간 계산하여야 할 영상의 양이 작기 때문에 실시간 영상처리에 접근할 수 있다. 이러한 방법이 n 시간 동안 받은 영상중 $T=1$ 과 $T=n$ 에 받은 2장의 영상에 대하여 영상처리를 하는 것과 다른 점은 2장의 영상을 처리하는 것은 $T=1$ 과 $T=n$ 사이의 중간과정을 알지 못하고 시간간격이 크기 때문에 에지를 트래킹하기 위한 탐색영역이 크다는 단점도 있다. 반면에 스캔라인 연속영상을 이용하면 n 장의 영상을 모두 사용하고, 또한 이웃한 영상의 시

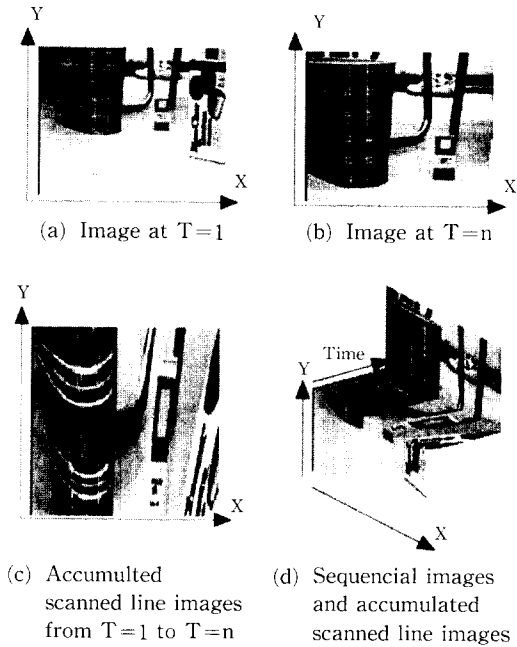


Fig. 2 Scanned line image

간격이 작기 때문에 상대적으로 에지를 트래킹 (tracking)하기 위한 탐색영역이 작아진다. 또한 하나의 Line CCD로도 구현이 가능하기 때문에 하드웨어 구성이 간단해지는 장점을 가진다.

2.3 스캐라인 상에서의 물체의 운동

물체의 좌표를 X, Z 라고 할 때, 현재 순간의 이미지에 맺힌 상의 좌표 x , 다음 순간의 좌표 x' 은 다음과 같은 관계가 있다.⁽⁹⁾

$$x = f \frac{X}{Z}, x' = f \frac{X - T_x}{Z - T_z} \tag{2}$$

기본 가정으로부터 물체는 바닥 평면상의 2차원 운동으로 나타낼 수 있다. 스테레오 카메라간의 거리를 B , 카메라의 초점거리 f , 전 순간에 스테레오 매칭으로 구한 이미지 좌표의 차를 d , 현재 순간에 구한 이미지 좌표의 차를 d' 라 하면 식 (3)과 같이 물체의 $X-Z$ 평면상의 상대운동 T_x, T_y 이 구해진다.

$$T_x = \frac{\Delta x_l + \Delta x_r}{d} \cdot \frac{\Delta d \cdot (x_l + x_r)}{d^2}$$

$$T_y = Bf \frac{-\Delta d}{d^2}$$

$$d = x_l - x_r$$

$$d' = (x_l + \Delta x_l) - (x_r + \Delta x_r)$$

$$= d + \Delta d$$

$$\Delta d = \Delta x_l - \Delta x_r \tag{3}$$

2.4 시각센서로 구한 물체의 위치오차에 대한 분산

x_l, x_r 는 스테레오 이미지상에 맺힌 점이며, d 는 두 점의 차이, s_x, c_x, B 를 각각 카메라의 scale factor와 이미지상의 x 좌표의 중심, 두 카메라 간의 간격을 나타내고, σ_l, σ_r 는 이미지의 이산오차에 의한 분산이라 할 때 측정거리 X, Y 의 상호분산 C_v 는 식 (4)와 같다.⁽¹³⁾

$$C_v = J \begin{bmatrix} \sigma_l & 0 \\ 0 & \sigma_r \end{bmatrix} J^T$$

$$J = \begin{bmatrix} \frac{\partial \hat{X}}{\partial x_l} & \frac{\partial \hat{X}}{\partial x_r} \\ \frac{\partial \hat{Z}}{\partial x_l} & \frac{\partial \hat{Z}}{\partial x_r} \end{bmatrix} = \begin{bmatrix} -B \frac{x_r - c_x}{d^2} & B \frac{x_l - c_x}{d^2} \\ -B \frac{s_x}{d^2} & B \frac{s_x}{d^2} \end{bmatrix} \tag{4}$$

2.5 제어신호로부터 계산된 단위시간동안 로봇의 운동량

기본 가정으로부터 로봇의 운동 M_i 은 바닥 평면상의 직선이동량 ΔT 와 회전이동량 $\Delta \theta$ 로 나타낼 수 있다. 로봇의 두 바퀴의 단위시간 이동량을 S_l, S_r , 바퀴사이의 거리 l 이라고 하면 M_i 는 다음과 같다.

$$M_i = \begin{bmatrix} \Delta T \\ \Delta \theta \end{bmatrix} = \begin{bmatrix} \frac{S_r + S_l}{2} \\ \frac{S_r - S_l}{l} \end{bmatrix} \tag{5}$$

3. 칼만필터를 이용한 물체의 위치와 로봇의 운동 추정

로봇이 이동함에 따라 연속적으로 구해지는 스테레오 이미지에서 매 순간마다 물체까지의 거리를 구할 수 있고 시간축으로 이웃한 두 쌍의 스테레오 이미지로부터 로봇의 직선운동과 회전운동을 구할 수 있다. 그러나, 이동로봇의 운동의 양에 비하여 물체의 거리를 구하는 과정에서 생기는 위치오차의 영향이 크기 때문에 매 순간 구해지는 운동은 많은 오차를 포함하게 된다. 시각센서와 로봇의 내부거리 센서로부터 나온 데이터를 가지고 칼만필터를 이용하여 물체의 최적위치 및 카메라의 운동을 구

하는 연구는 많이 있어 왔다.^(12,13) 본 연구에서는 스캔라인상에서 연속적으로 구해지는 물체의 위치의 최적위치 및 로봇의 최적운동을 구하기 위하여 칼만필터를 도입하였다. 고정물체와 이동물체가 구별이 되는 경우는 이동물체의 운동과 위치를 구할 수 있었다.

3.1 물체의 위치추정

현재 시각에서의 X-Z 평면상에 물체의 상대위치를 $x(k)$, 전 시각에서의 물체의 상대위치를 $x(k-1)$ 이라고 하고, $R(k)$, $T(k)$, $w(k)$ 를 로봇의 회전운동, 직선운동 및 시스템 노이즈라고 한다면, 다음과 같이 상태방정식이 정의된다.

$$x(k) = R(k)x(k-1) + T(k) + w(k) \quad (6)$$

시스템 노이즈에 대한 상호분산(covariance matrix)을 C_w , 선형화과정에서 생긴 운동변수에 대한 상호분산을 C_m 라 하고, $x_p(k)$, $P_p(k)$ 는 $t=k$ 순간에 위치 및 오차의 상호분산, $x_u(k-1)$, $P_u(k-1)$, $x_u(k)$, $P_u(k)$ 는 $t=k-1$, $t=k$ 순간에 칼만필터위치 update 및 오차의 상호분산, $K(k)$ 는 칼만필터 게인이라고 할때 위치에 대한 칼만필터 방정식은 식 (7)과 같다.

$$\begin{aligned} x_p(k) &= R(\theta_n)x_u(k-1) + T(\theta_n, \Delta z_n) \\ P_p(k) &= R(\theta_n)P_u(k-1)R(\theta_n) \\ &\quad + [D_\theta \ D_{\Delta z}]C_m[D_\theta D_{\Delta z}]^T + C_w \end{aligned}$$

$$M_v = \begin{bmatrix} \Delta z_n \\ \theta_n \end{bmatrix} = \begin{bmatrix} \sqrt{x_m(k)^2 z_m(k)^2 - x_u(k-1)^2} - z_u(k-1) \\ \sin^{-1}\left(\frac{x_m(k)}{\sqrt{x_m(k)^2 + z_m(k)^2}}\right) - \sin^{-1}\left(\frac{x_m(k-1)}{\sqrt{x_m(k)^2 + z_m(k)^2}}\right) \end{bmatrix} \quad (8)$$

M_i 는 식 (5)로 구하며, M_i 의 상호분산 C_i 는 실험으로 구한다. 이때 추정된 로봇의 운동 M_e 은 식 (9)과 같다.

$$\begin{aligned} M_e &= \left[\sum_{i=0}^N (D^{-1}[P_m R(\theta_n) P_u R(\theta_n)]^T \right. \\ &\quad \left. + C_w \right]_i D^{-T} \left[\sum_{i=0}^N [(D^{-1}[P_m R(\theta_n) P_u R(\theta_n)]^T \right. \\ &\quad \left. + C_w \right]_i D^{-T} \right]^{-1} + C_i^{-1} M_i \end{aligned} \quad (9)$$

3.3 이동물체의 위치와 운동추정

2차원 평면에서 운동을 하고 있다면, 다음과 같은 방법을 적용해 운동과 위치를 추정할 수 있다. 물체의 운동 $T^m(k)$ 은 식 (10)으로 모델링된다.

여기에서,

$$\begin{aligned} D_\theta &= \frac{\partial}{\partial \theta} (R(\theta)x(k) - T(\theta, \Delta z)) \Big|_{\theta=\theta_n, \Delta z=\Delta z_n, X=Xu(k)} \\ D_{\Delta z} &= \frac{\partial T(\theta, \Delta z)}{\partial \Delta z} \Big|_{\theta=\theta_n, \Delta z=\Delta z_n} \\ K(k) &= P_p(k)[P_p(k) + C_v]^{-1} \\ x_u(k) &= x_p(k) + K(k)[z(k) - x_p(k)] \\ P_u(k) &= P_p(k) - K(k)P_p(k) \end{aligned} \quad (7)$$

3.2 물체의 운동추정

로봇의 이동은 로봇의 진행방향의 회전 θ 와 진행방향의 이동 Δz 의 함수이다. 2장에서 언급한 바와 같이 로봇의 이동을 내부의 센서만을 이용하는 경우 매 순간 구해지는 오차는 작지만 시간이 지남에 따라서 오차가 계속 누적되어 전체 좌표에서의 운동오차는 커질 수 있는 반면에 시각센서로 구한 운동은 매 순간 구해지는 오차는 크지만 전체 좌표에서는 오차가 누적되지 않는다. 따라서 이 두 가지의 센서의 장점을 이용할 수 있는 칼만필터를 적용하여 로봇의 운동을 구해 낸다. 시각센서를 이용하여 구한 운동 M_v 는 $t=k$ 순간에 물체의 좌표 X, Z 와 $k=k-1$ 순간에 칼만필터로 update된 좌표를 이용하여 식 (8)과 같이 구한다.

$$T^m(k) = x(k) - R_x(K-1) - T + w(k) \quad (10)$$

\hat{R} , \hat{T} 은 정지해 있는 물체들로부터 구한 로봇의 운동추정값이라고 할때 시각센서를 이용하여 구한 물체의 운동을 T_p^m , 상호분산을 C_p^m 이 되고 물체의 운동과 상호분산의 추정값 $T_u^m(k)$, $P_u^m(k)$ 은 다음과 같다. 또한 이동물체의 위치는 고정위치에서 구한 것과 식 (11)과 같다.

$$\begin{aligned} T_p^m(k) &= x_m(k) - \hat{R}x_u(k) - \hat{T} \\ P_p^m(k) &= P_m(k) - \hat{R}P_u(k) \hat{R}^T + DC_m D^T + C_w \\ T_p^m(k) &= [P_u^m(k-1)^{-1} + P_p^m(k)^{-1}]^{-1} \\ &\quad [P_u^m(k-1)^{-1} T_u^m(k-1) \\ &\quad + P_p^m(k)^{-1} T_p^m(k)] \\ P_u^m(k) &= [P_u^m(k-1)^{-1} + P_p^m(k)^{-1}]^{-1} \end{aligned}$$

4. 실험 및 결과

제안된 스캐라인 연속영상을 적용함으로써 로봇이 이동하는데 필요한 주변 물체들의 위치를 실시간으로 알아낼 수 있는가 알아보기 위하여 고정 환경에서의 물체의 위치 및 로봇의 운동추정실험과 이동물체를 알고 있는 경우 이동물체의 위치 및 운동추정실험을 하였다. 그리고 스테레오 매칭에서 다중 매칭이 일어나면 물체의 거리가 잘못 구해지게 된다. 따라서 두개의 비슷한 기둥에 대하여 물체의 거리가 어떻게 구해지고 어떻게 해결할 수 있는가에 대한 실험을 하였다. 이와 같은 실험을 통하여 물체가 바닥에서 수직으로 서있는 경우에 대해서는 로봇의 이동에 필요한 주변 환경정보를 스캐라인 연속영상을 이용하여 구할 수 있다는 것을 알 수 있었다.

4.1 실험에 사용된 이동로봇

로봇은 스텝모터 두개와 종동축 하나로 구성되며, 스테레오 이미지를 받기 위한 두 개의 카메라와 TMS 320 C 30 DSP 영상처리보드로 구성되어 있어 소형 컴퓨터(PC)로부터 프로그램을 다운로드하여 사용할 수 있게 되어 있다. 소형 컴퓨터는 영상처리보드에서 나오는 데이터를 이용하여 칼만필터의 계산을 수행하고 스텝모터를 구동하여 로봇을 이동시킨다.

4.2 로봇의 운동추정실험

4.2.1 칼만필터를 적용한 경우 로봇의 운동추정 실험을 직선과 회전, 2 가지에 대하여 행하였다. — 직선운동추정실험

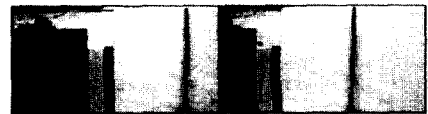
로봇을 Z의 방향으로 5.8(mm/Frame)의 속도로 이동시키며 100장의 영상을 받았다. Fig. 3(a), (b), (c)는 이 실험에 사용된 영상으로 시작점의 스테레오 영상과 끝점의 스테레오 영상 및 스캐라인 연속영상을 나타낸다. Fig. 3(d)은 직선운동실험의 경우 운동오차를 나타낸 것으로 직선운동오차는 시각센서로만 구한 경우는 ± 5 mm 정도의 오차가 나는 반면 칼만필터를 적용하여 구한 운동은 실제 값과 유사한 것을 알 수 있었다. 곡선운동의 경우 오차는 $\pm 0.05^\circ$ 정도인데 약간의 옅색이 존재한다. 역시 칼만필터를 적용한 것이 시각센서로만 구한 것보다는 오차가 적다. Fig. 3(e)는 절대좌표에



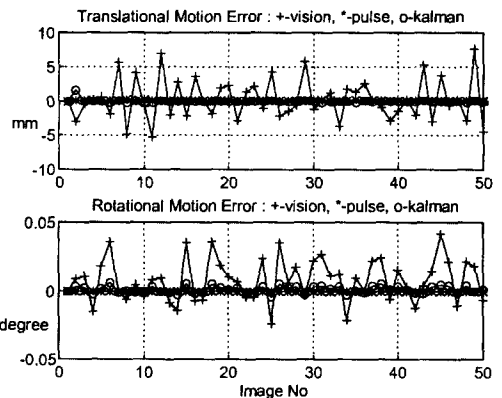
(a) Stereo image at T=1



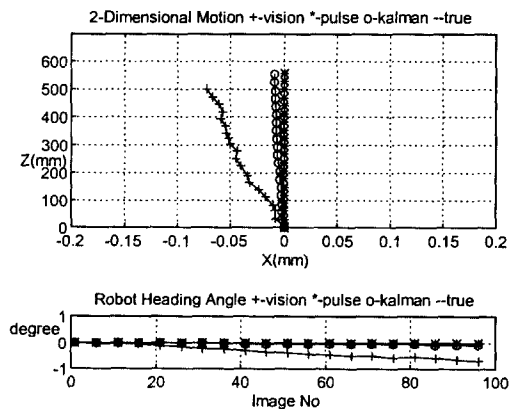
(b) Accumulated scanned line images from T=1 to T=100



(c) Stereo image at T=100



(d) Error of estimated motion



(e) Motion in world coordinate

Fig. 3 Experiment of pure translational motion

서 로봇의 움직임을 나타낸 것으로 580 mm 정도 이동에 시각센서로 구한 것이 -0.5 mm 정도의 오차를 보이며, 칼만필터로 구한 것은 -0.01 mm의

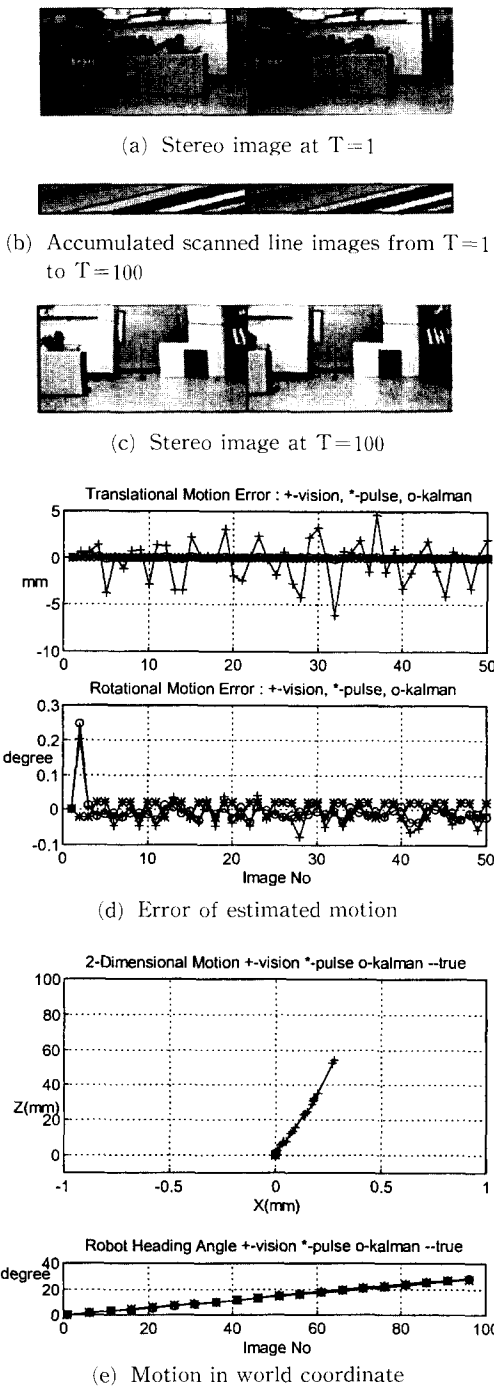


Fig. 4 Experiment of pure rotational motion

오차를 보이는 것을 알았다. 로봇의 진행방향각 (heading angle)의 오차는 -0.7° 정도로 크지 않다. 이 시스템은 정밀한 캘리브레이션을 요구하기

않기 때문에 초점거리등 카메라의 파라미터의 오차의 영향으로 옅셋이 생길 수 있다. 그러나 이 옅셋은 크지 않으므로 이 시스템에는 만족할 만하다.

4.2.2 회전운동의 경우

회전운동의 경우는 $0.283(\text{degree}/\text{Frame})$ 이동시켰으며 모두 100 장의 적층영상을 받았다. Fig. 4(a), (b), (c)은 회전운동실험에 사용된 영상으로 시작점에서의 스테레오 영상, 끝점에서의 스테레오 영상 및 스캔라인 연속영상을 나타낸다. Fig. 4 (d)은 회전운동의 경우에 직선운동오차, 회전운동오차를 나타낸 것으로 회전운동시에도 직선운동에서 구한 오차와 크게 다르지 않다. 그러나 시각센서로 구한 운동이 직선운동시에는 회전운동성분의 오차에 옅셋이 존재하였으나 회전운동의 경우는 직선운동성분의 오차의 옅셋이 존재하는 것을 알 수 있다. 이것은 회전하면서 물체가 나타나고 사라지는 과정에서 생기는 오차로 생각되며 앞에서 언급한 카메라의 캘리브레이션정도에도 영향이 있다. 즉, 직선운동시에는 거리의 차이가 운동이 되므로 직선운동오차의 옅셋이 줄어들고 회전운동의 경우는 회전의 차이가 운동이 되므로 회전운동오차의 옅셋이 줄어는 것으로 생각할 수 있다. Fig. 4 (e)는 절대좌표에서 로봇의 이동과 진행방향각의 변화를 나타낸 것이다. 시각센서로만 구한 로봇의 이동의 오차는 옅셋의 영향으로 커지게 된다. 칼만필터로 구한 로봇의 이동의 오차는 크지 않음을 알 수 있다.

4.3 물체의 위치추정실험

직선운동의 경우에 대한 물체의 위치추정실험을 하였다. Fig. 5는 4 개의 물체에 대하여 시각센서로만 구한 위치오차와 칼만필터를 적용하여 구한 위치오차를 나타낸 것이다. 로봇이 진행함에 따라 물체까지의 거리가 가까워지고 따라서 영상의 이산오차의 영향이 작아지기 때문에 물체의 Z방향의 위치오차가 줄어드는 것을 알 수 있다. 그러나 물체의 X방향의 오차는 크게 줄어들지는 않고 옅셋이 존재하는 데 이것은 카메라의 캘리브레이션의 정도에 따른다고 생각된다. 로봇이 진행하는 데 있어서 물체를 피해나가고자 하는 경우 물체의 옆을 바로 지나가지 않고 어느 정도의 허용오차의 범위를 두고 지나갈 것이다. 따라서 위에서 구한 X방향의 수 mm 위치 오차 및 Z방향의 수십 mm의 위치오차는 만족할 만 하다.

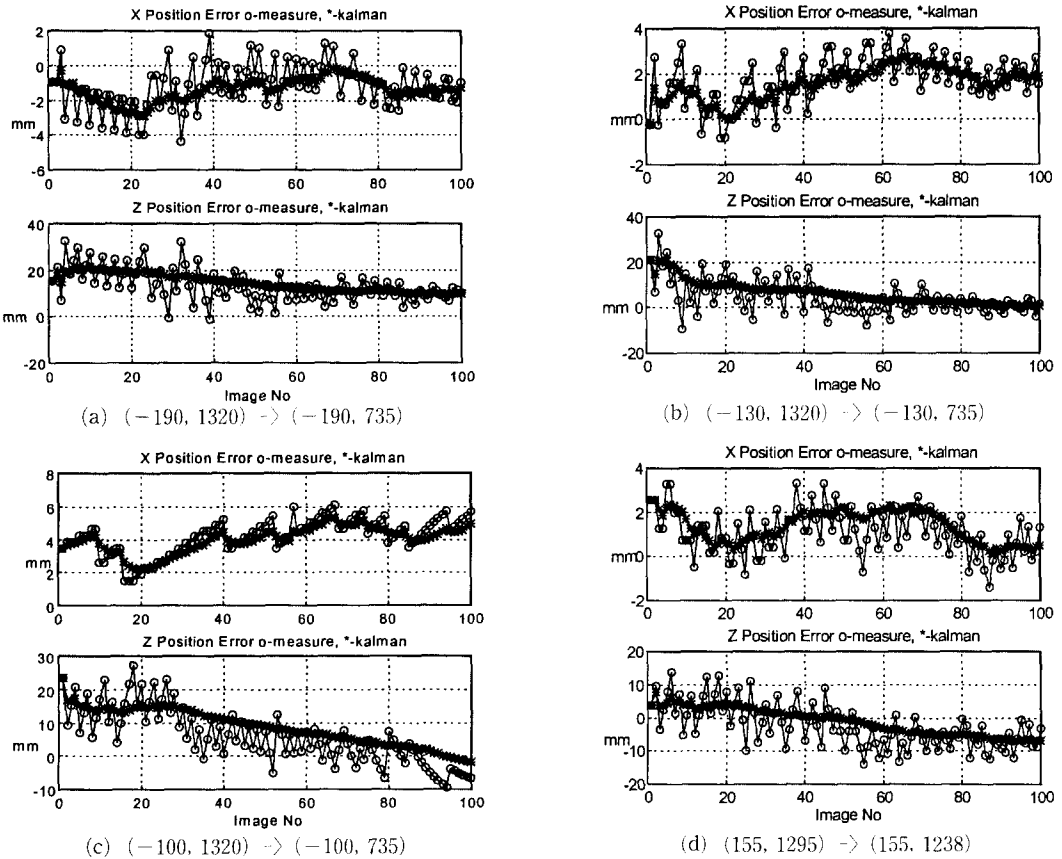


Fig. 5 Obstacle position error in translational motion

4.4 이동물체의 운동과 위치추정실험

Fig. 6(a), (b), (c)은 이 실험에 사용된 영상으로 로봇이 출발할 당시의 스테레오 영상, 정지하였을 당시의 스테레오 영상 및 스캔라인 연속영상을 나타낸다. 바닥 평면에서 수직한 물체가 X방향으로 이동하고 있으며, 로봇은 Z방향으로 이동하고 있을 때 Fig. 6(d)은 물체의 X방향의 운동오차와 Z방향의 운동오차를 나타낸 것이다. 또 Fig. 6(e)는 절대좌표에서의 물체의 이동을 나타낸 것이다. 물체의 위치에 대한 오차는 시각센서로 구한 물체의 위치에 크게 영향을 받는다는 것을 알 수 있었다.

4.5 물체의 2차원 맵작성 실험

로봇의 이동궤적과 지나온 환경에 대한 물체들의 위치를 구할 수 있는 가를 보기 위하여 긴 거리에 대한 실험을 하였다. 실험 환경은 연구실 내의 고정 환경이며, 로봇의 이동명령은 외부에서 RS-232

통신으로 주었다. 이 실험은 온라인(on-line)으로 진행되었으며, 로봇의 이동을 끝낸 순간에 로봇의 과거 운동궤적과 지나온 환경 내의 물체들의 위치를 저장하게 된다. Fig. 7은 로봇이 출발할 당시 위치를 절대좌표의 원점으로 하고 물체들의 위치와 로봇의 진행궤적을 나타낸 것이다.

4.6 다중 매칭문제 및 해결방법

Fig. 8에서 보듯이 근처에 비슷한 점이 있어 매칭점을 잘못찾은 경우에는 실제로 존재하지도 않은 점이 있는 것으로 인식한다. 이런 경우에 카메라를 이동시키면 실제 물체는 카메라의 운동량과 같은 운동이 일어나는 반면에 잘못 매칭된 점은 틀린 운동을 하게 된다. 따라서 카메라의 운동량을 비교적 정확히 알고 있다면 이와 같은 성질을 이용하여 잘못 매칭된 점을 제거할 수 있다. 그러나 카메라의 운동을 정확히 알고 있다고 하더라도 영상의 이산오차때문에 카메라의 이동량이 작다거나 또는 물체

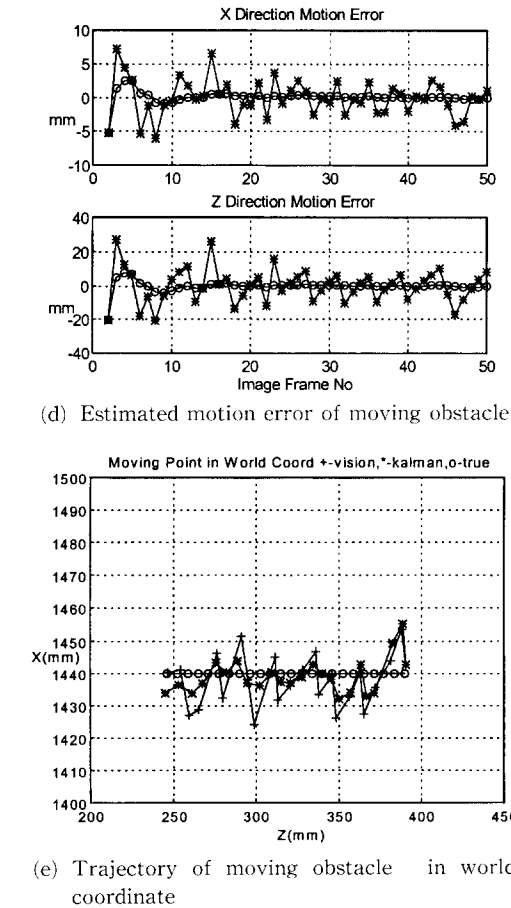
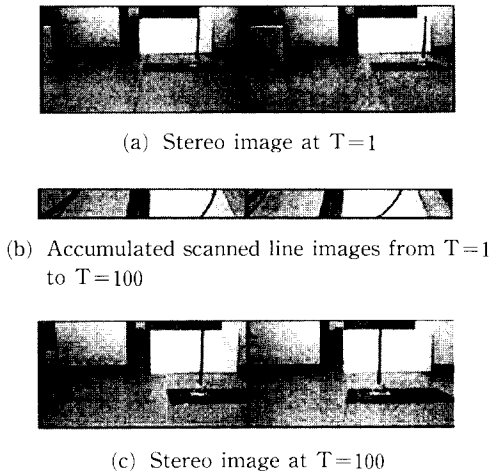


Fig. 6 Pure translational motion and moving obstacle

가 멀리 있는 경우는 구별하기가 어려운 단점이 있다. 따라서 본 논문에서는 비슷한 물체가 존재하여

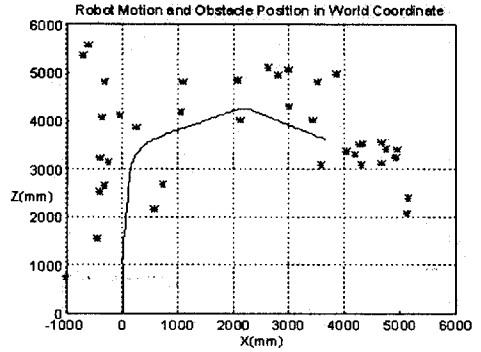


Fig. 7 Results of estimating robot motion and obstacle point in world coordinate, — Robot path, * Obstacle point (0, 0) Point means starting point

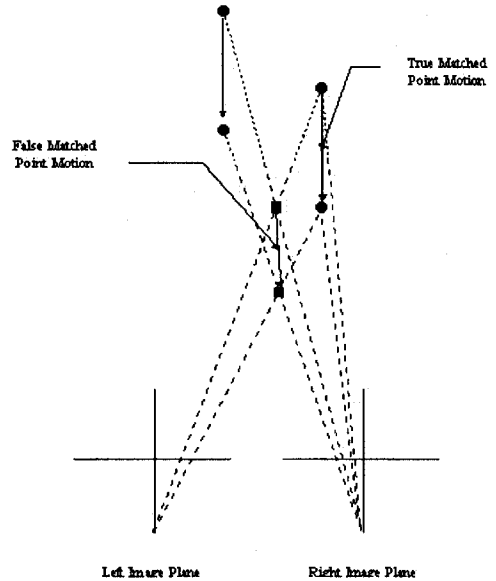


Fig. 8 Motion of true matched point and false matched point in stereo matching, ● — True matched point, ■ — False matched point

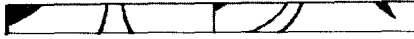
이와 같이 다중 매칭이 되는 경우에 한하여 식 (12) 과 같이 카메라의 운동량과 매칭점의 운동량의 차이가 영상의 이산오차의 영향으로 나타나는 깊이 방향의 오차보다 큰 경우에 잘못 매칭된 것으로 간주한다.

$$|T_{camera} - T_{match}| > \Delta Z_q \quad (12)$$

깊이방향의 이산오차에 의한 오차는 식 (13)으로 구할 수 있다.



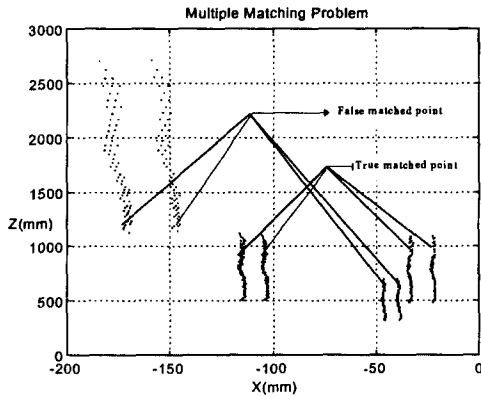
(a) Stereo image at T=1



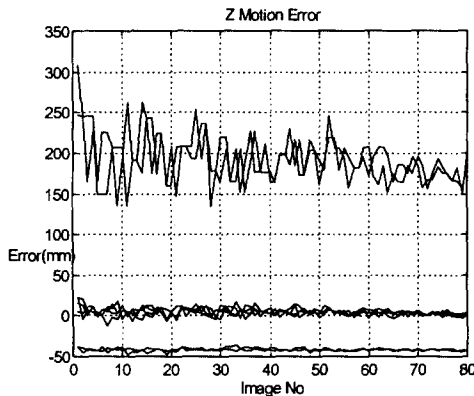
(b) Accumulated scanned line images from T=1 to T=100



(c) Stereo image at T=100



(d) Movement of multiple matched point



(e) Error of depth by quantization effect

Fig. 9 Motion of false matched point and true matched point, in case two similar pole : $T_2 = 166, Z = 1060 \rightarrow 480$

$$Z_{\max} = \frac{S_x B}{d + \Delta d}, \quad \frac{S_x B}{d - \Delta d}$$

$$\Delta Z_q = \Delta Z_{\max} - \Delta Z_{\min} \quad (13)$$

Fig. 9(a), (b), (c)은 다중 매칭이 일어났을 경우 맞게 매칭된 점의 운동과 틀리게 매칭된 점의 운동이 실제로 어떻게 되는가 보기 위하여 카메라로부터 1000 mm 거리에 있는 비슷한 영상을 갖는 두개의 기둥에 대하여 카메라가 이동한 경우에 대하여 다중 매칭이 일어났을 경우 실험을 행한 영상이며, Fig. 9(d)은 로봇이 진행함에 따라서 다중매칭으로 구해지는 물체의 거리를 나타낸다. 그림에서 보듯이 잘못 매칭되는 물체의 움직임이 카메라와 다른 것을 알 수 있다. Fig. 9(e)는 물체의 운동오차를 나타낸 것이다. 그림에서 보듯이 맞게 매칭된 점의 운동오차는 ΔZ_q 의 경계내에 들어오고 틀리게 매칭된 것은 경계 밖에 있게 된다. 여기서 제시한 방법을 이용하면 다중 매칭의 문제를 해결할 수 있음을 알 수 있다. 이 방법은 로봇의 가까운 거리에 있는 물체에 대해서는 ΔZ_q 의 양이 작기 때문에 더욱 좋은 결과를 얻을 수 있다.

5. 결론

본 연구에서는 전체 영상을 이용하여 주위 환경에 대한 정보를 얻는 경우 처리해야 할 정보량이 많은 반면에 시간축으로 이웃한 영상이 대부분이 공통된 영상이라는 것에 착안하여 스캔라인 연속영상을 도입함으로써 전체 영상을 처리한 것과 같은 효과를 얻으면서 빠른 속도로 거리정보를 얻을 수 있었다. 스테레오 카메라를 이용하여 물체까지의 거리를 구하는 경우에 칼만필터를 적용하여 물체의 최적위치를 재귀적(recursive)으로 추정하는 실험을 하였으며, 로봇의 내부 센서를 통해 구한 운동과 시각센서를 통해 구한 운동을 칼만필터를 통하여 로봇의 운동을 추정하였고, 이동중인 물체를 알고 있다는 가정하에 이동물체의 위치와 운동을 칼만필터를 사용하여 추정하였다. 또 고정된 실내 환경에서 로봇의 주행에 필요한 물체들의 맵을 온라인으로 작성할 수 있었다. 스테레오 매칭의 과정시 발생할 수 있는 다중 매칭의 문제를 로봇에 직접적인 영향을 미치는 범위 내에서 카메라 운동관계를 이용하여 해결할 수 있었다. 실험 결과로부터 제안된 스캔라인 연속영상을 이용하면 물체가 바닥으로

부터 수직으로 고정되어 있다는 가정하에서 이동로봇에 적용할 수 있다는 것을 알 수 있었다.

참고문헌

- (1) 정지윤, 1992, "실시간 시각 정보처리기능을 갖는 지능형 이동 로봇에 관한 연구," 전산학과, 석사학위논문, 한국과학기술원.
- (2) 윤동수, 1994, "이동로봇 실내 운항을 위한 다중 센서를 이용한 모델 작성기법에 대한 연구," 전산학과, 석사학위논문, 한국과학기술원.
- (3) 김성호, 1995, "영상 축소와 분산처리를 이용한 이동로봇의 실시간 영상처리," 전기 및 전자공학과, 석사학위논문, 한국과학기술원.
- (4) Ian Horswill, 1994, *Visual Collision Avoidance by Segmentation*, pp. 902~909.
- (5) Ulrich Solder and Volker Graefe, 1993, "Visual Detection of Distant Objects," *In Proc. IEEE. Int. Conf. on Intelligent Robots and Systems*, 26~30, pp. 1042~1049.
- (6) Cho Yong C. and Cho, Hyung S, 1994, "A Stereo Vision-based Obstacle Detecting Method for Mobile Robot Navigation," *Robotics*, Vol. 12, pp. 203~216.
- (7) 이재웅, 1991, "Optical Flow를 이용한 3 차원 거리 정보 추출에 관한 연구," 생산공학과, 석사학위논문, 한국과학기술원.
- (8) Xiaoping Hu and Narendra Ahuja, 1994, "Matching Point Features with Ordered Geometric, Rigidity, and Disparity Constraints," *In Proc. IEEE, Trans. Pattern Analysis and Machine Intelligence*, Vol. 16, No 10, pp. 1041~1049.
- (9) Oliver Faugeras, *Three Dimensional Computer Vision*, MIT Press.
- (10) Robert C. Bolles, Harlyn Baker and David H. Marimont, 1987, "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion," *Int. Journal of Computer Vision*, Vol 1, pp. 7~55.
- (11) Peter S. Meyback, 1982, *Stochastic Models. Estimation and Control*, Academic Press.
- (12) Hans P. Moravec, 1983, "The Stanford Cart and the CMU Rover," *In Proc. IEEE*, Vol. 71, No. 7, pp. 872~884.
- (13) Larry Matthies, 1989, *Dynamic Stereo Vision*, CMU, Ph. D Paper.
- (14) 차영엽, 권대갑, 1995, "레이저거리제를 이용한 자율주행로봇의 국부경로 계획 알고리즘," 대한기계학회논문집, 제19권, 제4호, pp. 887~897.