

실내 주행을 위한 3차원 장애물 검출

고복경^{*,}, 우동민^{**}

* 명지대학교 전기공학과, ** 명지대학교 제어계측공학과

Three Dimensional Obstacle Detection for Indoor Navigation

BokKyong Ko^{*,}, DongMin Woo^{**}

* Electrical Eng., MYONGJI Univ., ** C.I.E. MYONGJI Univ.

Abstract

The vision processing system for mobile robots requires real time processing and reliability for the purpose of safe navigation. But, general types of vision systems are not appropriate owing to the correspondence problem which correlates the points out of two images. To determine the obstacle area, we use correspondences of line segments between two perspective images sequentially acquired by camera.

To simplify the correspondence, the matching of line segments are performed in the navigation space, based on the assumption that mobile robot should be navigated in the flat surface and the motion of mobile robot between two frames should be approximately known.

I. 서론

AGVS(Automatic Guided Vehicle System)은 공장이나 사무실의 자재나 문서의 전달을 위한 이동형 로봇(mobile robot)로서 현재까지 실용화된 시스템들은 대부분 주행을 하기 위한 일정한 지역의 맵(map)을 가지고서, 바닥 면에 부착된 도선이나 반사 테이프(tape)에 의해 유도된다. 그러나 이러한 방식은 로봇이 가지고 있는 맵에 나타나 있지 않은 예기치 않은 장애물(unexpected obstacle)을 피할 수 없으며 복도나 벽의 주변환경을 인식하지 못한다. 시각처리 시스템이 예기치 않은 장애물이나 복도와 벽면과 같은 주변환경을 인식하기 위해서는 3차원 주행 환경을 인식하여야 한다. 3차원 주행환경을 인식하는 방법으로는 2개 이상의 카메라를 사용하는 스테레오 비전[1]-[4]이 일반적으로 널리 사용되며, 1개의 카메라를 이용하는 경우는 카메라가 이동함에 따라 연속적으로 여러 영상을 취득하여 카메라의 움직임과 물체의 형상을 추정하는 방법[5][6]이 있다. 그러나 이러한 방법들에서는 여러 영상들에서 동일한 특징들을 서로 일치시키는 일치성 문제(correspondence problem)에 어려움이 있으며 이로 인해 이동형 로봇에 필요한 실시간 처리, 신뢰성등의 요구 조건을 만족하기 힘들다.

본 논문에서는 이동형 로봇 주행에 적합한 1개의 카메라가 움직이며 연속적으로 영상을 취득하여 3차원 장애물을 인식하는 시각 시스템을 구성하였다. 각 취득된 영상은 그 전에 취득된 영상과 비교되어 환경 인식이 수행되는데 영상중 비교되는 특징을 선소(line segment)로 선정하였다. 이 때 두 영상의 선소들을 위한 시각시스템이 요구하는 실시간 처리 및 신뢰성을 만족 하도록 구성되어야 한다. 따라서 본 논문에서는 이동형 로봇의 대략적인 움직임(motion)이 바퀴의 encoder로부터 제공된다는 가정하에 두 영상의 선소 정합을 간소화하였다.

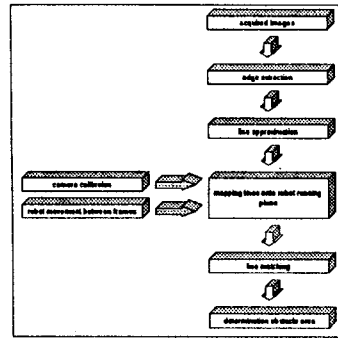


그림 1. 시각 처리 수행 과정

Fig. 1. The procedure of image processing

II. 선소 추출(line segment extraction)

에지 추출

2D 가우시안 필터를 사용하여 노이즈를 제거한 후 필터를 사용하여 영상을 컨빌루션하고 각 픽셀에 대한 방향성분을 계산한다.

$$\vec{n} = \frac{\nabla(G * I)}{|\nabla(G * I)|} \quad (1)$$

방향성분을 고려한 미분을 계산하고 기울기의 크기를 계산한다.

$$\frac{d}{dn}(G * I) = \left(\frac{d}{dn}G\right) * I = \nabla G_n * I \quad (2)$$

$$|\nabla(G_n * I)| \quad (3)$$

이웃들과의(8-connected neighborhood) 관계에 의해서 미분 크기의 극대값(local maxima) 갖는 위치가 에지의 위치가 된다. 그림 2는 이러한 전체적인 에지 추출 과정을 보여주고 있다.

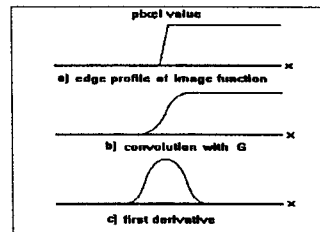


그림 2. 에지의 단면도

Fig. 2. The edge profile of the first derivative of the image function

에지 추적 방법(edge-following algorithm)

이미지에서 에지들의 위치를 찾고나면 이러한 에지들의 위치로부터 물체를 인식하기 위해서 물체의 특징을 이루는 직선이나 곡선과 같은 필요한 정보들을 추출해야 한다. 이웃들과의 관계에 의해서 서로 연결되어 곡선을 이루는 에지들은 다음과 같이 구할 수 있다.

① 전체 영상에 대한 에지의 크기를 이용하여 두 개의 임계값 (T_1, T_2)을 정한다.

$$T_1 = k_1 \cdot \frac{\sum_{x,y} |e(x,y)|}{N}$$

$$T_2 = k_2 \cdot \frac{\sum_{x,y} |e(x,y)|}{N} \quad (4)$$

단, $T_1 < T_2$

② 이웃들 중에서 곡선의 확장을 위한 다음 성분을 구한다. 이웃의 픽셀의 이웃들 중에서 에지의 크기가 T_1 보다 큰 이웃이 존재하는 경우 곡선의 확장을 위한 다음 에지로 선택한다. 만약 이러한 이웃들이 여러개 존재할 경우 각각에 대해서 평가 함수를 계산하고 이 중에서 가장 작은 값을 갖는 에지를 곡선의 확장을 위한 다음 에지로 선택한다. 이때 사용되는 평가 함수(cost function)는 다음과 같이 계산한다.

$$\text{costfunc}(x, y) = |\phi(x, y) - \phi(x, y)| + |\sigma(x, y) + \sigma(x, y)| \quad (5)$$

$e(x, y)$: 에지의 크기, $\phi(x, y)$: 에지의 방향

③ 곡선을 이루는 성분들 중에서 에지의 크기가 T_2 보다 큰 성분이 존재하는지를 확인한다. 만약 이러한 성분이 존재하면 곡선을 이루고 있는 에지들을 실 에지로 판단하고 그렇지 않은 경우 곡선을 이루고 있는 에지들을 노이즈에 의한 성분들로 판단하고 모두 제거한다.

직선 근사화(line approximation)

본 논문의 최종 목적인 장애물의 영역을 검출하기 위해서는 영상에서 검출된 직선들을 이동형 로봇의 주행 평면상으로 변환한 다음 정합하는 과정이 필요하다. 따라서 주행 평면상으로 변환 과정을 단순화하기 위해 이러한 곡선을 직선으로 근사화할 필요가 있다. 그림 3은 이러한 과정을 보여주고 있다.

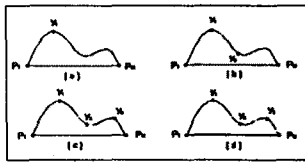


그림 3. 직선 근사화 과정
Fig. 3. The procedure of the line approximation

곡선을 이루는 좌표들의 집합을(sequence) $S = \langle P_1, P_2, \dots, P_n \rangle$ 라고 할 때 S의 성분들 중에서 첫부분(P_1)과 끝부분(P_n)에 해당되는 두 점을 연결하는 직선의 방정식을 구한다. 이 직선과 나머지 성분들과의 거리를 구한다. 이때, 최대 거리가 최대 허용오차보다 크면 그 점을 기준으로 다시 두 개의 곡선으로 나누어진다. 이러한 과정은 다시 두 개의 곡선에 대해서 되풀이 된다.

III 장애물 검출

카메라 보정 및 좌표변환

이동형 로봇트가 바닥이 평평한 실내에서 주행한다고 가정하고, 또 로봇트의 주행정보를 로봇이 알고 있다는 가정하에 하나의 카메라를 이용해서 연속적으로 영상을 취득하여 3차원 장애물을 인식하는 시각인식 시스템이 구성될 수 있다. 그림 4에서 나타난 바와 같이 이동형 로봇트가 주행하는 평면을 $z=0$ 인 실세계 좌표 평면(x, y)로 설정하고 첫 번째 frame에서 취득된 영상좌표를 (u_1, v_1), 로봇의 일정한 주행중에 얻어진 두 번째 frame의 영상좌표를 (u_2, v_2)

라 하자. 첫 번째 frame에서 취득된 영상 좌표의 모든 점들이 실세계 좌표평면 $z=0$ 으로 부터 취득된 점이라는 가정 하에 카메라 보정[8]을 할 수 있고, 보정계수 C_1 를 구할 수 있다. 두 번째 frame에서 취득된 영상좌표의 모든 점들은 보정계수 C_1 에서 단지 로봇의 이동에 따른 외부변수가 변화된 C_2 로 카메라 보정을 할 수 있다. 이때 실세계 평면상의 점($z=0$) P는 C_1, C_2 에 의해 각각 영상의 점 P_1, P_2 로 변환되는데 그 위치는 동일하게 된다. 만일 점 P가 평면 상의 점이 아니라면($z \neq 0$) 다른 위치에 대한 영상으로 나타난다. 따라서 본 논문에서는 취득된 두 개의 영상에서 선소(line segment)를 특징으로 추출하여 이를 주행 평면으로 변환하여 정합하는 방식으로 장애물을 검출하는 방식을 채용하여 정합을 실시간으로 처리되도록 하였다.

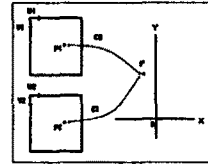


그림 4. 카메라 보정 및 좌표변환
Fig. 4. The camera calibration and coordinate transformation

선소 정합(line segment matching)

선소 정합 과정은 시각차리 과정 중 핵심적인 부분으로서 영상으로 부터 검출된 선소들이 일치하는가를 판별하는 과정이다. 선소 정합 알고리즘은 다음사항에 적합하게 구성되어야 한다.

- 카메라보정 등에서 기인되는 약간의 오차 및 바퀴의 encoder로부터 제공되는 두 영상사이의 움직임의 오차에도 무관하게 두 영상으로부터의 선소들이 정합될 수 있어야 한다.
- 선소들이 전체적으로 일치하는 지를 판별할 뿐 아니라, 선소 추출과정에서 여러 선소들로 끊어진 선소도 적절히 정합되어야 한다.

상기 요건을 만족하는 직선 정합을 위해 사용되는 평가 함수는 그림 5의 a)에서와 같이 한 직선이 다른 직선에 투영하여 나온 면적(S_m)과 거리(P_m)에 관한 함수로 구성된다. 면적 S는 그림 5의 b)에서와 같이 다각형의 삼각형 분할법으로 구할 수 있다.

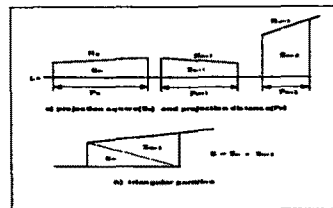


그림 5. 두 직선에 대한 투영면적과 투영거리
Fig. 5. The projection square and the projection distance with respect to two lines

본 논문에서 제시한 조건을 만족하는 직선 정합 과정은 다음과 같다. 만약, L_m 과 R_n 을 각각 연속적으로 취득된 이미지상에 존재하는 직선들의 집합이라고 한다면, $L_m(L_m \in L_N)$ 과 R_n 의 모든 성분들에 대해서 다음과정을 실시한다.

① $R_n(R_n \in R_N)$ 을 L_m 상으로 투영시켜 투영면적과 투영거리의 비율 구한다.

$$\text{투영거리와 투영면적의 비} = \frac{S_m}{P_m} \quad (6)$$

② 만약 식 (7)에서 구한 투영면적과 투영거리의 비가 T_3 보다 작으면 두 직선은 정합되는 것으로 일차 판정한다. 이때, L_m 과 정합 관계에 있다고 판단되는 여러개의 R_n 성분들이 존재할 수 있다. 투

영면적은 카메라 보정시 발생하는 오차의 정도와 관계된다. 만약 카메라 보정시 오차가 발생하지 않는다면 정합관계에 있는 두 직선은 정확히 일치하게 된다. 그러나 오차가 발생하는 경우 정합 관계에 있는 두 직선이 조금 떨어져 있게 되는 것을 알 수 있다. 따라서 T_3 는 다음과 같이 계산된다.

$$T_3 = k_3 \cdot e_2 \quad (7)$$

k_3 : 가중치(weightor)

e_2 : 카메라 보정시 발생하는 카메라의 오차.

③ 일차 판정된 성분들 중에는 실제로 일치되지 않는 직선도 포함된다. 따라서 직선 정합에 의해 정합되는 것으로 일차 판정된 직선에 대해서 투영거리를 이용하여 재차 판정한다.

$$\text{투영거리의 합} = \sum_i p_i \quad (8)$$

$$T_4 = k_4 \cdot (l_i \text{의 길이}) \quad (9)$$

만약 투영거리의 합이 T_4 보다 크면 두 직선은 정합되는 것으로 최종 판정한다.

장애물 영역의 진출

첫 번째 frame에서 취득된 영상좌표의 모든 점들이 좌표평면 $z=0$ 로부터 취득된 점이라는 가정하에 카메라 보정에 의한 변환을 구하였다. 그리고 그 이후에 로봇이 주행하면서 얻어지는 영상좌표는 같은 카메라가 이동한 것이기 때문에 앞에서 구한 보정변수와 일치하고, 다만 위치 이동에 따른 외부변수만 달라지게 된다. 따라서 로봇의 주행 정보를 알고 있다는 가정하에 첫 번째 frame 이후에 취득되는 영상 정보에 대한 보정 계수를 모두 알고 있다고 가정할 수 있고 실제 이를 적용하면 좌표 평면 $z=0$ 로 보정 변환을 할 수 있다. 실제 $z=0$ 인 평면상에 존재하는 모든 점들은 이동형 로봇의 주행 평면상에서 서로 일치하게 된다. 그러나 $z \neq 0$ 인 평면상에 존재하는 점들은 z 값에 비례하는 변이 성분을 가지게 된다. 변이란 실제 좌표에서 같은 점에 해당하는 좌표가 카메라 보정에 의해 변환되었을 때 이미지에서 보여주는 차이를 말한다. 따라서 본 논문에서는 연속적으로 취득된 두 영상의 보정으로 얻어지는 결과들을 서로 일치시켜 정합의 유무로 얻어지는 결과를 가지고서 장애물의 영역을 검출하였다. 이렇게 구하여진 결과로부터 완전한 장애물의 3차원 형태는 얻을 수 없으나 이동형 로봇트가 완전한 주행을 위해 경로를 선택하기는 충분한 정보가 제공된다.

IV. 실험 결과

본 논문에서 제안한 시각처리 과정의 수행을 위해 실제 카메라를 탑재한 이동형 로봇이 임의의 장애물이 있는 복도를 주행하며 취득할 수 있는 영상을 3차원 그래픽스를 이용하여 만들어 실험하였다. 3차원 시뮬레이션 환경은 X-Window의 3차원 전용 라이브러리인 PEXlib[9]를 이용하여 실험하였고, 취득된 영상은 X-Window에서 처리하였다.

장애물 영역의 결정

본 논문에서 제안한 시각처리 시스템에 따라 장애물의 영역을 검출하는 방법은 다음과 같다.

바닥면으로 보정변환된 두 영상을 직선 정합과정에 의해 정합 여부를 결정한다(그림 7). 각 영상에서 보이는 동일한 숫자를 가진 직선들은 서로 일치되는 것으로 판정된 직선들이다. 정합이 이루어지지 않는 직선들 중 장애물의 바닥과 닿아 있는 모서리를 이루는 직선들은 정합은 이루어지지 않지만 바닥면에 닿아 있는 부분은 서로 일치하므로 소멸점이 같다고 할 수 있다. 이를 이용하여 현 시점에서의 장애물의 영역을 구한다.

VI. 결론

본 논문에서의 시각처리 시스템은 근본적으로 경계선의 선소(line segment of edge)를 특징으로 채용하며, 이들을 주행 평면상으로 변환하여 정합하고, 정합 결과에 의해 주행 평면 혹은 장애물상의 선소로 속성을 부여하여 장애물의 영역을 구한다.

이러한 방법은 기존의 방법과는 달리 한 개의 카메라에서 연속적으로 얻은 정보를 효율적으로 활용하여 예기치 않은 장애물 뿐만 아니라 복도나 벽면과 같은 주변환경을 인식할 수 있으며 좌우 두 영상의 특징들만을 주행 평면상의 좌표로 변환하여 비교함으로써 두 영상에 대한 실시간 처리가 가능하게

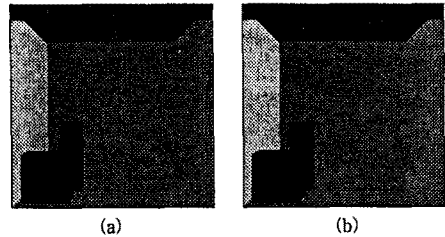


그림 6. 컴퓨터그래픽에 의해 연속적으로 취득한 원 영상
(a) 첫 번째 영상 (b) 두 번째 영상

Fig. 6. The original images acquired by computer graphics continuously

(a) first frame image (b) second frame image

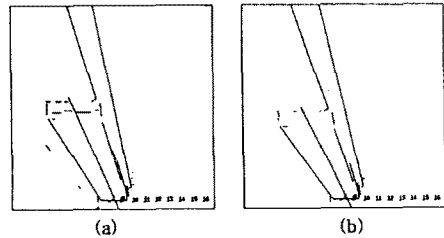


그림 7. 직선 정합 결과 영상

(a) 첫 번째 영상 (b) 두 번째 영상

Fig. 7. The result images of line matching algorithm

(a) first frame image (b) second frame image

된다. 또한 카메라 보정시 약간의 오차에도 불구하고 직선 정합과정에 의해 좌우영상에 대한 직선들을 정합함으로써 장애물 영역을 안정되게 검출할 수 있다. 본 논문에서의 연구는 이동형 로봇트를 위한 카메라 보정 방법, 에지 추출, 직선 정합, 장애물 영역의 검출에 초점을 두었다. 본 연구에서 사용한 선소 정합방법을 확장하게 되면 다른 여러 과제들도 실시간 처리할 수 있는 효율적인 알고리즘을 구성할 수 있을 것이다.

VI. 인용문헌

- [1] Y. Shirai, *Three-Dimensional Computer Vision*, Springer-Verlag: New York, 1987.
- [2] N. Ayache and B. Faugeras, "Efficient registration of stereo images by matching graph descriptions of edge segments," *Int. Journal of Comput. Vision*, vol. 1, 1987, pp.107-131.
- [3] R. Horaud and T. Skordas, "Stereo correspondence through feature grouping and maximal cliques," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-11, 1989, pp.1168-1180.
- [4] N. Ayache and F. Lustman, "Fast and reliable trinocular stereo vision," *Int. Conf. Comput. Vision*, 1987, pp.422-427.
- [5] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images - a review," *Proc. IEEE*, vol.76, no.8, 1988, pp.917-935.
- [6] T.S. Huang and A.N. Netravali, "Motion and structure from feature correspondences : a review", *Proc. IEEE*, vol.82, no.2, 1994, pp.252-268.
- [7] K. Storjohann, Th. Zielke, H. A. Mallot and W. von Seelen, "Visual obstacle detection for automatically guided vehicles," *IEEE Conf. on Robotics and Automation*, pp. 761 - 766, 1990.
- [8] 우동민, "렌즈의 왜곡모형을 이용한 카메라 보정에 관한 연구," 전자공학회논문지, 31권 B편 2호, 1994.
- [9] Tom Gaskins, "PEXlib Programming Manual - 3D Programming in X," O'Reilly & Associates, Inc. 1992.