

2차원 레이저 레인지 파인더를 이용한 빠른 로컬 실내 지도 제작

Fast Local Indoor Map Building Using a 2D Laser Range Finder

°최 응*, 고낙용**, 최정삼***

* 조선대학교 대학원 제어계측공학과(Tel : 062-230-7108; E-mail : cie98@shinbiro.com)

** 조선대학교 공과대학 제어계측공학과(Tel : 062-230-7108; E-mail : nyko@mail.chosun.ac.kr)

*** 조선대학교 공과대학 산업공학과(Tel : 062-230-7123; E-mail : jschoi@mail.chosun.ac.kr)

Abstract

This paper proposes an efficient method constructing a local map using the data of a scanning laser range finder. A laser range finder yields distance data of polar form, that is, distance data corresponding to every scanning directions. So, the data consists of directional angle and distance. We propose a new method to find a line fitting with a set of such data. The method uses Log-Hough Transformation. Usually, map building from these data requires some transformations between different coordinate systems. The new method alleviates such complication. Also, the method simplifies computation for line recognition and eliminates the slope quantization problems inherent in the classical Cartesian Hough transform method. To show the efficiency of the proposed method, it is applied to find a local map using the data from a laser range finder PLS(Proximity Laser Scanner, made by SICK).

1. 서론

이동 로봇 주행은 환경의 지식에 의존한다. 따라서 대부분의 로봇은 그들의 지도를 개선하거나

또는 발생시키기 위해 지도 제작 임무를 실행한다. 지도를 표시하기 위해, 메모리 셀에 데이터를 저장하는 그리드방법과 환경 특징을 피팅하는 피쳐드로잉 방법이 많이 쓰이고 있다. 그리드 방법은 초음파 센서의 데이터처럼 데이터의 양이 적고, 불확실한 데이터의 양이 많을 때 메모리 셀에 데이터를 저장해 환경을 인식하는 방법이고, 피쳐드로잉은 레이저 레인지 파인더나 비전 센서처럼 데이터의 양이 많고 빠른 계산 시간을 요구할 때 쓰이는 방법이다[1].

최근 몇 년 동안 매우 빠르고 정확한 레이저 레인지 파인더가 대부분 주행 임무에서 이동 로봇에 중요한 센서가 되었다. 레이저 레인지 파인더는 환경 데이터를 초음파 센서에 비해 정밀하고, 빠르게 얻을수 있기 때문에 이동 로봇의 장애물 회피, 위치 파악과 지도제작에 많이 이용되고있다. 그리고 마지막 두 문제는 빠르고 정확한 계산을 요구한다, 왜냐하면 이런 문제는 이동로봇에 계산 시간과 메모리량의 부하가 크고 실-시간 처리가 필요하기 때문이다. 그리고 지도를 제작해 환경 특징을 인식하기 위해, 현재 방법들은 레이저 레인지 파인더에 스캔된 데이터를 한 점 씩 직각좌표계로 변환하여 환경 특징을 추출하였다.

이 논문에 제시된 새로운 접근법은 레이저 레인지 파인더의 극 좌표 계 안에서 로컬 지도를 제작하는 방법을 제시한다.

2. 일반적인 선 감지 알고리즘

레이저 레인지 파인더 데이터를 기반으로 한 이동 로봇의 로컬 주행 시 주요한 문제는 레이저 레인지 파인더에서 얻어진 데이터를 이동 로봇 주행에 이용하기 알맞은 지도로 바꾸는 것이다. 기존에 개발된 몇몇 알고리즘은 극 좌표계 데이터를 직각 좌표계 데이터로 변환해 지도를 제작하는 것이다.

Line tracking: 이 알고리즘은 스캔된 데이터를 연속적으로 처리하기 위해 직선 방정식을 만든다. 이 방정식은 가상 선 위에 발견된 점들을 반복적으로 계산한다. 선에 대한 각각 점의 에러 추정은 최소 자승법에 의해 계산된다. 그때, 만약 현재 검사된 점이 선 위에 있다면 문턱값에 의해 검사된다. 선에서 검사된 점까지의 거리가 문턱값보다 클 때, 선은 그 점에서 종결되고, 새 선이 시작된다[2].

Iterative endpoint fit: 선분은 데이터의 처음과 마지막을 연결해 만든다. 선분에 대해 가장 큰 수직거리를 갖는 점을 찾는다. 만약 수직 거리가 문턱값보다 크면, 선분은 두 부분으로 쪼개진다. 새롭게 발견된 점과 이전 선분들의 끝점을 연결한다. 그리고 위의 과정을 반복해 문턱값을 넘지 않는 직선들을 만든다[3].

3. 호프 변환을 이용한 선 감지

이 방법은 비눗방울 방(bubble chamber)로부터 획득된 사진들에서 소립자의 복잡한 경로를 자동적으로 감지하기 위해 호프(Hough)에 의해 1962에 제시되었다[4]. 기본 아이디어는 한 점을 통과하는 무수히 많은 선들의 집합 파라미터 공간으로 변환하는 것이다. 이 파라미터 공간이 호프 공간이다. 여기 제시된 호프 변환은 일반적으로 직선을 감지한다[5, 6]. 이 방법은 임의의 영상 요소를 감지하기 위해 이용될 수 있다. 호프 변환은 이동 로봇 분야에서 지도 제작과 주행에서 로봇의 위치 평가에 이용되어왔다.

3.1 표준 호프 변환

직선의 표현식은

$$y = mx + c, \left(\frac{x}{a} + \frac{y}{b} = 1 \right) \quad (1)$$

$$\rho = x \cos \theta + y \sin \theta \quad (2)$$

$$\rho = r \cos(\alpha - \theta) \quad (3)$$

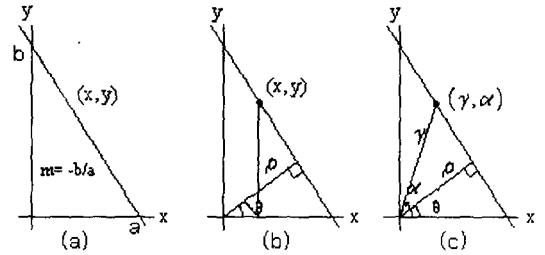


그림 1. 선을 표시하는 방법들

Fig. 1 Methods representing a line

여기서 $m(-b/a)$ 은 기울기, c 는 y 축의 절편, ρ 는 원점에서부터 직선에 그은 수선의 길이, θ 는 x 축과 수선과의 각도이다. m, c 또는 ρ, θ 는 직선을 결정하는 매개변수이므로 직선검출은 입력 데이터 (x, y) 또는 (γ, α) 들이 일정한 매개변수를 이루고 있는가를 알아보고 그 매개변수를 찾아내는 작업이다. 식(1)을 사용할 때는 수직선($m \rightarrow \infty, b \rightarrow \infty$)와 같은 표현할 수 없는 경우가 있으므로 잘 쓰이지 않는다. 식(2)와 식(3)은 모든 직선을 표현할 수 있고, 데이터로 주어진 점의 수가 적을 경우는 두 점을 지나 모든 직선의 매개변수(ρ, θ)를 계산하여 호프 공간에 사상시킨다. 이 방법은 계산할 때마다 \cos, \sin 함수를 사용해야 하므로 효율적이지 못하다. 실제 계산할 때에는 모든 점에 대하여 그 점을 지나 모든 직선(각도의 분해능에 의해 결정되는 증분에 따름)의 ρ, θ 를 구하고 호프 공간에 누적시킨다. 누적 값이 주어진 문턱 값 이상이면 직선으로 결정하는 작업을 표준 호프 변환(SHT)이라 한다.

그림 1(c)와 같이 극 좌표계로 주어진 γ 와 α 을 이용한 식(3)의 경우도 표준 호프 변환으로 검출할 수 있으나 불필요한 삼각함수 호출을 하지 않고 $\cos(\alpha_i), \sin(\alpha_i)$ 를 참조 테이블로 만들어 연산시간을 줄일 수 있다. 그러나 ρ 는 연산할 때

마다 구해야 한다. 이 방법을 이용해, 임의의 동일 선상의 데이터들을 연결하는 직선에 대한 방정식은 모든 점들을 호프 공간으로 변환해 발견할 수 있다. 그리고 결과적으로 코사인 곡선의 양의 교차점을 결정한다.

만약 교차점 탐색이 정해진 곡선 수보다 더 많은 교차점을 찾는 것을 문턱 값으로 제한한다면, 호프 변환은 노이즈에 대해 매우 강하다. 그러나 표준 호프 변환 구현 시 메모리 사용량이 많고 컴퓨터 계산에서 매우 시간이 걸린다. 메모리 사용량이 많은 이유는 교차점 탐색을 촉진하기 위해 모든 직선(각도의 분해능에 의해 결정되는 증분에 따름)의 ρ , θ 를 구하고 호프 공간에 누적시킨다. 컴퓨터 계산 시간 량이 많은 이유는 모든 한 점에 대해 식(3)으로부터 코사인 곡선의 재 계산이 필요한데 있다.

3.2 로그-호프 변환

식(3)의 양변에 로그를 취하면

$$\log \rho = \log \gamma + \log \cos(\alpha - \theta) \quad (4)$$

$$\log \gamma = \log \rho - \log \cos(\alpha - \theta) \quad (5)$$

여기서 $u = \log \gamma$, $v = \alpha - \theta$ 라 하면

$$u = \log \rho - \log \cos v \quad (6)$$

이고, $\log \rho$ 와 θ 는 직선 파라미터로서 일정하며 v 의 구간은 $-\pi/2 < v < \pi/2$ 이다[2, 7, 8]. u 와 v 의 관계를 직각 좌표계에 나타내면 그림. 2(a)와 같으며 그림. 2(b)의 곡선을 로그선(Log line)이라 한다.

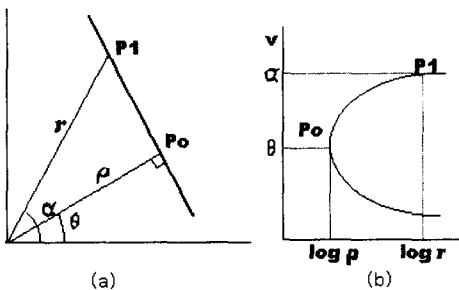


그림 2. 선과 이것의 로그 선
Fig. 2 A line and its log line.

극좌표계 γ , α 와 직각 좌표계 u , v 와의 관계를

좀 더 확실히 이해할 수 있도록 하기 위해 그림. 3을 제시한다. 그림 3에서 P_0P_1 , P_2P_3 는 u 축에 평행이 되고, 원호 P_0P_2 , P_1P_3 는 v 축에 평행이 된다.

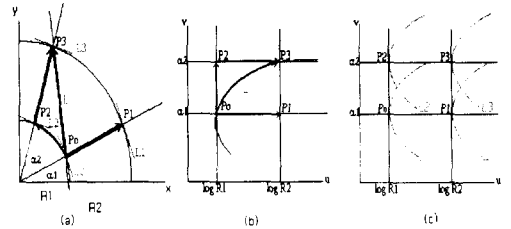


그림 3. 극 좌표계에서 곡선은 uv 평면에 로그선으로 사상된다.

Fig. 3 Curves in polar coordinate are mapped to uv plane to produce log lines.

다시 말하면 거리나 각도가 일정한 데이터는 (u , v)평면에서 축에 평행인 직선이 되고 그 이외의 직선은 P_0P_3 과 같은 곡선 L 이 된다. 그림. 3(a)에서, 각이 α_1 이고 수선의 길이가 각각 R_1 , R_2 인 두 직선 L_0 , L_1 와 두 직선이 α_2 까지 회전한 두 직선 L_2 , L_3 의 로그선(Log line)을 구하면 그림. 3(c)와 같다. L_0 , L_3 은 각이 $\Delta\theta = \alpha_2 - \alpha_1$ 만큼 회전하고 거리는 $k = R_2 / R_1$ 배로 변환된 직선이며, (u , v)평면에서, u 축으로 $\Delta u = \log k = \log R_2 - \log R_1$ 만큼 이동하였고, v 축으로 $\Delta v = \Delta\alpha$ 만큼 이동하였다. 그러나 곡선의 형태는 전혀 변하지 않는다. 그러므로 어느 위치에 있는지 간에 한 직선의 로그선(Log line)만 구한다면 어떤 형태의 직선도 식(5)를 이용하여 간단하게 표현할 수 있다. 이때 좌표의 평행이동 변환을 해주어야 한다. 그림에서 보면 직선 매개변수 ρ 와 θ 는 로그 선(Log line)의 정점(apex)에 사상된다. 한 점을 지나는 모든 직선의 ρ 와 θ 값의 사상을 구하면 그림. 4와 같다.

식(4)에서 γ , α 는 일정하므로 $u = \log \rho$, $v = \alpha - \theta$ 라 하면

$$u = \log r + \log \cos v \quad (7)$$

이 되어 우변 두 번째 항의 부호만 바뀌었으므로 P_0 점에 대하여 로그 선(Log line)과 대칭 되는 곡선이 된다. 그림. 4(a)에서 P_1 점이 존재한다면 P_1 점을 지나는 모든 직선 중에서 k 번째의 직선과 P_0

를 지나는 L_k 가 일치하게 되므로 P_0 와 P_1 을 지나는 직선의 매개변수는 ρ_k 와 θ_k 이다. 이와 같이 모든 점을 지나는 직선의 매개변수를 계산하고 동일한 값의 출현횟수가 일정한 임계값 이상인 것을 찾아서 직선의 매개변수로 판정한다.

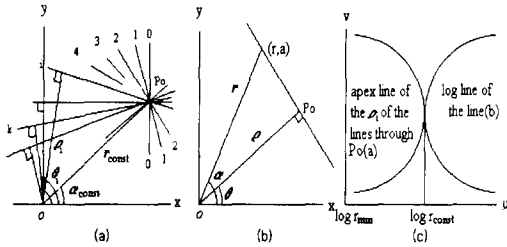


그림 4. 정점 선과 로그 선
Fig. 4 Apex line and log line.

3.3 선 연결

직선의 매개 변수가 구해지면 스캔된 데이터 위로 많은 선들이 그려진다. 데이터 군의 시작점과 끝점은 각 선과 그 선 위에서 발견되는 점과의 거리를 구해 문턱값을 넘으면 선을 종료하고 다시 새로운 선분을 만들어 데이터에 선분을 피팅한다.

4. 실험 및 고찰

로그 호프 변환 알고리즘은 레이저 레인지 파인더에서 데이터를 얻어 실험하였다. 먼저 레이저 레인지 파인더 하드웨어의 관해 간단히 설명한다.

4.1 실험에 이용된 레이저 레인지 파인더

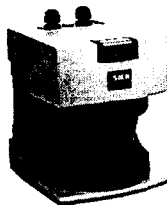


그림 5. SICK社의 레이저 레인지 파인더
Fig. 5 The Laser range finder from SICK
그림 5는 실험에 이용된 레이저 레인지 파인

더이다. Sick社에서 제작한 PLS 101-112이다. 이 레이저 레인지 파인더는 0.5° 각 분해능과 최대 50m까지 측정거리에서 $\pm 5\text{cm}$ 의 오차를 갖는 타임 오브 플라이트(Time of flight)측정을 수행한다. 레이저 안전 등급은 1등급으로 인체에 무해하고, 최근 RWI社 및 Nomadic社의 상업용 이동 로봇의 옵션으로 장착되고 있는 장비이다.

4.2 로그 호프 변환을 이용한 직선 추출

SICK PLS 레이저 레인지 파인더를 이용해 실험실의 환경 데이터를 얻었다. 센서의 특징상 $0^\circ \sim 180^\circ$ 범위의 데이터를 0.5° 간격으로 얻었다. 실험에 이용된 데이터는 361개로 10번 스캐닝하여 얻은 값의 평균값을 이용했다. 그림 6을 보면 왼쪽과 오른쪽은 벽이고 앞에는 두 개의 기둥이 있다. 데이터의 거리 측정 단위는 cm이다. 획득된 데이터의 최대 거리는 591cm이고 최소 거리는 195cm이다.

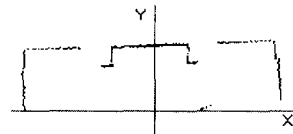


그림 6. 레이저 레인지 파인더로부터 획득된 데이터

Fig. 6 Data obtained by Laser range finder
그림 6에서 획득된 데이터를 로그 호프 변환에 적용해 환경을 인식 할 수 있는 선의 매개 변수를 찾아냈다. 그림 7은 로그 변환의 실행결과를 보이고 있다.

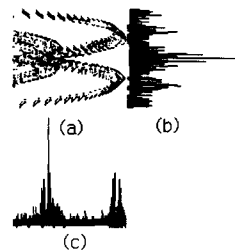


그림 7. 직선을 로그 호프 변환으로 검출하는 과정
Fig. 7 Processing of Log-Hough transform on the lines

그림 6의 데이터를 로그 호프 변환한 정점 선 (apex line)들의 누적된 그림은 그림 7(a)에 보였고 그림 7(b)는 방위각에 따른 누적 값의 히스토그램이고 그림 7(c)는 거리에 따른 누적 값의 히스토그램이다. 누적된 피크값의 첨예화를 위해 로그 호프 공간에 3×3 소벨(sobel)연산자를 적용해 첨예도를 높였다. 누적된 값의 최대값은 36이고, 로그 호프 변환을 수행하는데 걸린 시간은 0.0549초였고 문턱 값은 10으로 설정하였다. 문턱 값이 10일 때 얻어진 직선 매개변수를 그림 8과 같이 그렸다.

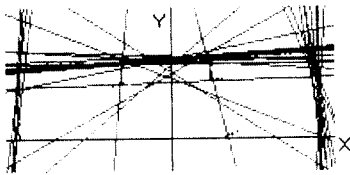


그림 8. 로그 호프 변환에 의해 얻어진 선들

Fig. 8 Lines obtained by Log-Hough Transform

얻어진 직선 매개 변수를 이용 환경을 인식할 수 있는 선을 찾기 위해 선 트래킹(Line tracking) 방법을 이용했다. 직선 위에 발견되는 점과 직선사이의 거리를 구해 문턱 값을 넘으면 직선을 그 점에 종료하고 다음 점부터 선을 찾는 방법으로 환경에서 획득된 데이터를 선 피팅하였다. 버블 쇼팅을 하여 누적된 값이 많은 매개변수부터 그려지도록 하였고 문턱 값은 5로 설정해 피팅하였다. 그림 9는 스캔된 데이터의 피팅된 결과를 보인다. 원 모양은 SICK PLS 센서를 설치한 위치이다. 누적된 값이 많은 매개변수들인 경우에는 버블 쇼팅을 하고 선 피팅한 경우가 오차 값이 적었고, 누적된 값이 적은 매개 변수들인 경우 버블 쇼팅을 하지 않고 선 피팅한 경우가 오차 값이 적었다.



그림 9. 선 피팅

Fig. 9 The line fitting

4.3 불확실성의 계산

피팅된 선이 스캔된 데이터와 얼마나 오차가 있는지 계산해 보았다. 그림 10과 같이 원 데이터를 통과하는 선은 직선 방정식 파라미터 (ρ, θ)의 관계식에 의해 오차가 구해질 수 있다. 그리고 실제 그림 9에서는 점과 직선사이의 거리를 구해 오차를 구했다.

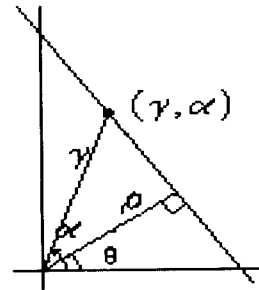


그림 10. 극 좌표계에서 선을 설명하는 파라미터

Fig. 10 A line in polar coordinates with the specifying parameters

$$\epsilon_\gamma = \gamma - \frac{\rho}{\cos(\alpha - \theta)} \quad (8)$$

$$\epsilon_\alpha = \alpha - \theta - \arccos\left(\frac{\rho}{\gamma}\right) \quad (9)$$

식 (8), (9)는 거리와 각도의 오차를 구하는 식이고

$$\mu_\gamma = \frac{1}{n} \sum_{i=1}^n \epsilon_\gamma(i) \quad (10)$$

$$\mu_\alpha = \frac{1}{n} \sum_{i=1}^n \epsilon_\alpha(i) \quad (11)$$

식 (10), (11)은 거리와 각도의 오차 평균을 구하는 식이고

$$\sigma_\gamma^2 = \frac{1}{n} \sum_{i=1}^n (\epsilon_\gamma(i) - \mu_\gamma)^2 \quad (12)$$

$$\sigma_\alpha^2 = \frac{1}{n} \sum_{i=1}^n (\epsilon_\alpha(i) - \mu_\alpha)^2 \quad (13)$$

식 (12), (13)은 거리와 각도의 오차 공분산을 나타낸다.

표 1에 극 좌표계에서의 거리와 각도의 오차와 직각좌표계에서 거리 오차를 제시했다. SICK PLS 센서의 거리오차가 50m거리 범위에서 $\pm 5\text{cm}$ 이다. 표 1에서 직각 좌표계에서 거리오차가 $2.5 \pm 1.4\text{cm}$ 이고, 극좌표계에서 거리오차가 $-0.41 \pm 3.7\text{cm}$ 이다. 이 오차는 그림 9와 같은 선 피팅을 신뢰

할 수 있게 한다.

표 1. 불확실성 계산

Table 1. Uncertainty computation

문턱값: 5, 버블 소팅후 피팅						
거리오차 평균	-0.406	각도오차 평균	0.245	수직거리오차 평균	2.508	직선수
거리오차 분산	13.880	각도오차 분산	13.589	수직거리오차 분산	2.045	80
거리오차 표준편차	3.725	각도오차 표준편차	3.686	수직거리오차 표준편차	1.430	계산 시간 101ms

5. 결론

빠른 로컬 지도 작성을 위해 레이저 레인지 파인더 데이터를 직각 좌표계 데이터로 변환하지 않고 직선을 검출하는 방법을 제시하였다. 로그 호프 변환을 이용한 직선 검출은 거리와 방위각으로 주어진 한 직선의 로그 선(log line)이나 정점 선(apex line)의 형상이 거리나 방위각에 따라 변하지 않기 때문에 단 한번의 계산으로 구해진 하나의 표준 정점 선의 평행 이동만으로 로그 호프 변환이 수행되어 직선 검출 시간이 단축된다. 그러나 로그 호프 공간에 피크 값이 문턱값 이상인 것들이 단일 점이 아니면 여러 개의 직선으로 인식되므로 피크 값을 침예화 하는 필터의 개발이 필요하고, 본 논문에서는 침예도를 높이기 위해 3×3 소벨 연산자를 적용하였다. 로그 호프 변환의 경우 데이터가 각도에 대한 거리의 형태로 이루어질 때 효율적인 방법으로 비디오 카메라에서 얻은 데이터의 경우에는 일반적인 호프 변환(SHT)과 계산 시간에서 별 차이가 없을 것으로 판단된다. 하지만 레이저 레인지 파인더와 초음파 어레이 레인지 센서를 장착한 이동 로봇의 로컬 지도 작성에는 계산 시간과 메모리 사용량의 측면에서 좋은 방법으로 판단된다.

앞으로 이동 로봇의 주행 시 주변 로컬 지도를 제공하는데 빠른 로그 호프 변환을 적용할 것이다.

참고문헌

- [1] T. Tsubouchi, : " Nowadays trends in map generation for mobile robots," Proc. of the 1996 IEEE International conference on Intelligent Robots and Systems, pp. 828-833, 1996.
- [2] B. Giesler, R. Graf and R. Dillman, "Fast Mapping Using the Log-Hough Transformation," IROS'98, Victoria, B. C. Canada, oct. 1998.
- [3] W.K. Pratt, "Digital image processing," A wiley-interscience publication, John Wiley & Sons Inc., 1991.
- [4] P.C.V. Hough, "Method and Means for Recognizing Complex Patterns," U.S. Patent No. 3,069,654. Technical report, HelpMate Robotics Inc., Dec. 1962.
- [5] P.E. Hart and R.O. Duda, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," In Communications of the ACM, Vol. 15, pp. 199-218, 1972
- [6] P.E. Hart and R.O. Duda, "Pattern Classification and Scene Analysis." John Wiley and Sons, 1973
- [7] C.F.R. Weiman, "Ploar Exponential Sensor Arrays Unify Iconic and Hough Space Representation," SPIE Syposium an Advances in Intelligent Robotics System, Nov. 1989
- [8] C.F.R. Weiman, "Application of Log-Hough Transform to Lidar Navigation," Phase I SBIR Final Report, NSF Award No. III-9361990. Technical report, HelpMate Robotics Inc., 1994.