

직선적합을 이용한 고속의 직선 성분 추적기

이진한, 장국현, \*서일홍  
한양대학교 전자컴퓨터통신공학과

e-mail : spacetrain@naver.com, imzgx@incorl.hanyang.ac.kr, ihsuh@hanyang.ac.kr

A fast Visual Line Segment Tracker Using Line Fitting

Jin Han Lee, Guoxuan Zhang, \*Il Hong Suh  
Department of Electronics & Computer Engineering, Hanyang Univ.

**Abstract** - 본 논문에서는 기울기 극대점을 검색하고 직선적합을 통해 (Line fitting) 실내 환경에서 적용이 가능한 고속의 직선 성분 추적 알고리즘을 제안한다. 단순히 일반적인 직선 검출기(Hough Transform 혹은 Douglas Peucker 알고리즘 등)를 적용해서는 카메라의 잡음 등의 이유로 연속적인 영상에서 안정적으로 직선 정보를 획득하기가 어렵다. 이를 위해 계산상의 비용이 크지 않으면서도 실효적인 알고리즘을 고안하게 되었는데 연속되는 영상에서 직선 성분을 검출하고 만약 어떤 직선 성분이 이전 영상에서는 검출 되었으나 현재 영상에서는 검출 되지 않았을 때에는 기울기 극대점 검색을 통해 직선 성분을 찾아 대응관계를 구하게 된다.

1. 서 론

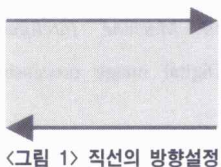
요즘 대부분의 SLAM(Simultaneous Localization and Mapping) 기법들은 랜드마크(Landmark)로서 주로 SIFT나 SURF, Fast corner detector와 같은 알고리즘을 통해 추출한 점 정보를 사용한다. 그런데 건물이나 가정의 실내와 같은 경우에는 점 정보뿐만 아니라 직선 성분 역시 환경을 표현하는 훌륭한 특징으로서 오히려 점 정보에 비해 상대적으로 적은 수로 훨씬 더 넓은 공간을 표현할 수 있는 등의 장점이 있다[3, 4]. 하지만 직선 성분을 일반적인 SLAM 기법에 적용하기 위해서는 점 정보와 마찬가지로 연속되는 각 이미지 상에서 안정적으로 직선성분을 추출하고 추적하며 대응관계를 구하는 과정이 필요하다. 영상에서 직선성분을 찾아내기 위해서는 일반적으로 기울기극대점(Gradient maxima point)을 찾는 방법을 사용하는데 이를 위해 대표적으로 허프변환(Hough transform)이 사용된다. 하지만 이렇게 획득한 선정보들은 연속적인 영상에서는 카메라의 움직임, 잡음, 불안정한 조명 등의 영향으로 안정적이지 못하다. 예를 들어 카메라와 대상물체 모두 정지한 경우에도 연속적인 영상에서 허프변환 등의 알고리즘을 통해 직선성분을 추출하면 추출되는 직선의 길이가 매우 불안정하게 변화하는 것을 볼 수 있는데 이 역시 안정적으로 직선성분을 추적하는데 어려움을 유발하는 요인이 된다. 이를 해결하기 위해 P. Neubert 등은 EKF(Extended Kalman Filter)를 적용한 모션모델을 이용하여 현재 프레임에서의 직선의 위치를 추정하고 추정된 위치에서의 지역적인 기울기 극대점 검색 방법(Local gradient maxima search)을 통해 직선 후보점을 검출하고 여기에 다시 카이자승검증법( $\chi^2$  test)을 적용해 직선성분을 복원, 추적하는 방법을 제안하였다[1].

본 논문에서는 [1]의 기본적인 개념을 따라 지역적인 기울기 극대점 검색(Local gradient maxima search)을 활용하되 SLAM의 실시간 성능을 해치지 않도록 계산상의 비용이 큰 EKF와 카이자승검증법을 사용하지 않고 안정적으로 직선성분을 추적하는 알고리즘을 고안하였다. 즉, 이전 프레임과 현재 프레임에서 안정적으로 추출된 직선 성분에 대해서는 간단한 병합검사(Merging test)를 통해 대응여부를 판단하고 이전 프레임에서는 추출되었으나 현재 프레임에서는 추출되지 않은 직선성분들에 대해서는 현재 프레임에서 이전 프레임에서의 해당 직선의 위치를 기준으로 일정 범위내의 지역적인 기울기 극대점 검색을 통해 직선 후보점을 찾아 직선적합을 통해 직선을 복원하고 여기에 다시 이전 프레임에서의 직선과의 병합검사를 통해 대응여부를 판단한다.

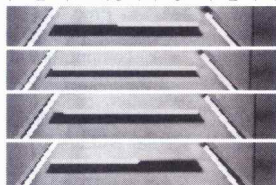
2. 본 론

2.1 직선성분의 표현

본 논문에서는 양 끝점과 직선이 x축과 이루는 각도  $\theta$ , 원점과의 거리  $\rho$ 를 갖는  $6 \times 1$ 의 벡터로서 직선을 표현한다. 즉  $l_i = [x_{i1} \ y_{i1} \ x_{i2} \ y_{i2} \ \theta_i \ \rho_i]$  이 된다. 이렇게 직선을 양 끝점으로 표현하는데 있어 <그림 1>과 같이 직선의 왼쪽에 밝은영역이 오도록 지정하였는데 이렇게 함으로써 후에 있을 병합검사를 통한 대응관계판단에서 제일 먼저 두 직선의 방향을 비교함으로써 비록 두 직선의 위치가 가깝더라도 방향이 다르면 서로 대응되지 않도록 할 수 있다.



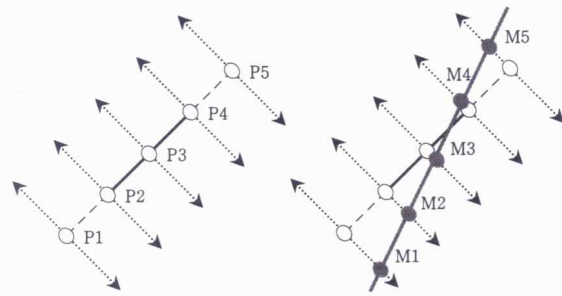
<그림 1> 직선의 방향설정



<그림 2> 직선의 불안정한 검출

2.2 허프변환을 통한 Seed 직선의 검색

카메라를 통해 획득한 영상에 먼저 가우시안필터를 적용하여 카메라 잡음의 영향을 줄인다. 이후 캐니엣지검출기(Canny edge detector, [5])를 적용하여 윤곽선에 해당되는 픽셀들을 찾고 허프변환을 이용해 Seed 직선 성분을 추출하게 된다. 앞서 언급했듯이 이 직선들은 연속적인 영상에서는 양 끝점이 불안정하게 검출되거나 아예 검출이 되지 않기도 하는데 <그림 2>에서 확인 할 수 있다.



(a) 직선과 검색 기준점 P1~P5 (b) 기울기극대점검색을 통해 찾은 극대점 M1~M5

<그림 3> 검색기준점 설정과 기울기 극대점 검색

2.3 기울기 극대점 검색

이전 프레임에서는 검출되었으나 현재 프레임에서는 검출되지 않은 직선성분을 찾기 위해 기울기 극대점 검색을 하게 된다. 기울기 극대점검색의 과정은 다음과 같다.

- 1) 먼저 찾고자 하는 직선에 대해 해당 직선이 마지막으로 검출되었던 프레임에서 <그림 3>의 (a)와 같이 일정 개수의 검색 기준점을 직선상에 균등하게 할당하고 각 기준점에서의 기울기의 방향(Gradient direction)을 기억한다.
- 2) 현재 입력된 프레임에서 찾고자 하는 직선의 위치가 이전 프레임에서 직선이 검출되었던 위치와 큰 차이가 없다면 가정하여 현재 프레임에 해당 직선의 기준점을 설정하고 <그림 3>의 (b)와 같이 직선에 수직인 두 방향으로 1픽셀씩 옮겨가며 기울기 극대점을 찾는다. 본 논문에서는 소벨연산자(Sobel operators - <표 1>, [6])를 이용하여 기울기값을 구하여 극대점을 찾았고 검색범위는 5픽셀로 하였다.
- 3) 검색을 통해 찾은 기울기 극대점의 방향과 1)에서 기억했던 기준점의 기울기 방향을 비교하여 그 차이가 일정 값(eg.  $90^\circ$ ) 이하이고 극대점의 기울기 값이 일정 기준값(eg. 50) 이상인 경우에 해당 극대점을 직선후보점 집합에 추가한다. 이 부분에서 [1]의 방법과 큰 차이를 갖게 되는데 [1]에서는 직선에 수직인 양방향으로 검색을 하여 하나의 기준점에 두 개의 기울기 극대점을 할당하고 이렇게 구한  $2n$ 개의 극대점을 직선후보점 집합에 추가한다.
- 4) [1]에서는  $2n$ 개의 직선후보점에 대해 카이자승검증법을 적용하여 직선을 복원하는 반면 본 논문에서는 총  $n$ 개 이하의 직선후보점을 갖게 되고 여기에 직선적합을 적용하여 가장 유사한 직선성분을 복원하게 된다.

-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

<표 1> 소벨연산자(Sobel operators)

2.4 병합검사를 통한 대응관계 판단

이전 프레임에서 추출한 직선성분들의 집합과 현재 프레임에서 추출한 직선성분들의 집합(허프변환을 통해 뽑은 Seed 직선성분과 기울기극대점검색으로 복원한 직선성분)의 원소들을 1대1로 비교하고 병합검사를 적용하여 대응관계를 판단한다. 여기에서, 비교할 두 직선성분의 위치변화가 크지

\* 교신저자 : 서일홍, ihsuh@hanyang.ac.kr

않다는 가정을 하게 되는데 30fps의 고속의 연속된 영상에서는 일반적으로 타당한 가정이라는 것을 알 수 있다. [2]의 방법을 이용하여 병합검사를 하는데 세 가지의 경우로 나뉘게 된다. 먼저 <그림 4>의 (a)의 경우,  $X_G$ 축 상에서 두 직선(이전 프레임에서의 직선과 현재 프레임에서의 직선)의 일부가 겹치고  $Y_G$ 축 상의 거리가 기준값(e.g. 5) 이하일 경우 둘을 동일 직선으로 판단한다. (b)의 경우,  $X_G$ 축 상에서 두 직선이 완전히 겹치고  $Y_G$ 축 상의 거리가 기준값(e.g. 5) 이하일 경우 둘을 동일 직선으로 판단한다. 나머지의 경우에는  $X_G, Y_G$  두 축상에서 모두 겹치는 부분이 없으므로 서로 다른 직선으로 판단한다. 이를 수식으로 정리하면 다음과 같다.

$$x_G = \frac{l_i(a_x + b_x) + l_j(c_x + d_x)}{2(l_i + l_j)} \quad (1)$$

$$y_G = \frac{l_i(a_y + b_y) + l_j(c_y + d_y)}{2(l_i + l_j)} \quad (2)$$

$(x_G, y_G)$  : 새로운 좌표계 G의 중심점,  $l_i, l_j$  : 각 직선의 길이  
 $a_x, b_x, c_x, d_x$  : a, b, c, d의 x좌표,  $a_y, b_y, c_y, d_y$  : a, b, c, d의 y좌표

$$\text{새로운 좌표계의 각도} : \theta_r = \frac{l_i\theta_i + l_j\theta_j}{l_i + l_j} \quad (3)$$

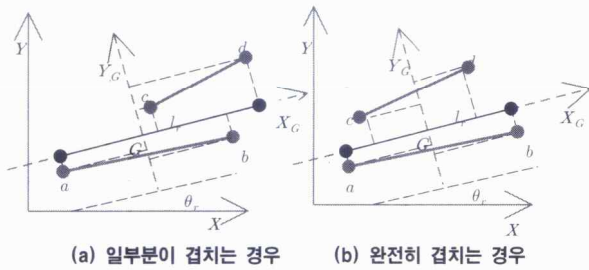
임의의 점  $\delta$ 의 좌표계 G에서의 좌표 :

$$\delta_{X_G} = (\delta_y - y_G)\sin\theta_r + (\delta_x - x_G)\cos\theta_r \quad (4)$$

$$\delta_{Y_G} = (\delta_y - y_G)\cos\theta_r - (\delta_x - x_G)\sin\theta_r \quad (5)$$

**Case 1**  $|l_r| = \sqrt{a_{X_G} b_{X_G}}$  or  $|l_r| = \sqrt{c_{X_G} d_{X_G}}$  일때,  
 $|\delta_{Y_G} \kappa_{Y_G}| \leq d_{MAX_{Y_G}}$  를 만족하면 두 직선은 동일 직선으로 판단한다.  
 $\delta_{Y_G} \kappa_{Y_G}$  : a, b, c, d의  $Y_G$  축 상의 가장 먼 두 좌표  
 $d_{MAX_{Y_G}}$  : 두 직선이  $Y_G$  축 상에서 최대로 떨어질 수 있는 기준값

**Case 2**  $|l_r| < (\sqrt{a_{X_G} b_{X_G}} + \sqrt{c_{X_G} d_{X_G}})$  이고  
 $|l_r| \neq \sqrt{a_{X_G} b_{X_G}}$ ,  $|l_r| \neq \sqrt{c_{X_G} d_{X_G}}$  이면 두 직선들의 일부가 겹치는 경우가 된다. 이때  $|\delta_{Y_G} \kappa_{Y_G}| \leq d_{MAX_{Y_G}}$  를 만족하면 역시 두 직선을 동일 직선으로 판단한다.

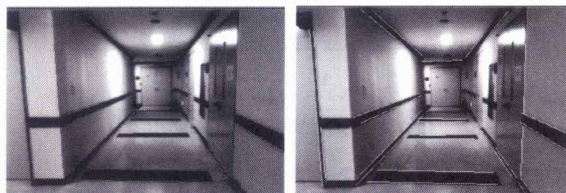


(a) 일부분이 겹치는 경우 (b) 완전히 겹치는 경우  
 <그림 4> 병합검사의 두가지 경우

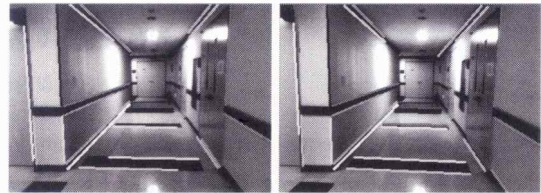
**2.5 실험결과**

본 논문이 제안하는 방법을 실험하기 위해 한양대학교의 IT/BT관 복도에서 얻은 두 개의 연속영상(각 <그림 6>, 1800프레임과 <그림 7>, 1500프레임)을 사용하였다. 미국 Mightex社의 CGE-C013-U 카메라와 AZURE社의 0420MM 렌즈를 사용하여 30fps, 320×240의 영상을 획득하였고 Windows XP, Pentium(R) Dual-Core E6300, 2GB 램을 장착한 데스크탑 PC에서 Microsoft Visual C++ 로 구현하고 실험하였다.

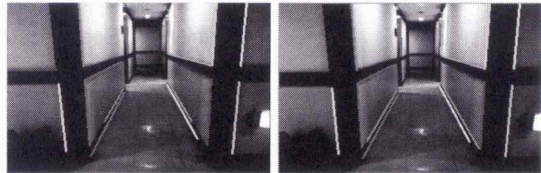
<그림 5>의 (a)는 허프변환으로 검출한 Seed 직선을 나타내고 (b)는 이를 기반으로 10프레임동안 기술기극대점검색과 직선적합을 통한 복원만으로 검출한 직선을 나타내고 있는데 오차가 크지 않은 것을 볼 수 있다. 실제 추적기에서는 <그림 5>의 (b)와 같이 모든 Seed 직선에 대해 직선복원을 수행하는 것이 아니라 이전 프레임에서는 검출되었으나 현재의 프레임에서는 검출되지 않은 직선에 대해서만 추적을 위해 직선복원과정을 수행하므로 추적기 전체적으로 고속을 보장한다. 실험 결과 영상에서 검출되는 대부분의 직선이 안정적으로 추적되고 한 영상을 처리하는데 평균 11ms 미만으로 소요되는 것을 확인할 수 있었는데 이는 <그림 8>을 통해 확인할 수 있다.



(a) 허프변환으로 검출한 Seed 직선 (b) 10프레임동안 기술기극대점검색과 직선적합을 통한 복원만으로 검출한 직선  
 <그림 5> Seed 직선과 직선적합으로 복원한 직선



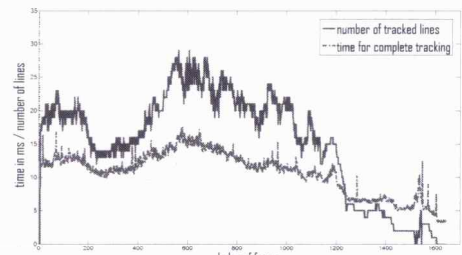
<그림 6> 복도1 연속영상과 추적 결과



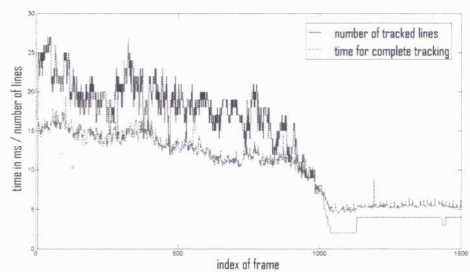
<그림 7> 복도2 연속영상과 추적 결과

**3. 결 론**

본 논문에서는 기술기 극대점 검색을 이용한 고속의 직선성분 추적기를 제안하였다. 기존의 직선성분추적기에 비해 계산상으로 비용이 크지 않으면서도 효율적으로 동작이 가능한데 이는 곧 일반적인 SLAM기법들의 실시간 성능을 해치지 않음으로써 기존의 점정보 위주의 SLAM 대신 선정보를 이용한 SLAM을 가능케 하고 이에 상대적으로 적은수의 특징들로 훨씬 더 넓은 공간을 표현할 수 있게 되는 것이다. 향후 더욱 강인한 추적기로 발전시키기 위해 효율적인 모션모델을 적용하고 직선들간의 대응관계를 판단하는데 있어 좀 더 효과적인 디스크립터에 대한 연구를 진행할 예정이다.



(a) 복도1, average time per image : 10.75ms



(b) 복도2, average time per image : 10.56ms

<그림 8> 연속영상에 대한 실험결과

**[참 고 문 헌]**

[1] P. Neubert, P. Protzel, T. Vidal-Calleja, S. Lacroix, "A fast visual line segment tracker," in Proc. of Emerging Technologies and Factory Automation, September 15-18, Hamburg, Germany, 2008.  
 [2] J. M. Tavares, A. J. Padilha, "A new approach for merging edge line segments," In 7 Congresso Portuguls de Reconhecimento de Padres, Aveiro, Portugal, 1995.  
 [3] J. Solà, T. Vidal-Calleja, M. Devy, "Undelayed initialization of line segments in monocular SLAM," in Proc. of The IEEE/RSJ International Conference on Intelligent Robots and Systems, October 11-15, St. Louis, USA, 2009.  
 [4] G. Zhang, I. H. Suh, "SoF-SLAM: Segments-on-Floor-based Monocular SLAM," in Proc. of The IEEE/RSJ International Conference on Intelligent Robots and Systems, Taiwan, 2010.  
 [5] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679-698, November, 1986.  
 [6] R. C. Gonzalez, R. E. Woods, "Digital image processing," Prentice Hall, second edition, 2001.