Journal of Institute of Control, Robotics and Systems (2011) 17(2):145-148
DOI:10.5302/J.ICROS.2011.17.2.145 ISSN:1976-5622 eISSN:2233-4335

전방향 구조광 영상을 위한 Bresenham 래스터 알고리즘 기반 영상 탐색 방법

Image Search Method Based on Bresenham Raster Algorithm for Omnidirectional Structured Light Image

신 진, 이 수 영*
(Jin Shin[1] and SooYeong Yi[1])
[1]Seoul National University of Science and Technology

Abstract: In this paper, we proposed a search method for structured light pixels of omnidirectional structured light image. Since the omnidirectional structured light image is composed of several circular arc segments, the proposed algorithm searches the structured light pixels in radial direction rather than horizontal or vertical directions. The proposed search algorithm is based on the well-known Bresenham raster algorithm for line drawing in discrete integer space, thereby computation of the algorithm is very efficient. Comparison results between the proposed search algorithm and the conventional horizontal search are presented in experiments.

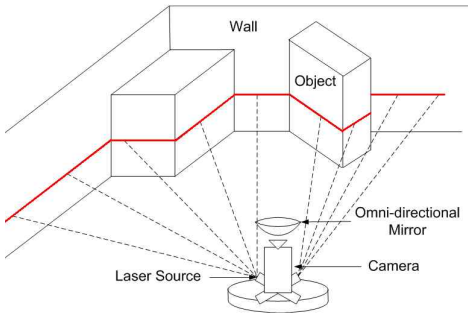
Keywords: omnidirectional structured light image, bresenham raster algorithm, image feature point search

I. 서론

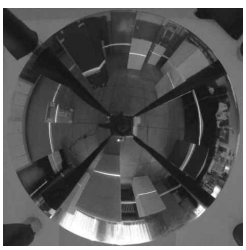
근래들어 보안감시, 이동 서비스로봇, 비디오 화상회의등 분야에 많이 활용되고 있는 전방향(omnidirectional) 영상은 한 장의 영상에 360° 모든 방향의 주변 이미지를 담는 것으로 일반적으로 화각이 100° 이하인 기존의 카메라에 비해서 훨씬 많은 정보를 포함할 수 있다는 장점이 있다[1,8]. 이러한 전방향 영상은 보통 곡면형 거울과 기존의 카메라를 이용한 catadioptric (catoptric(reflective)+dioptric(refractive)) 영상시스템을 통해 얻을 수 있다. 한편 카메라 센서와 구조광을 이용한 거리측정 시스템은 상대적으로 경제적이고, 거리측정 효율이 높기 때문에 초음파, 레이저 스캐너등을 대체할 수 있는 거리센서가 될 수 있다[2].

특히 [3,4]에서는 그림 1에 설명한 바와 같이 catadioptric 전방향 영상시스템과 구조광을 이용하여 360° 모든 방향의 거리를 한 번에 측정할 수 있는 구조광 영상 기반 전방향 거리측정 시스템을 개발한 바 있으며, 그림 1(b)와 같은 구조광 영상에서 배경영상을 삭제하고 구조광 화소만을 추출하기 위한 전처리 과정으로서 영상 차적분 알고리즘[5]이 제안되었다.

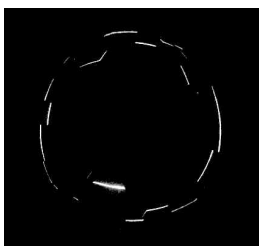
그림 1(c)와 같이 얻어진 구조광 영상에서 물체까지의 거리를 측정하기 위해서는 구조광 화소를 탐색하기 위한 영상처리과정을 거쳐야 한다. 영상 탐색은 일반적으로 수평 방향, 수직 방향, 혹은 대각선 방향의 탐색 방법이 알려져 있다[2]. 기존 카메라를 통해 얻은 수직원근 영상에 대해서는 이러한 탐색방법이 적합하지만, 그림 1(c)와 같은 전방향 구조광 영상에 대해서는 단순 수평, 혹은 수직 방향 탐색 보다는 그림 2에 설명한 바와 같이 광축을 기준으로한 방사향 탐색이 구조광 화소 검출 성능면에서 보다 적합하다고 할 수 있다.



(a) Omnidiectional ranging system based on structured light image.



(b) Omnidirectional structured light image.



(c) Integration of differential structured light image.

그림 1. 전방향 구조광 영상 시스템 및 구조광 영상.

Fig. 1. Omnidirectional structured light image system and structured light image.

* 책임저자(Corresponding Author)
논문접수: 2010. 11. 25., 수정: 2010. 11. 26., 채택확정: 2010. 12. 23.
신 진, 이수영: 서울과학기술대학교 전기공학과
(gomlands@naver.com/suylee@snut.ac.kr)
※ 상기 논문은 제어·로봇·시스템학회 전북제주지부와 광주전남지부와의 학술대회에서 초안이 발표되었음.

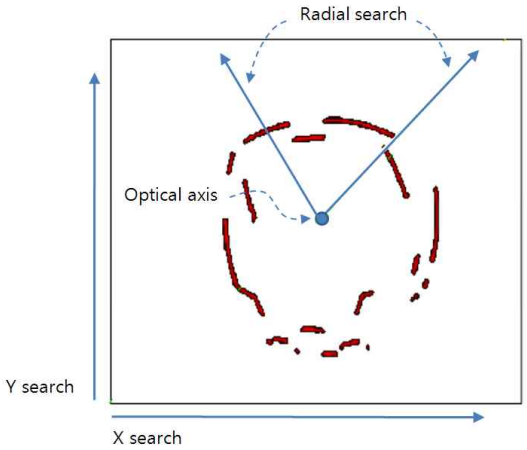


그림 2. 전방향 구조광 영상에 대한 방사향 탐색.
Fig. 2. Radial search for omnidirectional structured light image.

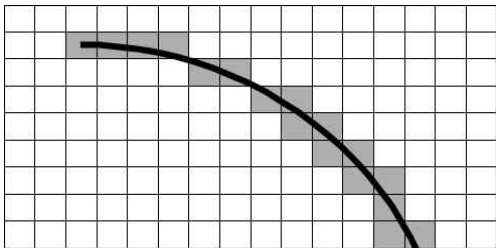
본 논문에서는 Bresenham의 직선 그리기 알고리즘[6,7]을 원용하여 전방향 구조광 영상에 적합한 방사향 탐색 알고리즘을 제안하고, 그 효용성을 보이고자 한다. 컴퓨터 그래픽스 분야에서 일반적으로 사용되는 Bresenham 알고리즘은 주어진 시작점과 끝점을 잇는 직선을 그리기 위한 것으로 정수형 덧셈, 뺄셈과 같은 단순한 연산들만으로 이루어져 있기 때문에 매우 계산 효율이 높은 것으로 알려져 있다. 제안하는 방사향 탐색 알고리즘은 일반적인 수직, 혹은 수평 탐색 알고리즘에 비해 구조광 화소 검출에 매우 유리하며, 또한 높은 계산 효율을 갖는다.

II. Bresenham 알고리즘 기반 전방향 구조광 영상 탐색

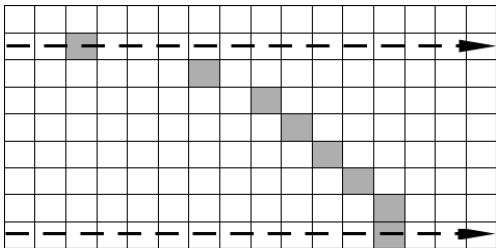
카메라 영상은 단위화소들로 이루어진 양자화된 공간이므로 주로 원호들로 구성되는 전방향 구조광 영상에 대한 단순한 수평, 혹은 수직 방향 탐색은 많은 구조광 화소들의 경계점 정보를 놓칠 수 있다. 그림 3에서 전방향 구조광 영상에 대한 수평 탐색과 방사향 탐색의 결과를 비교한다. 그림 3(a)와 같은 원호 형태의 영상에 대해서 경계점을 찾기 위해 수평방향으로 탐색하는 경우에 얻어지는 결과는 그림 3(b)와 같게 된다. 여기서는 설명을 간단히 하기 위해 이진화 영상에 대한 수평방향 미분 연산을 가정하였다. 한편, 그림 3(c)에서 보이는 방사향 탐색 방법에서는 방향각 샘플링 간격 설정에 따라 방향각 해상도를 임의로 설정할 수 있으며, 탐색결과로서 원호를 구성하는 모든 화소들을 경계점으로 얻을 수 있다.

전방향 구조광 영상에 대해서 방사향으로 경계점을 탐색하기 위해서는 광축을 기준으로 하나의 방향각에 대해서 방사향으로 영상을 탐색하고, 이 과정을 360^o 모든 방향각에 대해서 반복해야 한다. 방향각, θ에 대한 방사향 탐색 알고리즘으로 그림 4와 같은 기본적인 방식을 생각할 수 있다.

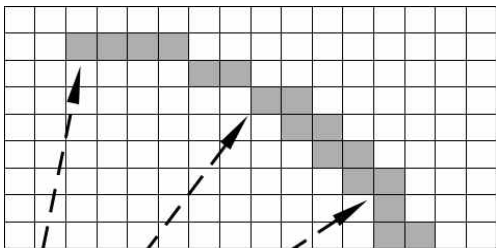
여기서 (x_s, y_s)는 방사향 탐색의 시작점으로서 영상면에서 광축의 위치 (x_c, y_c)로 설정하며, 영상의 크기는 640×480로 가정하여 반지름 r의 최대 길이를 $\sqrt{320^2+240^2} = 400$으로 간주하였다. 그리고 $round(\cdot)$



(a) Circular arc segment of omnidirectional structured light image.



(b) Result of horizontal search.



(c) Result of radial search.

그림 3. 전방향 구조광 영상에 대한 경계점 탐색 결과.
Fig. 3. Result of feature point search for omnidirectional structured light image.

```
for (r = 0; r < 400; r += Δr)  {
    x = x_s + r cosθ;
    y = y_s + r sinθ;
    getpixel(round(x), round(y));  }
```

그림 4. 기본적인 방사향 탐색 알고리즘.
Fig. 4. Basic radial search algorithm.

은 인자를 정수로 변환하는 함수를 나타내며, $getpixel(\cdot,\cdot)$는 위치 $(\cdot,\cdot)$에 해당하는 화소 값을 얻어내는 함수를 의미한다. 이 방식은 $\sin\theta$와 $\cos\theta$에 부동 소숫점이 포함되므로 많은 계산량이 필요하고, Δr의 크기 설정에 따라 하나의 화소를 중복 탐색하거나, 혹은 탐색과정에서 빠트릴 수 있으므로 탐색이 제대로 이루어지지 않을 수 있다. 이러한 문제들을 해결하기 위하여 본 논문에서는 Bresenham 직선 그리기 알고리즘에 기반한 방사탐색 알고리즘을 제안한다.

Bresenham 직선 알고리즘은 원래 이산화된 정수형 공간상에서, 주어진 임의의 시작점 (x_s, y_s)과 끝점 (x_e, y_e)을 잇는 직선을 그리기 위한 것이다. 기울기 m이 $0 \le m = \dfrac{y_e - y_s}{x_e - x_s} \le 1$인 경우, x축을 기준으로 주어진 x_s에서 시작

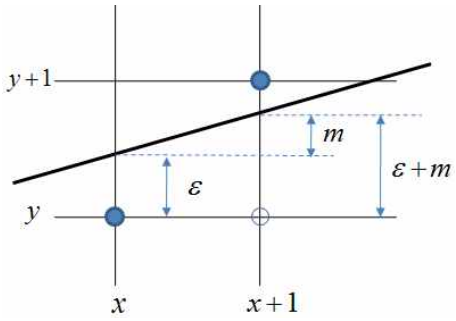


그림 5. Bresenham 직선 그리기 알고리즘.
Fig. 5. Bresenham line drawing algorithm.

```
ϵ = 0, y = y_s;
for ( x = x_s; x < x_e; x++ )  {
    putpixel(x, y);
    if(ϵ + m < 0.5 )
        ϵ = ϵ + m;
    else
        y = y + 1, ϵ = ϵ + m − 1;
    }
```

그림 6. Bresenham 직선 그리기 알고리즘의 의사코드.
Fig. 6. Pseudo code for Bresenham line drawing algorithm.

하여 1씩 증가시키면서 y축 오차, ϵ를 누적하고, 누적오차를 최소화할 수 있는 화소를 선택하여 점을 찍어나간다(그림 5). 그림 6은 $0 \le m \le 1$인 경우 Bresenham 직선 그리기 알고리즘을 의사코드(pseudo code)로 나타낸 것이다.

여기서 $putpixel(\cdot,\cdot)$은 $(\cdot,\cdot)$의 위치에 점을 찍는 함수를 의미한다. 원래의 Bresenham 알고리즘은 직선을 그리기 위한 용도로 개발되었으므로 점을 찍기 위해 $putpixel(\cdot,\cdot)$과 같은 함수를 사용하였으나, 탐색을 위한 용도로는 앞의 그림 4에서 처럼 $getpixel(\cdot,\cdot)$ 형식의 함수를 사용하면 된다. 이 알고리즘은 정수 덧셈, 뺄셈 연산만으로 구현할 수 있기 때문에 계산면에서 매우 효율적인 알고리즘이다. 주어진 시작점과 끝점까지의 기울기가 $m > 1$인 경우는 $0 \le m \le 1$인 경우의 직선 $y = x$에 대한 대칭이므로 위 그림 6의 알고리즘에서 x와 y를 바꿔서 y축을 기준으로 하여 같은 과정을 적용할 수 있다. 또한 기울기 m이 음수인 경우에 대해서도 위 알고리즘을 간단히 수정하여 적용할 수 있다[6,7].

이제 Bresenham 직선 그리기 알고리즘을 이용한 전방향 영상 탐색 방법에 대하여 그림 7에서 설명한다. 영상의 크기는 앞에서와 마찬가지로 640×480으로 가정한다. 영상면에서 광축 (x_c, y_c)가 주어졌을 때, 한 방향각 θ에 대해서 Bresenham 직선 탐색을 위한 시작점과 끝점은 각각 다음 식을 만족해야 한다:

$$\tan\theta = \frac{y_e - y_c}{x_e - x_c} = \frac{y_c - y_s}{x_c - x_s} \tag{1}$$

직선의 기울기가 $0 \le m \le 1$, 즉 $0^o \le \theta \le 45^o$인 경우에는 Bresenham 알고리즘에서 x축 방향으로 탐색하게 되므로 탐색 구간은 $0 \le x \le 639$이고, 따라서 x_s와 x_e는 각

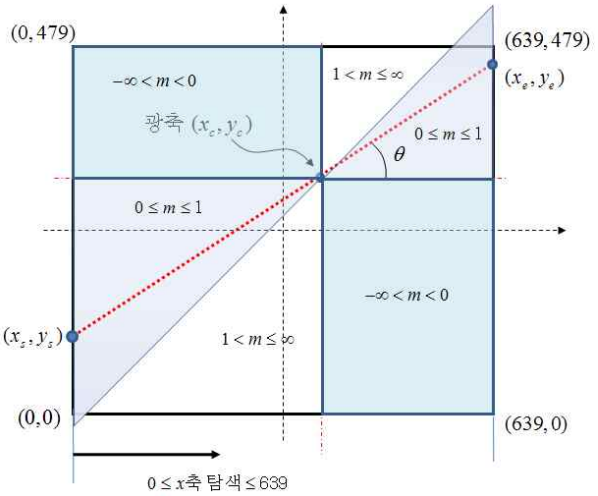


그림 7. Bresenham 직선 알고리즘을 이용한 방사향 탐색.
Fig. 7. Radial search based on Bresenham line drawing algorithm.

각 $x_s = 0$, $x_e = 639$이어야 한다. 그러므로 윗 식에서 y_s와 y_e는 다음과 같다:

$$\begin{aligned} y_s &= y_c - x_c\tan\theta \\ y_e &= y_c - (639 - x_c)\cdot\tan\theta \end{aligned} \tag{2}$$

물론 y의 범위가 $0 \le y \le 479$ 사이이고 또한 정수이어야 하므로, 결국 Bresenham 직선 탐색의 시점과 종점은 다음과 같이 된다:

$$\begin{aligned} (x_s, y_s) &= (0, \max(0, iy_s)), \\ (x_e, y_e) &= (639, \max(479, iy_e)) \end{aligned} \tag{3}$$

여기서 iy_s와 iy_e는 각각 $iy_s = round\{y_c - x_c\tan\theta\}$, $iy_e = round\{y_c + (639 - x_c)\tan\theta\}$, 즉 식 (2)의 값들을 정수 변환한 값을 의미한다.

직선의 기울기가 $1 < m < \infty$, 즉 $45^o < \theta \le 90^o$인 경우에는 y축 방향 탐색에 맞추어 유사한 방법으로 탐색의 시작점과 끝점을 구하면 된다. 또한 $m < 0$인 경우도 마찬가지다. 정리하면, 방향각 $-90^o \le \theta \le 90^o$ 범위 내에서 $\Delta\theta$의 샘플링 단위로 모든 θ에 대해서 (3)과 같이 직선의 시작점과 끝점을 구하여 Bresenham 직선 알고리즘에 따라 탐색하는 것이다.

III. 실험결과 및 분석

본 절에서는 전방향 구조광 영상에 대한 방사향 탐색 영상처리의 구조광 화소 검출 성능과 분석결과를 보인다. 그림 8은 그림 1(c)의 전방향 구조광 영상에 대해서 기존의 수평 방향 탐색에 의한 구조광 화소 탐색 결과와 본 논문에서 제안한 방사향 탐색에 의한 결과를 비교하여 보여준다. 그림 8(a)의 수평 방향 탐색 결과 영상이 전체적으로 흐려 보이는 것은 탐색결과로 찾은 구조광 화소들이 조밀하지 않기 때문이다. 특히 그림 8(a)에서 점선 타원으로 표시된 부근들에서는 많은 구조광 화소들을 놓치고 있음을 볼 수 있다. 반면에 그림 8(b)에서 보이는 방사향 탐색결과

(a) Horizontal search.

(b) Radial search.

그림 8. 탐색 결과.
Fig. 8. Search result.

는 전체적으로 구조광 화소 영상이 밝으며, 그림 1(c)에 보이는 원래 영상의 구조광 화소들을 충실히 반영하고 있음을 볼 수 있다. 여기서 방사향 탐색의 방향각 샘플링 간격은 $\Delta\theta = 0.1^o$로 설정하였다.

그림 4와 같은 기본적인 알고리즘에서는 하나의 직선에 대해서 400번의 for 루프 반복이 필요하므로 방향각을 $\Delta\theta = 0.1^\circ$ 단위로 샘플링하는 경우에 360° 전체로 총 $400 \times 360 \times 10(0.1^\circ) = 1{,}440{,}000$번의 반복 과정이 필요하다. 반면에 본 논문에서 제안한 방사향 탐색 알고리즘에서는 $|m| \le 1$인 경우에는 $0 \le x \le 639$ 범위에서 탐색하므로 $90^\circ \times 10 \times 640 = 576{,}000$번, 그리고 $|m| > 1$인 경우에 $0 \le y \le 479$ 범위에서 탐색하므로 $90^\circ \times 10 \times 480 = 432{,}000$번으로 총 1,008,000번의 반복이 필요하게 된다. 그러므로 그림 4의 기본 알고리즘에 비해서 정수연산 뿐만 아니라, 적은 반복연산으로 인해 계산 효율이 매우 높다는 장점이 있다.

IV. 결론

본 논문에서는 전방향 구조광 영상으로부터 구조광 화소들의 경계점을 추출하기 위한 방사향 탐색 알고리즘을 제안하였다. 전방향 구조광 영상은 원형(circular)이므로, 영상면에서 구조광 화소들의 경계점을 찾기 위해서는 기존의 수평, 혹은 수직방향의 탐색 보다는 광축을 중심으로 한 방사향 탐색이 적합하다. 본 논문에서 제안한 방사향 탐색 알고리즘은 Bresenham 래스터 알고리즘을 기반으로 단위화소들의 배열로 이루어진 정수형 공간에서 360° 모든 방향을 탐색하기 위한 것으로 정수연산과 적은 반복탐색 횟수로 인해 계산효율이 매우 높다는 장점이 있다.

샘플 전방향 구조광 영상에 대한 실험을 통해 본 논문에서 제시한 방사향 탐색 알고리즘이 기존의 수직, 혹은 수평 탐색 알고리즘에 비하여 구조광 화소들의 경계점 추출 성능면에서 매우 우수함을 확인하였다.

참고문헌

[1] S. Nayar, “Catadioptric omnidirectional camera,” *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* pp. 482-488, 1997.

[2] R. Jain, R. Kasturi and Brian G. Schunck, *Machine Vision,* McGraw-Hill, 1995.

[3] I. Joung and H. Cho, “An active omnidirectional range sensor for mobile robot navigation,” *Control Engineering Practice,* vol. 6, no. 3, pp. 385-393, 1988.

[4] S. Yi, B. Choi and N. Ahuja, “Real-time omni-directional distance measurement with active panoramic vision,” *Int'l Jour. of Control, Automation, and Systems,* vol. 5, no. 2, pp. 184-191, 2007.

[5] J. Shin, S. Yi, Y. Hong and J. Suh, “Omnidirectional distance measurement based on active structured light image,” *Jour. of Institute of Control, Robotics and Systems*, vol. 16, no. 8, pp. 751-755, 2010.

[6] J. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25-30, 1965.

[7] http://en.wikipedia.org/wiki/Bresenham's_line_algorithm

[8] G. Kweon, S. Hwangbo, G. Kim, S. Yang and Y. Lee, “Wide-angle catadioptric lens with a rectilinear projection scheme,” *Applied optics,* vol. 45, no. 34, pp. 8659-8673, 2006.

신 진

2010년 2월 서울산업대학교 전기공학과 졸업(공학사). 2010년 3월~현재 서울과학기술대학교 산업대학원 석사과정 재학중.

이 수 영

1988년 2월 연세대학교 전자공학과 졸업(공학사). 1990년 2월 KAIST 전기및전자공학과 졸업(공학석사). 1994년 8월 KAIST 전기및전자공학과 졸업(공학박사). 1995년 3월~1999년 8월 KIST 시스템연구부 선임연구원. 1997년 2월~1998년 2월 Univ. of Southern California 박사후과정. 1999년 9월~2007년 2월 전북대학교 전자정보공학부 부교수. 2005년 6월~2006년 8월 Univ. of Illinois at Urbana-Champaign 방문교수. 2007년 3월~현재 서울과학기술대학교 전기공학과 교수. 관심분야는 보행로봇, 로봇비전, 로봇센서.