

논문 2010-47SC-3-2

천장의 반복 무늬를 이용한 로봇의 방향 및 위치 계산

(Computation of Robot Orientation and Localization using Repeating Feature of Ceiling Textures)

진 홍 신*, 양 창 주*, 이 왕 희**, 김 형 석**

(Hongxin Chen, Changju Yang, Wanghee Lee, and Hyongsuk Kim)

요 약

천장의 반복 무늬를 이용하여 로봇의 방향과 위치를 계산하는 새로운 알고리즘을 제안하였다. 방향 계산을 위해서 천장 영상으로부터 검출한 직선과 로봇의 현재 방향 간의 상대 각도를 계산한 후, 천장무늬에 포함된 여러 기준 직선의 방향들에 상기 상대각도를 더하여 로봇의 절대방향 후보 집합을 구성한다. 로봇 방향은 이 절대방향 후보 집합 중, 이전의 로봇 방향과 가장 가까운 방향으로 결정한다. 로봇의 위치도, 영상에서의 격자점의 화소 위치를 이용하여 로봇의 격자점과 로봇 간의 상대 위치를 계산 한 후, 이전 로봇 주변의 격자점에 상기 상대위치 벡터를 더하여 절대 위치 후보 집합을 구성하며, 이들 중 이전 로봇의 위치와 가장 가까운 위치를 현재 로봇 위치로 결정한다. 제안한 알고리즘을 이용하여 구한 위치 및 방향의 정확성을 검증하기 위한 실험을 수행하였다.

Abstract

A novel algorithm which computes the orientations and the locations of mobile robots with repeating feature of ceiling textures is proposed. For orientation computation, the relative angle between a straight line on a ceiling image and the center line of the robot is firstly computed. The relative angle is added to the global reference angles of straight lines of the ceiling and a group of robot orientation candidates is composed. The robot orientation is determined from such orientation candidate group by choosing the closest one to the previous robot orientation. Similarly, the location of the robot is determined from the robot location candidate group by choosing the closest one to the previous robot location, where robot location candidate group is composed by adding the relative location of the robot to global locations of neighboring nodes of the previous robot locations. The experiment has been done to test the accuracy of the algorithm in computing locations and orientations.

Keywords : 로봇위치, 방향, 천장무늬, 교차점, 직선

I. 서 론

이동로봇의 위치계산을 위해서 과거에는 로봇의 초기 위치를 기준으로 계산하는 dead reckoning 방법이 많이 사용되었다^[1]. 이 방법은 간단하지만, 바퀴가 미끄러짐으로 인해서 생기는 에러가 누적된다는 문제가 있었다. 이 문제점을 극복하기 위해서, 초음파 센서도 많이 사용되었다. 이 방법은 초음파가 물체까지의 비행시

간 (time-of-flight (TOF))을 계산하여 거리를 측정하는 방법인데, 정확성이 크지 않으며 환경의 영향을 많이 받는다^[2-3].

최근에는 랜드마크를 사용하는 방법이 많이 연구되고 있다. 이에 대한 방법 중의 하나는 RFID (Radio기술을 이용하는 것인데, RFID tags를 작업환경에 다수 배치한 후, RFID reader를 장착한 로봇이 움직이면서 RFID tag 위치를 알아내는 방법에 의해 로봇의 위치를 계산하는 방법이다. 여기서 각 RFID tag 는 각각 자신의 위치를 가지고 있기 때문에 로봇의 위치를 알아내는데 활용할 수 있다^[4-6]. 또, 다른 방법은 시각적인 랜드마크를 사용하는 방법이다. 비전센서는 랜드마크의 특

* 학생회원, ** 정회원, 전북대학교 전자정보공학부
(Division of Electronics and Information
Engineering, Chonbuk National University)
접수일자: 2009년9월17일, 수정완료일: 2010년5월13일

징을 인식하여 로봇의 위치를 계산하는 것이다. 이를 위해서 특수한 모양으로 제작된 랜드마크를 제작해서 사용한다. 랜드마크를 이용하는 방법 중 효과적인 방법은 랜드마크를 천장에 부착하는 방법이다. 랜드마크가 약속된 코드들 표현하게 하고, 작은 크기의 랜드마크에 많은 코드 정보를 포함시키기 위해서 개발된 칼라 기반 천장 부착 랜드마크 방법이다^[7~8]. 이 방법에서는 각 랜드마크가 고유한 상대적인 위치를 표현하게 하여 로봇은 랜드마크의 상대적 위치를 누적시키는 방법으로 위치를 계산하는 것이다. 이 방법은 사용되지 않는 천장에 랜드마크를 부착하여 공간의 활용도를 높인다는 장점은 있지만, 천장에 랜드마크를 부착해야 하는 노력과 미관을 해친다는 문제점이 있다.

본 연구는 천장에 랜드마크를 부착하지 않고, 천장의 반복 무늬만을 활용해서 로봇의 방향과 위치를 알아내는 새롭고 효과적인 방법이다. 격자형 반복 무늬를 사용하면, 직선이나 교차점들을 활용해서 위치나 방향 정보에 누적되는 에러를 줄일 수 있으며, 직선이나 교차점등이 검출되지 않는 영상 프레임에서는 화소 간의 이동 벡터인 Optical Flow 벡터들을 영상 프레임마다 누적한 합 벡터를 활용함으로써 위치정보를 예측할 수 있다.

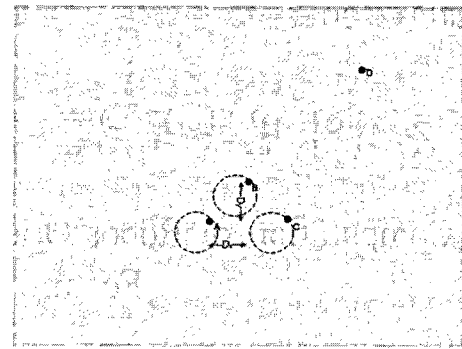
II. 반복격자형 천장 무늬에서의 방향 및 위치 성분

천장이나 벽지는 그림 1과 같이 일정한 무늬가 반복되는 경우가 많으며, 벽지의 경우에는 그림 1(가)처럼 복잡한 반복 무늬를 사용하는 경우가 많지만, 천장무늬의 경우에는 그림 1(나)처럼 격자형의 단순한 반복 무늬를 갖는 경우가 많다. 어느 경우나 반복되는 무늬 정보 내에 위치정보와 방향정보를 포함하고 있다.

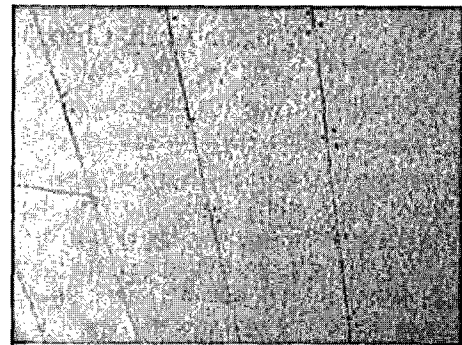
특히 그림 1과 같은 무늬가 천장 무늬라고 하면, 로봇에 설치된 카메라의 영상을 이용하여 로봇의 위치와 로봇의 방향을 계산할 수 있다. 그림 1(a)와 같은 무늬의 경우에 A의 위치 (A_x, A_y) 를 알고, 무늬 패턴의 반복되는 두 특징 점간의 x, y 축 거리 D_x 및 D_y 를 알고 있다고 하자. 그러면, B의 위치 (B_x, B_y) 는

$$\begin{aligned} B_x &= A_x + D_x \\ B_y &= A_y + D_y \end{aligned} \quad (1)$$

로서 계산할 수 있다.



(a)



(b)

그림 1. 반복무늬를 갖는 벽지(a)와 천장 무늬 (b)의 예. 반복되는 무늬에는 위치정보와 방향정보를 포함하고 있다.

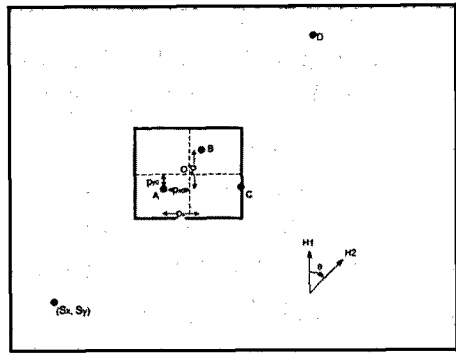
Fig. 1. Illustrations of the repeating texture of (a) wall paper (b) ceiling. Repeating textures contain the information of location and orientation.

또, C의 위치는,

$$\begin{aligned} C_x &= A_x + 2D_x \\ C_y &= A_y \end{aligned} \quad (2)$$

로서 점의 좌표를 계산할 수 있다. 이 천장무늬의 위치가 천장을 향해 수직으로 설치된 카메라 영상에 포착되면 화소간의 거리를 이용하여 로봇의 위치를 계산할 수 있다. 즉, 그림 2(a)의 네모 안의 영상을 로봇에 장착한 카메라가 포착한 영상이라고 하고, 두 점간의 영상에서의 화소 수를 P_x 및 P_y 라고 하자. 또, A 점으로부터 영상의 중심점까지의 화소 수를 P_{x_0} 및 P_{y_0} 라고 하면, 로봇의 현재 위치 R_x 및 R_y 는

$$\begin{aligned} R_x &= A_x + \frac{P_{x_0}}{P_x} D_x \\ R_y &= A_y + \frac{P_{y_0}}{P_y} D_y \end{aligned} \quad (3)$$



(a)



(b)

그림 2. 영상을 이용한 로봇 위치 및 방향 계산
(a) 천장 전체 영상
(b) 로봇 카메라에 나타나는 부분 영상

Fig. 2. Image-based computation of the location and the orientation. (a) global view of a ceiling (b) local area of the ceiling.

로 계산 할 수 있다.

또, 상기와 같은 반복무늬로부터는 로봇의 방향정보도 얻을 수 있다. 즉, 로봇 이동 중에 그림 2(a)와 같은 영상을 얻었다고 가정하자. 이 영상으로부터 그림 2(a)에서 H1 방향으로 진행하고 있던 로봇이 그림 2(b) 같은 영상을 얻었다면, H2 방향으로 만큼 회전해 있음을 알 수 있다.

이와 경우 같이, 반복되는 천장 무늬를 가진 곳에서는 무늬의 종류에 관계없이 위치정보와 방향정보를 얻을 수 있다. 다음 절에서는 사무실에서 많이 사용하는 격자형 반복무늬가 있는 환경에서의 로봇의 위치 및 방향을 계산하는 알고리즘을 제안하였다.

III. 격자형 천장 무늬를 이용한 로봇의 방향 및 위치 계산

1. 로봇의 방향 계산

이동 중인 로봇의 방향은 그림 3에서와 같이 영상에서 직선을 검출하여 이 직선과 로봇의 현재 방향 간의 각도를 계산 한 후, 미리 준비한 작업장의 원점 기준 직선들에 대한 개별 각도에 더하여 이들 중, 이 전 영상에서의 로봇 방향과 가장 가까운 방향을 현재의 로봇 각도로 결정한다.

격자무늬의 천장 영상에는 평행하거나 교차하는 2 방향의 직선이 존재한다. 그림 4(a)에서와 같이 실제

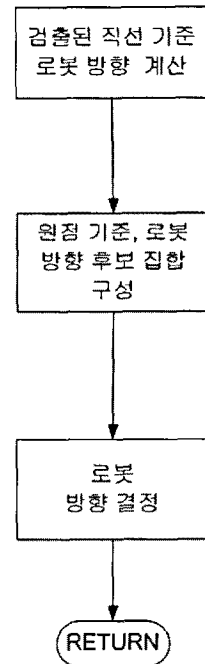


그림 3. 천장 무늬 영상에서의 직선을 이용하여 로봇의 방향 결정 흐름도

Fig. 3. Flow diagram of the orientation computation using straight lines on ceiling images.

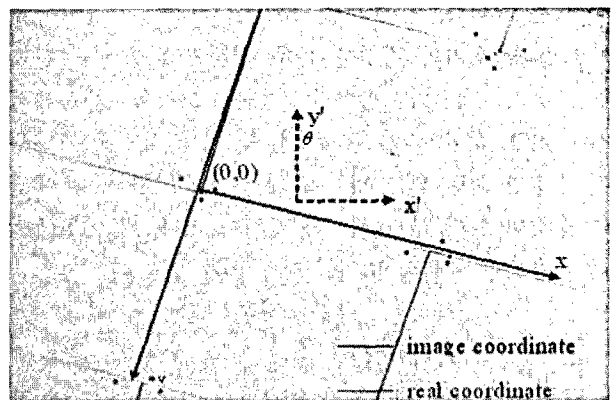


그림 4 격자형 천장 무늬에서 직선 및 교차점에 대한 원점기준 좌표 계 (x',y') 와 로봇좌표 계(x,y)

Fig. 4. World coordinate(x',y') and robot coordinate (x,y) of straight lines and crossing nodes.

공간의 원점을 기준으로 x 축에 평행한 직선의 방향을 $-\pi/2$ 이고, y 축에 평행한 직선이 방향을 0로 정의한다. 이렇게 하면, -x 축에 평행한 방향은 $\pi/2$ 이고, -y 축에 평행한 방향은 π 가 되는 것이다.

그림 4(b)와 같은 공간 좌표 계에서 각 직선의 방향은 $\{-\pi/2, 0, \pi/2, \pi\}$ 와 같은 4 개의 방향을 갖게 된다. 또, 영상 좌표계에서는 로봇과 카메라 시스템의 배치 상, 영상의 위 쪽 방향을 로봇의 전면 방향으로서, y'로 정의하고, y 축과의 사이 각을 θ 라 하자. 한 개의 직선을 구해서 그 직선 로봇의 앞면과의 각도인 θ 를 구했다고 하면 실제의 로봇의 방향은 $\{\theta - \pi/2, \theta, \theta + \pi/2, \theta + \pi\}$ 중의 하나가 된다.

즉, 영상의 상하를 연결하는 수직선과 가장 가까운 직선을 우선 구하고, 이 수직선과 직선이 이루는 각도를 θ 라고 한다. 그런데, 영상에서 구한 직선은 원점을 기준으로 0도 방향, $\pi/2$, π 혹은 $-\pi/2$ 중 어느 직선과 θ 인지를 구별해야 한다.

만약 로봇의 회전 속도가 충분히 낮다고 가정하면, 즉, 로봇의 방향 변화가 매우 작다면, 로봇의 현재 방향은 가장 최근에 포착한 영상의 방향과 유사한 방향이 될 것이다. 이 방향을 a_{i-1} 로 한다면, 현재의 방향은 $\{\theta - \pi/2, \theta, \theta + \pi/2, \theta + \pi\}$ 중 a_{i-1} 에 가까운 값이다.

이를 위해서, a_i 를 현재의 방향이라고 하면,

$$a_i = \min|\text{mod}((\beta - a_{i-1}) + 3\pi, 2\pi) - \pi|, \quad (4)$$

$$\beta \in \{\theta - \pi/2, \theta, \theta + \pi/2, \theta + \pi\}$$

이 되는 것이다.

그런데, 초기의 방향을 매우 정확하게 계산할 필요는 없고, 대충의 방향만 알면, 그것을 이용해서 $\{\theta - \pi/2, \theta, \theta + \pi/2, \theta + \pi\}$ 중 어느 각도에 가까운지 결정만 하면 되는 것이기 때문이다. 따라서 이 에러는 누적되지 않는다는 큰 장점이 있다. 다음 절에서는 허용되는 에러의 범위를 계산한다.

2. 로봇의 위치 계산

로봇의 위치를 계산하기 위해서는 그림 5와 같이 작업장의 교차점들의 위치 집합 중, 로봇의 이전 위치 점을 중심으로 가능한 주변의 교차점들의 위치 좌표들을 취한다. 또, 현재의 영상에서 가장 뚜렷한 교차점을 선

택하여 영상 중심의 위치를 상기 교차점에 대한 상대 벡터로 표현하고 이를 기준 원점의 좌표로 회전 변환한 후, 이전 영상에서의 로봇 위치와 가장 가까운 위치를 실제 로봇 위치로 결정한다.

그림 4에서 보는 바와 같이 대상으로 하는 천장 무늬에는 두 개의 직선이 교차하여 만드는 많은 교차점들이 있다. 천장무늬의 두 인접 교차점 간 거리를 d라고 하자. 따라서 이 교차점의 위치들은 알 수 있고, 좌표축이 정의된다면, 교차점들의 위치를 계산할 수 있다. 만약, 이 교차점으로부터 영상의 중앙까지의 거리를 계산하면 로봇의 현재의 위치도 계산할 수 있다.

이전 프레임의 영상에서 인식된 교차점의 실제 위치를 $(X_{i-1}, Y_{i-1})^T$ 라고 할 때, 현재 영상 프레임에서의 교차점의 실제 위치는 다음과 같은 점 중의 하나이다.

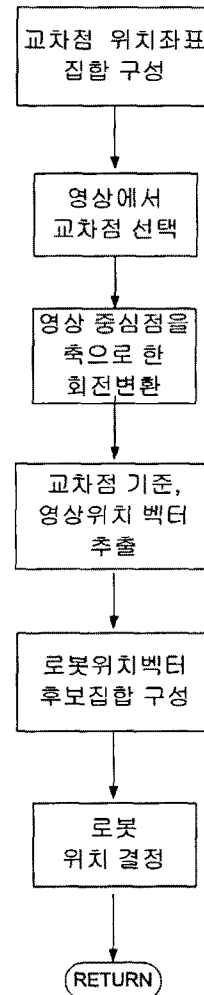


그림 5. 영상에서 직선들의 교차점을 이용한 로봇의 위치 계산 루틴

Fig. 5. Flow diagram of the location computation using crossing nodes on ceiling images.

$$\{(X_i, Y_i)^T | (X_i, Y_i)^T = (X_{i-1} + k_1 d_1, Y_{i-1} + k_2 d_2)^T, k_1, k_2 \in \{-1, 0, 1\}\} \quad (5)$$

여기서 d_1 과 d_2 는 x 축과 y축 간의 거리이다.

상기와 같은 방법을 사용하여 교차점의 위치를 계산한 후, 로봇의 위치는 다음과 같이 구할 수 있다. 즉, 로봇 위치는 영상의 중심점의 위치와 같으므로 이를 C 라하고, 검출된 교차점의 위치 점을 A, B를 인접위치의 교차점이라고 하면, 벡터 \vec{CF} 의 방향은 로봇의 방향과 같다.

현재의 로봇 방향은 a_i 로 계산되어 있으므로, 각 $\angle BAC = \varphi$ 를 계산할 수 있다. 만약, 영상 평면과 천장평면의 크기비율 관계를 이용하면, 실제 위치점 A로부터 $C = (x_{pi}, y_{pi})^T$ 점의 위치는 (6)과 같이 계산할 수 있다.

$$(x_\pi, y_\pi)^T = (X_i, Y_i)^T + s \cdot (\cos(a_i - \varphi), \sin(a_i - \varphi))^T \quad (6)$$

점 $(X_i, Y_i)^T$ 은 9개의 후보 점을 가질 수 있으므로, $(x_{pi}, y_{pi})^T$ a점 역시 9 점을 갖는다. 이와 같은 변환의 절차는 그림 6의 실제거리 상대위치 벡터로 변환방법에서 을 기술하고 있다.

그런데, 로봇의 속도는 빠르지 않으므로, $(x_{pi}, y_{pi})^T$ 점은 $(x_{i-1}, y_{i-1})^T$ 점에 가까이 위치하게 된다. 따라서 로봇의 현재 위치 점은

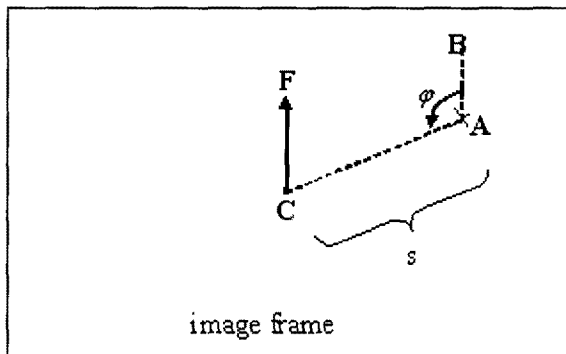


그림 6. 로봇의 방향 및 노드로봇 중심 간 벡터를 이용한 로봇의 절대 위치 계산

Fig. 6. Computation of robot location using the orientation of robot and the vector between a node on an image and robot center.

$$\begin{aligned} (x_i, y_i)^T &= \min |(x_{pi}, y_{pi})^T - (x_{i-1}, y_{i-1})^T|, \\ (x_{pi}, y_{pi})^T &= (X_i, Y_i)^T \\ &\quad + s \cdot (\cos(a_i - \varphi), \sin(a_i - \varphi))^T \\ &= (X_{i-1} + k_1 d_1, Y_{i-1} + k_2 d_2)^T \\ &\quad + s \cdot (\cos(a_i - \varphi), \sin(a_i - \varphi))^T \\ &\quad, k_1, k_2 \in \{-1, 0, 1\} \end{aligned} \quad (7)$$

로 계산되고,

$$(x_i, y_i)^T = \min_{k_1, k_2 \in \{-1, 0, 1\}} \{ |(X_{i-1} + k_1 d_1, Y_{i-1} + k_2 d_2)^T + s \cdot (\cos(a_i - \varphi), \sin(a_i - \varphi))^T - (x_{i-1}, y_{i-1})^T | \} \quad (8)$$

로서 로봇의 위치점을 구할 수 있다. k_1 와 k_2 는 $\{-1, 0, 1\}$ 중에서 (8) 식이 최소 값을 갖는 값이다. 식 (8)에 의하면, 현재의 화면상에 나타나는 교차점 $(X_i, Y_i)^T$ 의 위치도 구할 수 있다.

결론적으로 알고 있는 교차점 $(X_{i-1}, Y_{i-1})^T$ 과 이전 프레임에서의 로봇 위치점 $(x_{i-1}, y_{i-1})^T$ 를 알면, 로봇의 현재 위치와 현 영상에 나타나는 교차점의 위치는 식 (8) 및 식 (5)를 이용하면 계산할 수 있다. 따라서, 로봇의 초기 위치가 주어지면, 식 (5) 에 의해, 로봇의 위치를 점차적으로 계산할 수 있는 것이며, 식 (8)를 이용하므로, $(x_i, y_i)^T$ 점에 대한 에러가 누적되지 않는다는 중요한 특징이 있다.

IV. 실험 및 검토

제안한 알고리즘을 이용해서 실제 사무실 천장의 로봇 방향 및 위치 검출성능을 확인하였다. 사무실 천장 무늬는 그림 1(b)와 같은 모양의 일반적인 사무실의 천장무늬로서, 긴 직 사각형들이 엇갈리게 배치되어 있는 형태이다. 직사각형의 각 변에 해당하는 곳에서는 직선이 검출되고, 직사각형이 이어지는 위치에서 격자점을 구성한다.

카메라는 천장을 향하여 설치하였고, 정확히 수직이 될 수 있도록 본문에서 기술한 방법에 의해 캘리브레이션 하였다. 실험 환경에서의 바닥에서부터의 천장 높이는 270cm 이었고, 카메라 높이는 50cm이었다. 방향 및 위치 검출의 정확성 실험을 위해서 본 연구에서 고안한 방법은 실험장 바닥의 모눈종이 위에 일정한 도형을 그

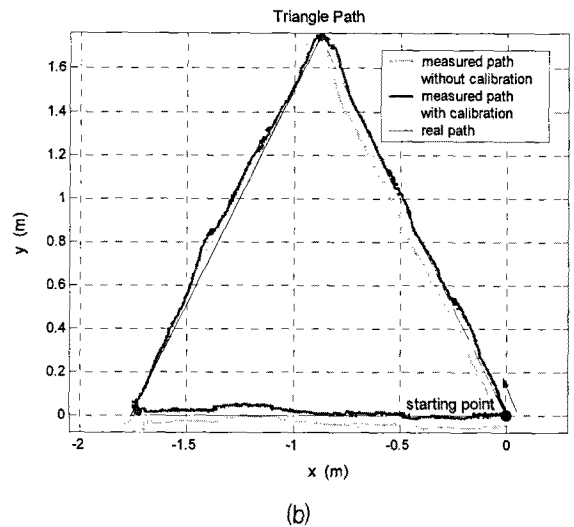
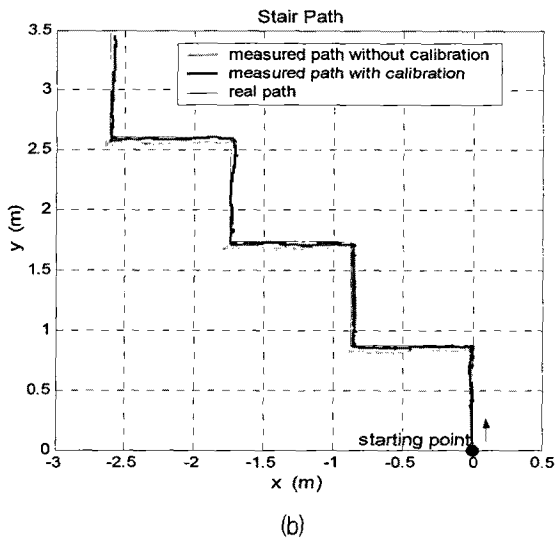
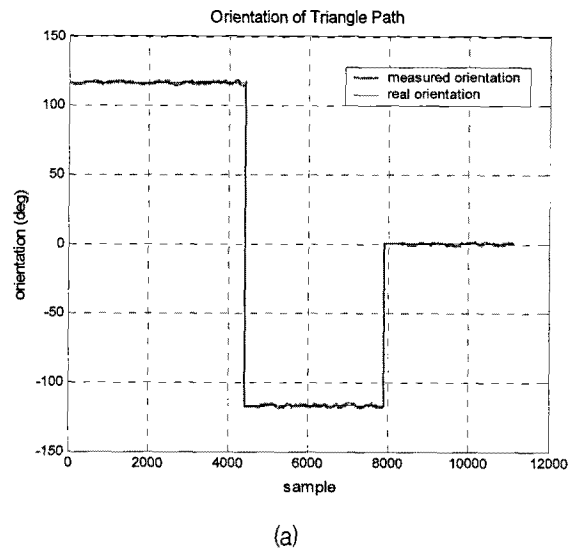
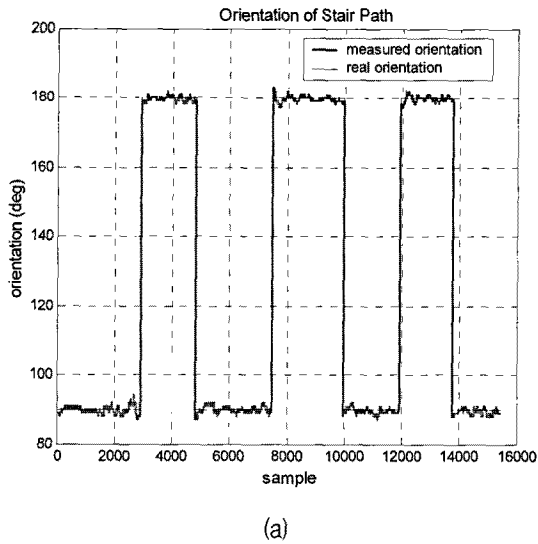


그림 7. 로봇을 계단형 경로의 우하 위치에서 출발하여 좌 상 위치로 이동시키는 동안 측정된 로봇 방향 값(a)와 로봇 위치 값 (b)의 비교

Fig. 7. Comparisons of robot orientations (a) and locations (b) between real path and measured paths while the robot is moved along a stair path starting from a point of bottom right.

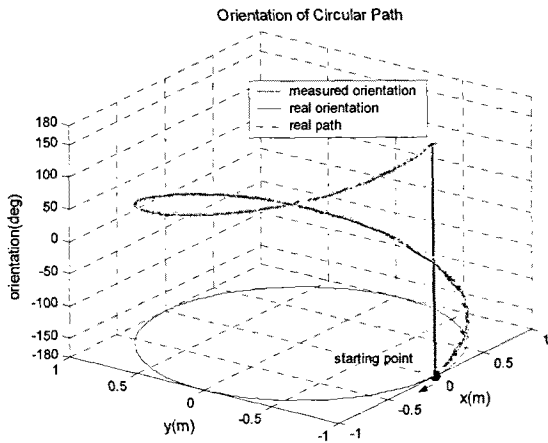
그림 8. 로봇을 삼각형 경로의 우하 위치에서 출발하여 한 바퀴 이동시키는 동안 측정된 로봇 방향 값 (a)와 로봇 위치 값 (b)의 비교

Fig. 8. Comparisons of robot orientations (a) and locations (b) between real path and measured paths while the robot is moved along a triangular path starting from a point of bottom-right.

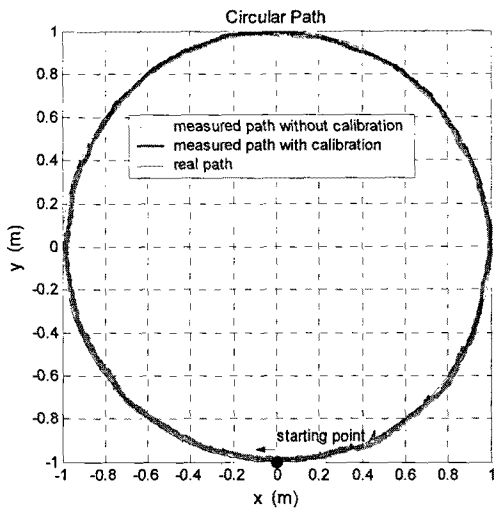
려놓은 후, 로봇을 수동으로 그 도형을 따라 이동시키는 동안, 카메라가 천장 무늬만 활용하여 방향 및 위치를 계산하여 제시하도록 하였다. 제안한 방법을 사용하면, 로봇이 도형 위의 어디에 위치하던지 상관없이 로봇의 진행 방향과 위치를 기준 정보로 알 수 있으며, 동시 제안한 알고리즘에 의한 측정값을 얻어 비교할 수 있다.

첫 번째 실험은 바닥에 그려놓은 그림 7(b)와 같은 계단형 길을 따라 우하 위치에서부터 좌 상 위치로 이

동하는 동안 측정된 로봇의 방향 (그림 7(a))과 로봇의 위치 측정 값(그림 7(b))정보이다. 이를 위해서 로봇의 위치는 90도 방향으로 진행하다가 180도 방향으로 전환하는 과정을 반복해야 한다. 그림 7(a)는 이에 대한 측정 실험으로서 점선은 실제 로봇 방향이고, 실선은 측정된 로봇 방향이다. 실험결과에서 확인할 수 있는 바와 같이 측정예러는 2도 이하로서 매우 정확한 측정이 가능함을 할 수 있다. 위치에 대한 정확도 측정실험도



(a)



(b)

그림 9. 로봇을 원형 경로의 아래 위치에서 출발하여 한 바퀴 이동시키는 동안 측정된 로봇 방향 값 (a)와 로봇 위치 값 (b)의 비교

Fig. 9. Comparisons of robot orientations (a) and locations (b) between real path and measured paths while the robot is moved along a circular path starting from a point at bottom.

수행하였는데, 그림 7(b)에서 점선은 실제 기준 위치이고, 점선은 측정된 위치 값이다. 점선들 중, 긴 점선은 캘리브레이션 전의 위치 값이고, 작은 점선은 캘리브레이션 후의 위치 값이다. 로봇을 캘리브레이션을 한 후 사용한다고 가정할 때, 로봇의 위치 에러는 2cm이하로서 매우 작은 값을 알 수 있다.

두 번째 실험은 그림 8(b)와 같은 삼각형의 길을 따라서 삼각형 우하 꼭지점에서 출발하여 반 시계방향으로 한 바퀴 도는 동안 측정된 방향 값 (그림 8(a))과 위

치 값 (그림 8(b)의 기준 값과의 비교 그래프이다. 그림 (그림 8(a)에서 로봇은 +120의 방향으로 진행하다가 -120로 전환한 후, 0도 방향으로 전환하고 있어서, 예측한 값이 정확히 일치함을 알 수 있으며, 그림 8(b)에서의 캘리브레이션 후의 측정된 위치 정보도 거의 정삼각형을 보여주고 있어서 기준 값과의 차이가 매우 적음을 알 수 있다.

그림 9는 원형의 길을 좌측 중간 위치에서 출발하여 시계방향을 따라서 로봇이 움직이면서 측정된 방향과 위치를 실험 결과이다. 그림 9(a)의 그래프에서 상측 끝이 로봇의 출발 위치이다. 그림 9(a)에서 보는 바와 같이 로봇의 방향과 위치가 정확히 측정되었음을 알 수 있다.

V. 결 론

본 연구는 격자무늬가 있는 천장의 무늬만으로도 로봇의 위치 및 방향을 인식하는 기술로서 천장에 있는 직선과 격자점 인식을 통해서 로봇의 방향과 위치를 계산할 수 있게 하는 알고리즘이다. 이 알고리즘은 이전에 계산된 방향 및 위치정보를 기준으로 한 예측정보를 활용하며, 현재의 영상을 통하여 얻은 방향 및 위치정보를 활용하여 정보를 교정하기 때문에 에러가 누적되지 않는다는 특징이 있다.

본 알고리즘에 의한 방향 및 위치 계산의 정확성을 시험하기 위해서, 실험장 바닥의 모눈종이 위에 그려진 일정한 도형을 따라 로봇을 수동으로 이동시키는 동안, 카메라가 천장 무늬만 활용하여 방향 및 위치를 계산하여 바닥에 그려진 실제 도형에서의 방향 및 위치와 비교하도록 하였다. 계산의 정확성을 위해서 카메라가 똑바로 천장을 향하도록 사전에 캘리브레이션을 수행하였다.

실험에 사용한 도형은 계단식도형, 삼각형 도형, 원형도형을 대상으로 하였으며, 캘리브레이션을 거친 실험 결과는 방향에러 2도 이하, 위치에러 2 cm이하를 유지하였다.

본 알고리즘을 이용한 방향 및 위치 정보를 활용하여 사무실용 커피 배달용 로봇의 제어에 활용한 결과, 로봇은 정확한 길을 따라 항상 목적지에 정확히 도달했으며, 심부름 임무를 정확히 완수하였다. 실험에서의 커피 배달 응용은 단순작업에 불과하지만, 본 연구에서 개발한 로봇의 방향 및 위치 계산의 정확성을 확인하는 데

는 충분한 방법이다.

특히, 본 연구는 천장의 무늬를 그대로 이용하기 때문에 기존의 제품들에서 필수적인 랜드마크나 엔코더 등의 부가장치가 없이 정확한 위치를 측정할 수 있으며, 대부분의 사무실은 격자무늬를 가지고 있으므로, 사무실용 로봇에 유용하게 활용이 가능하다고 할 수 있다.

참 고 문 헌

- [1] B.Barshan and H.F.Durrant-Whyte, "Inertial navigation systems for mobile robots," IEEE trans. Robot Automat, vol. 11, pp. 328-342, June 1995.
- [2] J. J. Leonard and H. F. Durrant-Whyte, "Directed Sonar Sensing for Mobile Robot Navigation," Kluwer Academic Publisher, 1992.
- [3] B. Triggs, "Model-based sonar localization for mobile robots," Robotics and Autonomous Systems, vol. 12, pp. 173-184, 1994.
- [4] Yu Zhou, Wenfei Liu, "Preliminary Research on Indoor Mobile Robot Localization using Laser-activated RFID," 2007 IEEE International Conference on RFID, pp. 78-85, 2007.
- [5] Jing Liu, Yang Po, "A Localization Algorithm for Mobile Robots in RFID System," WiCom 2007 International Conference on Wireless Communications, Networking and Mobile Computing, pp. 2109 - 2112, 2007.
- [6] P. Bahl and V. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," IEEE INFOCOM' 02, VOL.2, pp. 775-784, Mar. 2000.
- [7] Sooyong Lee, Jae-Bok Song, "Mobile robot localization using infrared lightreflecting landmarks," ICCAS '07 International Conference on Control, Automation and Systems, pp. 674-677, 2007.
- [8] 진홍신, 왕실, 양창주, 이준호, 김형석, "천정 부착 셀코드 랜드마크에 기반한 이동 로봇의 정밀 위치 계산," 전자공학회논문지, Vol.46-CI, No. 2, pp.75-83, 3월, 2009.

저 자 소 개



진 홍 신(학생회원)
2004년 Wuhan University of Technology, China, 학사 졸업
2007년 Wuhan University of Technology, China, 석사 졸업

2007년~현재 전북대학교 일반대학원 전자정보공학부 박사과정
<주관심분야: 멀티미디어와 통신 시스템, 로봇비전, 원격제어>



이 왕 희(정회원)
2001년 전북대학교 제어계측공학과 석사 졸업
2001년~2004년 LG 산전 중앙연구소 연구원
2005년~현재 전북대학교 산학협력단

<주관심분야: 로봇비전, 로봇센서시스템>



양 창 주(학생회원)
2008년 전북대학교 전자정보공학부 학사 졸업
2010년 전북대학교 일반대학원 전자정보공학부 석사과정
2010년~현재 전북대학교 일반대학원 박사과정

<주관심분야: 로봇비전, 아날로그 회로설계>



김 형 석(정회원)
1992년 University of Missouri, Columbia, 박사 졸업
2003년~현재 전북대학교 전자정보공학부 교수
<주관심분야: 로봇비전, 로봇센서시스템, 아날로그 병렬처리 회로>