한국항해항만학회지 제38권 제5호
J. Navig. Port Res. Vol. 38, No. 5 : 457-462, October 2014 (ISSN:1598-5725(Print)/ISSN:2093-8470(Online))
DOI : http://dx.doi.org/10.5394/KINPR.2014.38.5.457

이동노드의 실내위치파악 시스템을 통한 벡터기반 상대방위각 알고리즘

† 손주영

† 한국해양대학교 IT공학부 교수

A Vector-based Azimuth Algorithm using Indoor-Positioning Systems for Mobile Nodes

† *Joo-Young Son*

† *Div. of IT Engineering, Korea Maritime and Ocean University, Busan, S. Korea*

요 약 : 실내에 있는 노드의 위치를 알려주는 시스템은 여러 유용한 응용에 활용된다. 그 가운데 가장 대중적인 응용이 내비게이션 시스템이다. 여기서는 노드가 움직이는 방향에 대한 정보를 필요로 한다. 특히 위치 이동에 따른 변화량과 방향에 대한 정보가 실시간으로 제공되어야 한다. 이 논문에서는 방위각 센서가 작동하지 않는 실내에서 기존의 위치를 파악할 수 있는 시스템을 이용하여 움직이는 노드의 이동 방위각의 변화량과 변화방향을 정확하게 파악하는 데 효과적인 벡터기반 알고리즘을 제시한다. 기존 알고리즘은 여러 기하학적 계산 단계들을 통해 이동방향의 변화량을 파악한다. 이 논문에서 제안하는 알고리즘은 벡터를 기반으로 하는 단순한 산술식을 통해 이동 노드의 진행방향의 방위각 변화량을 구하고, 노드가 직전에 이동한 방향에 근거하여 도출된 단순한 수식의 부호값(음 또는 양)에 따라 변화방향을 파악한다. 지속적으로 이동하는 노드의 변화하는 방위각에 대한 파악이 기존 알고리즘에 비해 신속하고 정확한 결과를 얻을 수 있음을 논리식과 수식으로 증명하였다.

핵심용어 : 방위각 변화량, 이동방향, 벡터 알고리즘, 실내위치파악 시스템, 내비게이션 시스템

Abstract : *Indoor-positioning systems are useful to various applications. Navigation system is one of the most popular applications, which needs the information of directions of nodes' movements. Specifically the applications should get the information in real-time to properly show the current moving position of a node. In this paper, simple vector-based algorithms are proposed to compute amount and direction of changes of azimuth of mobile nodes' heading directions using existing indoor positioning systems in indoor environments where azimuth sensors do not work properly. Previous algorithms calculate the azimuth changes by too many steps of topology-based formula. The algorithms proposed in this paper get the amount of changes of azimuth by simple formula based on vector, and determine the direction of changes by the sign of value of simple formula based on the previous movement of nodes. The algorithms are much simpler and less error-prone than previous ones, and then they can detect changes in many location-based applications as well. The performance of the algorithms is proved logically and mathematically.*

Key words : *Azimuth, Heading Directions, Vector Algorithm, Indoor Positioning Systems, Navigation Systems*

1. 서 론

근래에 지리적 위치를 기반으로 제공되는 서비스(LBS - location-based services)들이 다양하게 등장하였고, 실제 손에 들고 다니는 장비(hand-held devices)가 보편화되면서 널리 이용되고 있다. 주로 건물 밖 열린 공간에서 GPS에 의해 위치를 파악하고 (LDT - Location Detection Technology), 그 위치의 지리적 문맥(geographical context)과 개인정보를 결합한 개인 맞춤형 서비스(personalized services)를 제공한다.(Woo, 2004) 이 가운데 가장 대중적인 서비스가 내비게이션 서비스라 할 수 있다. 이 서비스는 현재 주로 건물 밖 도로를 중심으로 제공되고 있으나 오래전부터 건물 내 서비스도 시도되고 있다.(Hopper, 1993)

현재 위치에서 목적지까지 최적경로를 찾아 안내하는 서비스이므로 현재 이동하는 노드(사람, 차량 등)가 전진하는 방향을 정확하고 빠르게 파악하는 것이 매우 중요하다. 특히 방향

† Corresponding author : 정회원, mmlab@kmou.ac.kr 051)410-4575

을 감지하는 방위각 센서가 작동하지 않거나 여러 요인으로 작동이 부정확할 가능성이 매우 높은 실내 환경에서 방위각 변화가 발생할 때 이를 신속 정확하게 파악하는 것은 매우 중요하다. 이 논문에서는 위치파악 시스템을 통해 구한 위치 정보를 이용하여 방위각을 파악할 수 없는 실내 환경에서 이동하는 노드의 이동방향을 정확하고 신속하게 계산하는 알고리즘을 제안한다. 기존 알고리즘은 계산과정이 복잡하여 계산결과가 즉각적으로 나와 변화에 빠르게 반응해야 하는 응용에 적용하는 것이 적합하지 않다.(Kim, 2013)

2. 기존 기술 연구

2.1 위치파악기술

실내 내비게이션 서비스는 우선 이동노드의 정확한 위치를 파악하는 기술(positioning techniques)이 우선이다. 위치파악 기술은 네 가지로 구별된다.(Liu, 2007) 첫째는 원격 위치파악 기술이다. 신호 전송자가 이동노드에 있고, 고정된 위치에 있는 여럿 측정단위들이 신호를 읽어 한 곳(master station)에 보낸다. 여기서 전송자의 위치가 계산된다. 둘째는 신호측정단위가 이동노드에 있는 자가 위치파악 (self-positioning)을 하는 기술이다. 이동노드는 위치가 알려져 있는 몇몇 신호 전송자로부터 신호를 수신하고 이를 기반으로 자신의 위치를 계산한다. 셋째는 위치가 알려진 신호를 이동노드가 수신하여 측정된 신호 값들을 원격에 있는 다른 노드로 보내져 거기서 위치를 계산하는 경우 간접 원격 위치파악(indirect remote positioning) 기술이라 한다. 넷째로 원격 위치파악 노드로부터 이동노드로 측정결과가 전달되는 경우를 간접 자가 위치파악(indirect self-positioning) 기술이라 한다.

위치파악기술에서 무선통신매체는 블루투스, Wi-Fi, 3G, LTE, UWB, 초음파, 또는 적외선 등이 쓰인다. 그러나 실내에서는 다중경로 문제와 가시선 경로 확보 어려움, 그리고 다양한 실내구조와 반사면 등의 건물마다 독특함이 있어 전파 특성을 모델링하기가 매우 어렵다. 신호를 좀 더 정확하게 측정하기 위해 UHF를 활용하는 새로운 솔루션을 제시하거나 RF와 초음파, 그리고 RF와 적외선 등 여러 매체를 복합적으로 활용하기도 한다.

2.2 위치파악 알고리즘

위치파악기술에서 가장 중요한 기술요소는 측정된 신호값을 이용하여 위치를 실질적으로 계산하는 위치파악 알고리즘(positioning algorithm)이다.

삼변측량법(trilateration method)이 대표적이다.(Liu 2007) 도착시간(TOA - time of arrival), 도착시간차(TDOA - time difference of arrival) 또는 왕복비행시간(RTOF - round time of flight)을 측정한 후 신호의 세기변화 또는 전파 시간과 신호전달속도를 곱한 값으로 이동노드의 위치를 계산한다. 삼변측량법에는 구체적으로 두 가지 기법을 쓴다. 첫째, 거리기반 기술(lateration technique)이다. 여기서는 N개의 기지국에서 이동노드의 신호를 측정해서 그 결과로 식 (1)의 비용함수가 최소가 되는 (x, y, t)인 이동노드의 위치를 추정한다.

$$F(x) = \sum_{i=1}^{N} \alpha_i^2 f_i^2(x) \quad (1)$$

여기서 α_i는 수신신호의 신뢰도를 반영하는 계수이고, $f_i(x)$(식 (2))는 각 신호측정단위에 의해 계산되는 거리함수이다.

$$f_i(x) = c(t_i - t) - \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (2)$$

따라서 이동노드의 위치는 $F(x)$가 최소가 되는 (x, y) 값으로 추정할 수 있다.

둘째, 각도기반기술(angulation technique)은 여러 쌍의 각도를 가진 직선들의 교차점으로 이동노드의 위치를 추정한다. 적어도 위치가 알려진 두 기준점과 측정된 두 개의 각도를 이용하여 2-D 상의 위치를 추정한다. 여기서는 방향성이 있는 안테나 또는 안테나 배열이 있어야 추정이 가능하다.

2.3 방위각 알고리즘

실제 이동노드가 이동하는 방향에 대한 정보는 길안내 서비스에 매우 중요하다. 방위각 센서의 작동이 불가하거나 불량한 실내 환경에서 이 정보를 획득하기 위해서는 노드가 이동하는 방향의 각 변화를 파악하면 된다. Kim(2013)에서는 이동노드가 임의의 반경을 가지고 원호를 그리며 이동한다고 가정한다. 따라서 여기서 제안하는 알고리즘에서는 이동노드의 궤적인 원호의 중심점을 찾아 그 중심점을 중심으로 이동시작지점에서 이동종료시점으로 회전한 각도를 계산한다. 회전각 추정을 위해 다음과 같은 계산과정을 거친다.

① 양끝 두 지점의 위치정보를 이용하여 두 지점을 잇는 선분의 중간지점을 계산한다.

② 양끝 두 지점을 잇는 선분의 중간지점을 지나는 법선을 구한다.

③ 파악된 위치정보 가운데 법선 위의 점 또는 그와 가장 인접한 점을 찾아 중앙점으로 삼는다.

④ 이동시작지점과 중앙점을 잇는 선분의 중간지점을 지나는 법선을 구한다.

⑤ 이동종료지점과 중앙점을 잇는 선분의 중간지점을 지나는 법선을 구한다.

⑥ ④,⑤단계에서 구한 두 법선의 교차점을 구한다. 이 점이 노드가 이동한 궤적 원호의 중심이다.

⑦ ⑥단계에서 구한 중심점과 양끝 두 지점을 잇는 삼각형의 각 변의 길이를 구한다.

⑧ 코사인 제2공식에 ⑦단계에서 구한 각 변의 길이를 대입하여 양끝 지점과 중심점의 사이각을 구한다. 이 각이 최종적으로 구하고자 하는 이동노드의 각 변위 값이다.

이 논문에서 제안하는 간편한 방위각 변화 값 계산 알고리즘은 위와 같은 계산과정을 단축하여 시간 복잡도를 크게 줄일 수 있다.

3. 벡터기반 상대방위각 알고리즘

2장에서 언급한 바와 같이 방위각 센서가 불능이거나 불량하게 작동하는 실내 환경에서 이동하는 노드의 전진하는 방향(heading direction)의 방위각 변화를 파악하는 것은 매우 중요하다. 특히 길을 안내하는 내비게이션 서비스에서는 시시각각 변하는 이동노드의 이동방향을 즉각 파악하여 유의미한 길안내가 유지되도록 활용되어야 한다.

이 논문에서 제안하는 알고리즘은 방위각 변화량과 변화방향을 파악하는 알고리즘이다. 기본적으로 상대적인 방위각 변화량을 파악하는 데 있어서 Kim(2013)에서 제안하는 환경과 제반조건이 유사하다고 가정한다. 즉, 위치정보를 추측하기 위해 삼각측량법을 활용하고, 이를 위한 신호는 UWB를 활용하는 것도 동일한 것으로 가정한다. 입력된 원 데이터(raw data)를 Kim(2013)에서 제시한 필터에 의해 잡음특성이 감소되도록 가공된 데이터를 이용한다. 이렇게 수집된 위치정보에는 오류가 무시할 정도라고 가정한다.

그러나 이 논문에서 제안하는 알고리즘은 Kim(2013)에서 가정한 것과 같이 이동노드의 이동 궤적이 원호를 그리며 회전하는 것에 국한하지 않는다. 다시 말해 이동노드의 이동 패턴에 대하여는 어떠한 제약도 두지 않는다. 따라서 일반적인 이동노드의 이동 형태에 대해서도 적용이 가능하다. 그리고 이 논문에서는 변화량뿐만 아니라 임의로 변화하는 방향도 즉각 파악할 수 있는 알고리즘도 함께 제시한다.

3.1 상대방위각 변화량 알고리즘

이동노드는 임의 지점 $I=(X_i, Y_i)$에서 이동을 시작한다. 연이어 파악된 지점 $J=(X_j, Y_j)$을 지나고, 세 번째로 파악된 지점 $K=(X_k, Y_k)$로 이동하는 형태로 이동한다. 노드의 궤적은 두 개의 연속된 $\overrightarrow{IJ}$와 $\overrightarrow{JK}$ 벡터로 표현이 가능하다. (**Fig. 1**)

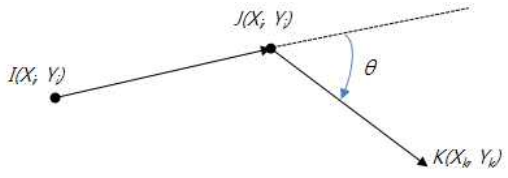

Fig. 1 A moving pattern of a mobile node

노드가 J 지점에서 K 지점으로 이동했을 때 변경된 전진방향의 방위각은 I->J->K 지점으로 이어지는 이동구간에서 변화된 위상각θ이다. **Fig. 1**의 예와 같이 이 각은 두 벡터의 사이각이고, 일반적인 벡터의 내적 공식인 식 (3)으로부터 유도된 식 (4)를 이용하여 구한다.

$$\vec{a} \cdot \vec{b} = |\vec{a}||\vec{b}|\cos\theta \tag{3}$$

$$\theta = \cos^{-1}\left(\frac{\vec{a} \cdot \vec{b}}{|\vec{a}||\vec{b}|}\right) \tag{4}$$

식 (4)에 의거하여 이동노드의 전진방향 방위각 변화량을 구체적으로 구하는 공식을 구하기 위해서는 우선 $\overrightarrow{IJ}$와 $\overrightarrow{JK}$ 벡터를 평행 이동하여 원점 O을 시점으로 하는 위치벡터 $\overrightarrow{J-I}$와 $\overrightarrow{K-J}$로 변환한다. (**Fig. 2**)

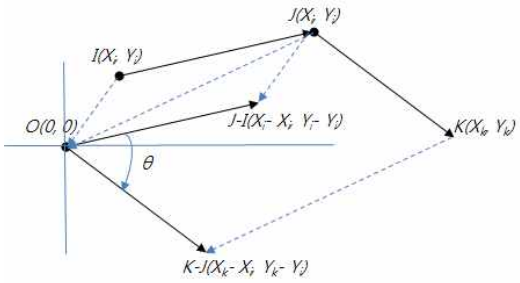

Fig. 2 Parallel transference of $\overrightarrow{IJ}$ and $\overrightarrow{JK}$ to $\overrightarrow{J-I}$ and $\overrightarrow{K-J}$

그러면 여기서 구하는 변화된 방위각 θ는 위치벡터 $\overrightarrow{J-I}$와 $\overrightarrow{K-J}$의 사이각이 된다. 이 사이각은 위의 식 (4)를 통해 아래와 같이 구할 수 있다.

$$\overrightarrow{J-I} \cdot \overrightarrow{K-J} = (X_j - X_i)\times(X_k - X_j) + (Y_j - Y_i)\times(Y_k - Y_j) \tag{5}$$

$$|\overrightarrow{J-I}| = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \tag{6}$$

$$|\overrightarrow{K-J}| = \sqrt{(X_k - X_j)^2 + (Y_k - Y_j)^2} \tag{7}$$

이므로, 최종적으로 구하는 θ는

$$\cos^{-1}\left(\frac{\overrightarrow{J-I} \cdot \overrightarrow{K-J}}{|\overrightarrow{J-I}|\times|\overrightarrow{K-J}|}\right) \tag{8}$$

이다. 여기서 도출되는 θ는 0과 π사이의 양의 값을 가지는 각 변화량이다.

이후 이어지는 노드의 이동으로 발생하는 방위각 변화는 가장 오래된 지점 (**Fig. 1**의 예에서는 I 지점)의 위치정보를 버리고, 새롭게 파악된 하나의 지점(노드의 현재 지점)에 대한 위치정보를 추가적으로 반영하여 식 (8)로부터 구할 수 있다.

3.2 상대방위각 변화방향 알고리즘

3.1절에서 구한 상대방위각 변화량 θ이 0이면 이전의 이동방향과 동일한 방향으로 직선 이동을 한 것을 뜻한다. θ이 π이면 이전의 이동방향과 정반대 방향으로 되돌아가는 형태의 이동을 의미한다. 그 외의 경우는 이동노드의 이동방향이 임의로 변하는 것이 가능하기 때문에 사실상 시계(clock-wise)방향 또는 반시계(counter clock-wise)방향 등 두 가지 방향 가운데 한 방향으로의 변화량을 나타낸다.

여기서 노드의 이동방향을 파악하는 알고리즘을 제시한다. 이것은 결국 노드가 시계 또는 반시계 방향으로 움직였는지

파악하는 알고리즘이다. 기본적으로 이번에 움직인 방향을 파악하기 위해서는 직전에 있었던 이동의 형태를 우선 파악해야 한다. 이를 위해 노드의 직전이동방향을 기준으로 상하, 좌우, 그리고 임의 방향으로 나누어 최종 이동방향을 파악한다.

첫 번째, **Fig. 3**의 예와 같이 노드가 직전에 상하로 움직인 경우이다. (a) 경우와 같이 위에서 아래 방향으로, 반대로 (b) 경우와 같이 아래에서 위 방향으로 이동한 경우이다. $\overrightarrow{IJ}$ 벡터가 Y 축과 평행하게 형성된 경우이다. 그 이후 다시 노드가 이동하여 현재 K 지점에 있다고 하자. 이때 J 지점에서 K 지점으로 이동하면서 발생한 방위각 변화량은 (a)와 (b) 경우 모두 θ이다.

이 상황에서 이동방향을 파악한다. $\overrightarrow{IJ}$ 벡터가 위에서 아래로 향할 때 노드가 J 지점에서 출발하여 이동한 K 지점이 $\overrightarrow{IJ}$ 벡터와 그것의 연장선으로 나뉜 공간에서 직선의 왼편에 있다면 노드는 θ만큼 시계방향으로 이동한 것이다((a)의 예). 반대로 직선의 오른편으로 이동했으면 노드는 반시계방향으로 이동한 것이 된다. 반면 $\overrightarrow{IJ}$ 벡터가 아래에서 위로 향하는 경우, $\overrightarrow{IJ}$ 벡터와 그것의 연장선으로 나뉜 공간에서 직선의 왼편으로 이동하면 반시계방향 이동이 되는 것이고((b)의 예), 오른편으로 이동하면 시계방향 이동이 되는 것이다.

이러한 조건과 방향을 수식으로 표현하면 다음과 같다. 이때 $\overrightarrow{IJ}$ 벡터는 시점 I 와 종점 J 간에 X값 변화는 없고, Y값 변화만 일어난 경우이다. 즉, 식 (9)가 참인 경우이다.

$$(X_j = X_i) \wedge (Y_j \neq Y_i) \tag{9}$$

노드의 직전 이동이 이럴 때 식 (10)이 양의 값을 가지면 최종 이동방향은 시계방향이다.

$$(X_k - X_j) \times (Y_j - Y_i) \tag{10}$$

반대로 식 (10)이 음이면 최종 이동방향은 반시계방향이다. 이에 대한 증명은 부록을 참조하라.

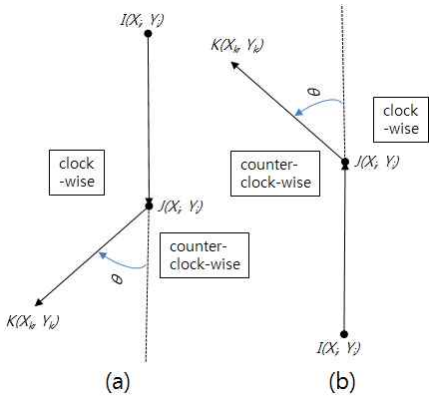

Fig. 3 Examples of movement of a node

두 번째, **Fig. 4**의 예와 같이 직전의 노드 움직임이 좌우로 이루어진 상황이다. (a) 경우는 우에서 좌로, (b) 경우는 좌에서 우로 이동한 경우이다. 여기서 $\overrightarrow{IJ}$ 벡터는 X 축과 평행하게 형성된다. 그 이후 이전 예와 동일하게 다시 노드가 이동하여 현재 K 지점에 있다고 하자. 이때 J 지점에서 K 지점으로 이동하면서 발생한 방위각 변화량은 (a)와 (b) 경우 모두 θ이다.

이 상황에서 앞의 예와 같이 이동방향을 고려한다. $\overrightarrow{IJ}$ 벡터가 우에서 좌로 향할 때 노드가 J 지점에서 출발하여 이동한 K 지점이 $\overrightarrow{IJ}$ 벡터와 그것의 연장선으로 나뉜 공간에서 직선의 위편에 있으면 노드는 θ만큼 시계방향으로 이동한 것이다((a)의 예). 반대로 직선의 아래로 이동했으면 노드는 반시계방향으로 이동한 것이 된다. 반면 $\overrightarrow{IJ}$ 벡터가 좌에서 우로 향하는 경우, $\overrightarrow{IJ}$ 벡터와 그것의 연장선으로 나뉜 공간에서 직선의 위로 이동하면 반시계방향으로 이동한 것이고((b)의 예), 아래로 이동하면 시계방향 이동이 되는 것이다.

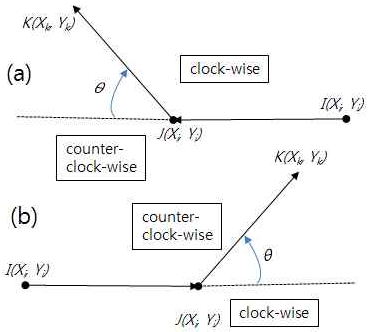

Fig. 4 Other examples of movement of a node

이 경우 $\overrightarrow{IJ}$ 벡터는 시점 I 와 종점 J 간에 Y값 변화는 없고, X값 변화만 일어난 경우이다. 즉, 식 (11)이 참인 경우이다.

$$(X_j \neq X_i) \wedge (Y_j = Y_i) \tag{11}$$

노드의 직전 이동이 이럴 때 식 (12)가 음의 값을 가지면 최종 이동방향은 시계방향이다.

$$(X_j - X_i) \times (Y_k - Y_j) \tag{12}$$

반대로 식 (12)가 양이면 최종 이동방향은 반시계방향이다. 이에 대한 증명도 부록을 참조하라.

마지막으로 직전에 노드의 이동이 임의로 이루어진 경우이다. 즉, 이 경우 $\overrightarrow{IJ}$ 벡터는 시점 I 와 종점 J 간에 X, Y값 모두 변화가 일어난 경우이다. 즉, 식 (13)이 참인 경우이다.

$$(X_j \neq X_i) \wedge (Y_j \neq Y_i) \tag{13}$$

이 조건에서 식 (14)가 음의 값을 가지면, 최종 이동방향은 시계방향이고, 반대로 양의 값을 가지면 반시계방향이다. 이에 대한 증명도 부록을 참조하기 바란다.

$$(X_j - X_i) \bullet (Y_k - Y(X_k)) \tag{14}$$

여기서 $Y(X_k)$은 X가 X_k일 때 $\overrightarrow{IJ}$ 벡터의 연장선의 Y값이다. 연장선은 식 (15)와 같다.

$$\frac{Y_j - Y_i}{X_j - X_i}(X_k - X_j) + Y_j \tag{15}$$

3.3 기존 상대방위각 알고리즘과 비교

이동노드가 전진하는 방향에 대한 방위각 변화가 내비게이

션 같은 위치 기반 응용에 중요한 정보이다. [3]의 알고리즘에서는 노드의 이동 궤적을 원호로만 이루어지는 것으로 가정하고 전진방향의 방위각 변화가 아니라 노드의 회전각을 구하고 있다. 이것으로 알고리즘이 적용될 수 있는 노드의 이동패턴이 제한적이 될 수밖에 없다.

본 논문에서 제안하는 알고리즘은 중심 회전각이 아닌 노드의 이동방향의 변화량과 방향을 구하는 것이므로 이 제약점이 없다. 그리고 기존 알고리즘에 비해 계산과정이 아주 단순해진 장점을 가지고 있다.

기존의 Kim(2013) 알고리즘에서는 법선을 구하는 과정이 ②, ④, ⑤ 단계에서 한 번씩 모두 세 차례 반복된다. 거기에다 원호의 중심점을 찾기 위해 ⑥ 단계에서 교차점을 구하는 과정도 있어야하고, 마지막으로 코사인 제2공식을 적용하는 여러 단계를 거치고 많은 계산 과정을 마쳐야 중심각 변화를 구할 수 있다. 반면 이 논문에서 제안한 알고리즘은 식 (8)에 의해 방위각 변화량을 구하고 이동방향은 식 (10), (12), 또는 (14)의 값의 부호값(양 또는 음)에 의해 결정되는 단순한 구조와 절차로 되어 있다.

추가적으로 기존의 알고리즘에서는 ② 단계에서 구한 법선과 가장 가까운 지점을 구하는 단계에서 원호의 중심을 구하는 데 오차(error)가 포함될 여지가 발생한다. 왜냐면 원호의 중심은 정확하게 그 법선 상에 존재하기 때문이다. 이로 인해 알고리즘에 의해 최종적으로 구하는 중심각 변화량에도 오차가 포함될 여지가 내재되는 것이다.

이 논문에서는 노드의 이동패턴에 대한 제약을 두지 않고, 이 패턴을 벡터로 인식하여 노드의 전진방향 방위각 변화를 벡터의 내적 공식을 적용하여 구함으로써 단 하나의 단계로 알고리즘을 단순화할 수 있었다.

Kim(2013)에서 이동 궤적을 원호라고 가정하고 여덟 단계를 거쳐 구하는 중심각을 이 논문에서 식 (8)을 통해 구한 벡터 사이각 Θ를 이용하면 식 (16)으로 간단하게 구할 수 있다.

$$(\text{중심각}) = 2\Theta \tag{16}$$

이에 대한 증명도 부록을 참조하기 바란다.

이 논문에서 제안하는 방위각과 이를 응용한 중심각 변화량 계산 알고리즘은 간단한 산술계산식으로 단순화되었기 때문에 매우 신속할 뿐만 아니라 계산과정에서 오류를 내포할 가능성이 없어져 정확하게 방위각 변화량을 구할 수 있다.

4. 결 론

위치기반 응용을 구현하기 위해서는 위치파악 기술이 필요하다. 내비게이션과 같은 길안내 응용에서는 이동노드의 전진방향에 대한 방위각 변화에 민감하게 반응해야 한다. 그러나 방위각 센서가 불능이거나 불량하게 작동하는 실내 환경에서 방위각 변화를 측정하는 알고리즘은 실제 이동노드의 전진방향 변화에 즉각적인 반응이 필요하다. 이 논문에서 제안한 간편한 방위각 알고리즘은 간단한 벡터의 내적과 사이각 산술식을 활용하여 방위각 변화량을 구하고 변화의 방향은 단순한 논리식에 의해 구하는 기법이 기존의 것과 차별화된 점이다. 그리고 기존의 계산과정에서 구조적으로 오류를 범할 수 있는 과정을 포함하고 있지 않기 때문에 신속하게 정확한 결과를 도출할 수 있다.

향후 이 논문에서 제안하는 알고리즘을 고도의 변화를 포함하는 3-D 위치 방위각 변화에 대한 연구로 확대할 필요가 있고, 더불어 알고리즘이 현실적으로 적용하였을 때 발생할 수 있는 위치정보의 오류 등에 적절히 대처할 수 있도록 정확도와 성능을 제고하는 방식에 대한 연구가 필요하다.

References

[1] Hopper A, Harter A, and Blackie T.(1993), “The Active Badge System,” Proceedings of the INTERCHI (International Conference on Human Factors in Computing Systems)-93, pp. 533-534.

[2] Kim, J. H., et al.(2013), “A Study on Indoor Navigation System using Localization based on Wireless Communication,” Journal of the Korean Society of Marine Engineering, Vol. 37, No. 1, pp. 114-120.

[3] Liu, H., et al.(2007), “Survey of Wireless Indoor Positioning Techniques and Systems”, IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, Vol. 37, No. 6, pp. 1067-1080.

[4] Woo, J. S.(2004), “A Study on Regulation of Telecommunication Service Providers related to the Changing Environment of Telecommunication Market”, Graduate School of Public Administrator, Seoul National University.

부 록

【상하이동 증명】 노드가 직전에 상하로 움직인 (예: **Fig. 3 (a)**와 **(b)**) 조건일 때 최근 이동으로 발생한 X와 Y값 변화에 따른 이동방향을 표로 정리하면 **Table 1 (a)**와 같다. 직전 이동의 Y 변화와 최종 이동의 X 변화에 대해 통합하여 식 (10)으로 표현하면, 그 값의 기호에 따라 **Table 1 (b)**와 같이 노드의 최종 이동 방향을 정리할 수 있다. ■

Table 1 Direction of node's previous movement was up or downward

(a) moving direction depending on changes of X, Y values

Previous Movement(condition)			Most Recent Movement	
X change	Y change	direction	X change	direction
$X_j = X_i$	$Y_j < Y_i$	↓	$X_k > X_j$	counter-clock
			$X_k < X_j$	clock
$X_j = X_i$	$Y_j > Y_i$	↑	$X_k > X_j$	clock
			$X_k < X_j$	counter-clock

(b) Integrated and simplified determination of moving direction

Previous Movement(condition)		Most Recent Movement	
X change	Y change	X, Y changes	direction
$X_j = X_i$	$Y_j \neq Y_i$	$(Y_j-Y_i)\times(X_k-X_j) > 0$	clock
		$(Y_j-Y_i)\times(X_k-X_j) < 0$	counter-clock

【좌우이동 증명】 노드가 직전에 좌우로 움직인 (예: **Fig. 4 (a)** 와 **(b)**) 조건일 때 최근 이동으로 발생한 X와 Y값 변화에 따른 이동방향을 표로 정리하면 **Table 2 (a)**와 같다. 직전 이동의 Y 변화와 최종 이동의 X 변화를 통합하여 식(12)로 표현하면, 그 값의 기호에 따라 **Table 2 (b)**와 같이 노드의 최종 이동 방향을 정리할 수 있다. ■

Table 2 Direction of node's previous movement was right or left-hand side

(a) moving direction depending on changes of X, Y values

Previous Movement(condition)			Most Recent Movement	
X change	Y change	direction	Y change	direction
$X_j > X_i$	$Y_j = Y_i$	→	$Y_k > Y_j$	counter-clock
			$Y_k < Y_j$	clock
$X_j < X_i$	$Y_j = Y_i$	←	$Y_k > Y_j$	clock
			$Y_k < Y_j$	counter-clock

(b) Integrated and simplified determination of moving direction

Previous Movement(condition)		Most Recent Movement	
X change	Y change	X, Y changes	direction
$X_j \neq X_i$	$Y_j = Y_i$	$(X_j-X_i)\times(Y_k-Y_j) > 0$	counter-clock
		$(X_j-X_i)\times(Y_k-Y_j) < 0$	clock

【임의이동 증명】 노드가 직전에 상하좌우가 아닌 다른 임의 방향으로 움직인 조건일 때 최근 이동으로 발생한 X와 Y값 변화에 따른 이동방향을 표로 정리하면 **Table 3 (a)**와 같다. 직전 이동의 Y 변화와 최종 이동의 X 변화를 통합하여 식 (14)로 표현하면, 그 값의 기호에 따라 **Table 3 (b)**와 같이 노드의 최종 이동 방향을 정리할 수 있다. ■

Table 3 Direction of node's previous movement was arbitrary

(a) moving direction depending on changes of X, Y values

Previous Movement(condition)			Most Recent Movement	
X change	Y change	direction	Y change	direction
$X_j > X_i$	$Y_j > Y_i$	↗	$Y_k > Y(X_k)$	counter-clock
			$Y_k < Y(X_k)$	clock
$X_j < X_i$	$Y_j > Y_i$	↖	$Y_k > Y(X_k)$	clock
			$Y_k < Y(X_k)$	counter-clock
$X_j < X_i$	$Y_j < Y_i$	↙	$Y_k > Y(X_k)$	clock
			$Y_k < Y(X_k)$	counter-clock
$X_j > X_i$	$Y_j < Y_i$	↘	$Y_k > Y(X_k)$	counter-clock
			$Y_k < Y(X_k)$	clock

(b) Integrated and simplified determination of moving direction

Previous Movement(condition)		Most Recent Movement	
X change	Y change	X, Y changes	direction
$X_j \neq X_i$	$Y_j \neq Y_i$	$(X_j-X_i)\times(Y_k-Y(X_k)) > 0$	counter-clock
		$(X_j-X_i)\times(Y_k-Y(X_k)) < 0$	clock

【식 (16) 증명】 **Fig. 5**에서 선분 IJ의 중앙을 지나는 법선과 선분 JK의 중앙을 지나는 법선이 서로 교차하는 O 지점 (I, J, K 지점을 원호 상의 세 점으로 하는 원의 중심점임.)을 중심으로 I 지점과 K 지점의 사이각 $\angle IOK$ (Figure 4에서 붉은 색으로 표현된 각)이 구하는 식 (16)의 (중심각) 이다.

$\triangle IOJ$와 $\triangle JOK$는 원호 위의 세 점이므로 이등변 삼각형이다. 그러므로 $\angle OIJ = \angle OJI = a$ 이고, $\angle OJK = \angle OKJ = b$ 라 할 수 있다. 이를 통해,

$$(\text{중심각}) + 2\cdot(a+b) = 2\pi \quad (17)$$

$$a + b = \pi - \Theta \quad (18)$$

임을 알 수 있다. 식 (10)을 식 (9)에 대입하면,

$$(\text{중심각}) = 2\pi - 2\pi + 2\Theta \quad (19)$$

이므로 식 (16)이 증명된다. ■

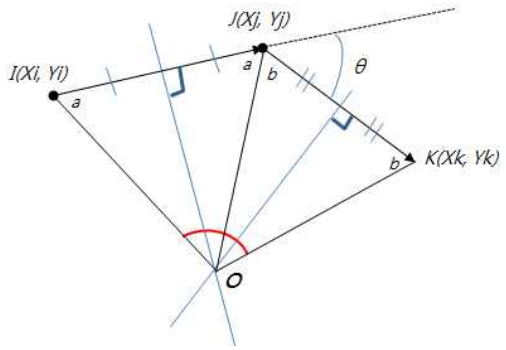

Fig. 5 Proof of Equation (16)

원고접수일 : 2014년 8월 1일
심사완료일 : 2014년 9월 24일
원고채택일 : 2014년 9월 25일