IMU 센서를 활용한 보행자 추측 항법 기반 위치탐지 및 방향안내 알고리즘 개발

김호중 [1], 오연주 [2], 김도균 [3], 임승훈 [4], 최상호 [5]
[1] 가톨릭대학교 정보통신전자공학부 학부생, 에스앤와이코퍼레이션 연구원
[2] 가톨릭대학교 정보통신전자공학부 학부생, 에스앤와이코퍼레이션 연구원
[3] 가톨릭대학교 정보통신전자공학부 학부생, 에스앤와이코퍼레이션 연구원
[4] 아주대학교 기계공학과 학부생, 에스앤와이코퍼레이션 연구원
[5] 가톨릭대학교 정보통신전자공학부 명예교수, 에스앤와이코퍼레이션 대표

jack777998@catholic.ac.kr, seuha0925@catholic.ac.kr, spacekdk33@catholic.ac.kr, limshoon0224@ajou.ac.kr

A Development of Location Detection and Direction Guidance Algorithms Using IMU Sensors

Ho-Jung Kim[1], Yeon-Ju Oh[2], Do-Kyun Kim[3] , Seung-Hoon Lim[4], Sang-Ho Choi[5]
[1] Dept. of Information Communication Electronics Engineering, The Catholic University of Korea
[2] Dept. of Information Communication Electronics Engineering, The Catholic University of Korea
[3] Dept. of Information Communication Electronics Engineering, The Catholic University of Korea
[4] Dept. of Mechanical Engineering, Ajou University
[5] Dept. of Information Communication Electronics Engineering, The Catholic University of Korea

요 약

GPS 는 실외에서는 안정적이고 정확도 높은 서비스를 제공하지만 실내와 같은 음영지역에서는 신호가 차단되어 연속적인 서비스가 불가하다. 본 연구에서는 IMU 센서를 활용한 보행자 추측 항법 기반 위치 추적 알고리즘을 제안한다. 또한 임계값 필터링과 가중 평균 필터를 적용하여 노이즈와 오차 누적을 최소화하였다. 제안 알고리즘은 Dart 언어 기반 Flutter Framework 로 구현하였으며, Android/iOS 크로스플랫폼 환경에서 동작하는 APP 프로토타입을 제작, 검증하였다 본 연구는 GPS 음영지역에서도 연속적인 서비스 제공을 가능하게 할 뿐 아니라, 향후 실내 내비게이션 및 시각장애인 보행 지원 등 다양한 응용에 적용 가능하게 한다.

1. 서론

스마트폰 기반 위치 추적 서비스는 현대 사회에 있어 필수적인 서비스로 자리 잡았다. 기존 GPS 기반 서비스는 실외에서 높은 정확도를 제공하는 반면, 실내, 터널, 지하도와 같은 음영지역에서는 신호가 차단되어 연속적인 서비스 제공이 어려운 근본적인 한계를 지닌다.

이를 해결하기 위해 IMU(inertial measurement unit) 기반 보행자 추측 항법(PDR, Pedestrian Dead Reckoning)이 대안으로 제시되고 있다. IMU 의 가속도계, 자이로스코프, 지자기계를 통해 실시간으로 움직임을 감지하며**[1]**, GPS 등 외부 인프라에 의지하지 않는다는 장점이 있다. 그러나 센서 드리프트 및 장시간 사용에 따른 오차 누적 문제가 존재한다.

본 연구는 임계값 및 가중 평균 필터를 적용해 오차 누적을 줄이고 안정성을 확보하였다. 또한 GPS 음영지역에서도 끊김 없는 위치 추적과 경로 안내를 가능하게 하여, 실내 내비게이션과 시각장애인 보행 지원 등 다양한 분야에 활용 가능함을 보인다.

아울러 제안 알고리즘은 Dart 기반 Flutter Framework 로 구현하였으며, Android/iOS 크로스플랫폼 환경에서 동작하는 APP 프로토타입을 제작, 검증하였다.

2. IMU 위치 추적 알고리즘

제안한 APP 프레임워크 설계를 위한 보행자 추측 항법 알고리즘은 IMU 센서 데이터를 통해 사용자의 실내외 위치를 추적한다. 알고리즘의 순서는 센서 데이터 수집 → 속력 계산 → 방향 계산 → 위치 계산 및 위경도 변환 → 지도 업데이트로 진행된다.

정확한 위치 추적을 위해 20ms 주기로 센서 데이터를 수집한다. IMU 의 가속도계 및 자이로스코프의

UserAccelerometerEvent 를 통해 사용자의 순수 가속도 데이터를, GyroscopeEvent 의 z 축 각속도를 이용해 회전 방향(yawRate)을, 나침반의 CompassEvent 를 통해 자북 기준의 절대 방위각 데이터를 수집한다.

가속도계 데이터를 기반으로 x,y 축 가속도 값을 수치 적분하여 사용자의 이동 속도를 계산한다. 센서 노이즈 제거를 위해 0.06 m/s^2 임계값 필터링을 적용하고, 임계값을 초과하는 경우, 가속도 차분 값에 샘플링 주기(dt=0.02s)를 곱한 값을 누적하여 사용자의 이동 속도를 얻는다.

velocityX += (currentpreAccX - preAccX) * dt;
velocityY += (currentpreAccY - preAccY) * dt;

5-tap 가중 평균 필터(가중치[0.1, 0.2, 0.3, 0.4, 0.5])를 적용하여 센서 노이즈를 최소화한다.

피타고라스 정리를 통해 x,y 축 방향별 속도를 최종 속력으로 변환한다.

currentSpeed = sqrt($velocityX^2$ + $velocityY^2$)

같은 방법으로 자이로스코프 센서로부터 z 축 각속도 데이터(event.z)를 수집하여 방향 계산을 수행한다. 노이즈 제거를 위해 0.3rad/s 임계값 필터링을 적용하고, 임계값을 초과하는 경우, 샘플링 주기(dt=0.02s)를 곱하여 회전 증분을 계산한다.

yawRatePerDt = (event.z * dt);
yawRate += -(yawRatePerDt + addYawRateNoise[0]);

이때, yawRate 는 라디안 값이므로, 180/π를 곱하여 각도(degree) 단위로 변환한다. 계산된 currentSpeed 와 yawRateDegree 를 이용하여 위치 변화량을 계산 후 누적한다.

px += (currentSpeed * cos(yawRate));
py += (currentSpeed * sin(yawRate));

최종적으로 누적된 위치 변화량으로부터 출발점 대비 총 이동거리와 방위각을 산출한다.

distance = sqrt(px^2 + py^2)
bearing = atan2(py, px)

3. 방향 안내 알고리즘

사용자가 경로를 따를 때 분기점이나 이탈 상황에서 정확한 방향을 계산하는 알고리즘이다. 본 알고리즘은 초기 설정 → 방향 판단 → 분기 처리의 세 단계로 구성된다.

경로 데이터 로딩시 인접 분기점 간 방위각을 계산하고, calculateBearing 함수로 현재 분기점에서 다음 분기점까지의 목표 방위각을 산출한 뒤 IMU 센서를 초기화한다.

GyroscopeEvent.z 의 z 축 각속도 데이터를 통해 현재 회전 상태(yawRate)를 수집한다. 수집된 데이터는 다음 식에 따라 누적되며, 2π 범위로 정규화하여 현재 방향을 추적한다.

yawRate += -(yawRatePerDt + noise)

이때 noise 는 센서 보정을 위한 잡음 제거 함수 addYawRateNoise 의 출력값이다.

사용자 방향이 목표 범위 내이면 분기점 통과 처리하고, 벗어나면 '경로 이탈 유도 각 계산'을 실행한다.

벡터 외적을 사용하여 사용자의 경로 이탈 방향을 판단한다. 본 알고리즘은 다음 절차를 수행한다.

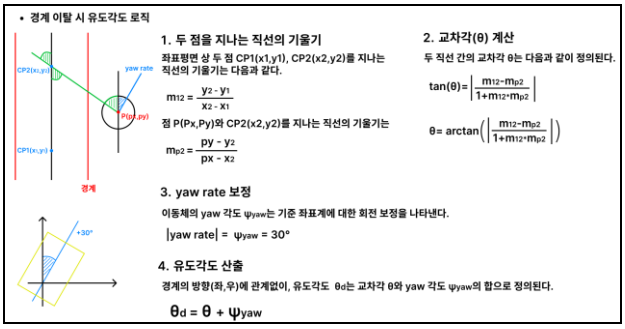
현재 분기점에서 다음 분기점으로의 경로 벡터와 사용자 위치 벡터를 생성한 뒤, 두 벡터의 외적을 통해 양수는 좌측 이탈, 음수는 이탈, 0 은 경로상 위치로 판단한다.

사용자가 복귀해야 할 정확한 각도를 계산하기 위해 구면삼각형을 구성하고, breakpoint 함수를 통해 각 삼각형의 내각을 계산한다.

angleToTarget 함수는 계산된 breakPointAngleP 와 센서 기반 회전 보정값 yawRateTurn2 를 결합하여 최종 유도각을 산출한다.

계산된 guidanceAngle 을 사용자 친화적인 시계 방위 표현으로 반환하는 과정에서 getGuidanceDirection 함수는 먼저 각도 정규화를 수행한다. 180° 를 초과할 경우 -180° ~180° 범위의 값으로 조정한다. 이후, 방향 인덱스를 계산하기 위해 guidanceAnlgle 에 15° 를 더해 360° 로 모듈 연산을 수행하고 30 으로 나누어 0 부터 11 까지의 정수 값으로 도출한다. 이렇게 계산된 인덱스를 통해 12 방향 시계 레이블 배열에서 해당하는 방향 텍스트를 반환한다.

특히 좌/우 이탈 시 서로 다른 방향 레이블 배열을 사용함으로써 직관적인 방향 안내를 제공한다



(그림 1) 방향 안내 수식의 보기 쉬운 정리

4. 결론

본 연구는 IMU 기반 보행자 추측 항법으로 GPS 음영지역에서도 실시간 위치 추적과 방향 안내가 가능함을 보인다. 다만 센서 드리프트와 비정상 보행에 따른 오차가 존재하며, 향후 WiFi, BLE 비콘 융합과 보행 상태 적응형 필터링으로 정확도를 개선할 예정이다. 제안된 알고리즘은 APP 프레임워크로 구현되어 시각장애인 보행 안내, 실내 내비게이션, 재난 구조 등 다양한 분야에 적용 가능하다.

사사문구

본 논문은 경기도 기술개발사업의 사업비지원(과제번호 PJ20250892)에 의해 수행되었습니다.

참고문헌

[1] Wonho Kang, Youngnam Han, "SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization", IEEE SENSORS JOURNAL, VOL. 15, NO. 5, pp. 2906~2916, 2015