(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0023284
(43) 공개일자 2024년02월21일

(51) 국제특허분류(Int. Cl.)
B60W 40/09 (2012.01) *B60W 30/18* (2006.01)
B60W 40/02 (2006.01) *B60W 50/00* (2006.01)
G06N 3/08 (2023.01)
(52) CPC특허분류
B60W 40/09 (2013.01)
B60W 30/18054 (2013.01)
(21) 출원번호 10-2022-0100794
(22) 출원일자 2022년08월11일
심사청구일자 2022년08월11일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
권장우
부산광역시 남구 오륙도로 85, 109동 2503호(용호동, 오륙도 에스케이뷰 아파트)
(74) 대리인
양성보

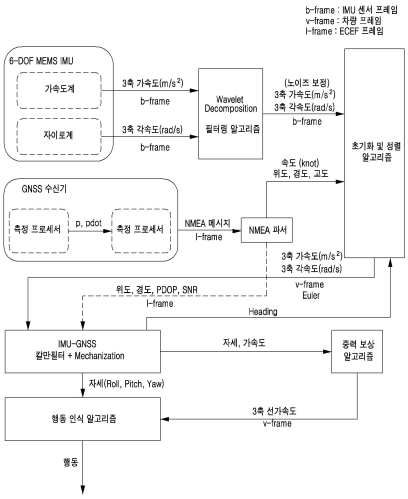
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 **딥러닝 기반의 행동 판단 모델을 통해 GNSS-IMU센서 퓨전을 이용하여 생성된 차량 정보로부터 운전자의 운전 습관을 판단하는 방법 및 시스템**

(57) 요 약

딥러닝 기반의 행동 판단 모델을 통해 GNSS-IMU센서 퓨전을 이용하여 생성된 차량 정보로부터 운전자의 운전 습관을 판단하는 방법 및 시스템이 개시된다. 일 실시예에 따른 컴퓨터 장치에 의해 수행되는 운전 습관 판단 방법은, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델에 입력받는 단계; 및 상기 딥러닝 기반의 행동 판단 모델을 이용하여 상기 차량 정보로부터 운전자의 운전 습관을 판단하는 단계를 포함할 수 있다.

대 표 도 - 도5



(52) CPC특허분류
B60W 40/02 (2013.01)
G06N 3/08 (2023.01)
B60W 2050/0005 (2013.01)
B60W 2050/0052 (2013.01)
B60W 2520/04 (2013.01)
B60W 2520/10 (2013.01)
B60W 2540/30 (2013.01)
B60W 2555/40 (2020.02)
B60W 2556/50 (2024.01)

명 세 서

청구범위

청구항 1

컴퓨터 장치에 의해 수행되는 운전 습관 판단 방법에 있어서,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델에 입력받는 단계; 및

상기 딥러닝 기반의 행동 판단 모델을 이용하여 상기 차량 정보로부터 운전자의 운전 습관을 판단하는 단계

를 포함하는 운전 습관 판단 방법.

청구항 2

제1항에 있어서,

상기 딥러닝 기반의 행동 판단 모델은,

상기 관성 측정 장치의 6축 데이터를 쿼터니언(quaternion)으로 변환하고, 상기 변환된 쿼터니언의 값을 일정 시간만큼 추출하여 컨볼루션을 수행하고, 상기 컨볼루션, 맥스 풀링 및 소프트맥스를 통과한 분류 점수가 임계치를 초과할 경우, 운전 행동 후보로 출력하도록 학습된 것을 특징으로 하는 운전 습관 판단 방법.

청구항 3

제1항에 있어서,

상기 운전자의 운전 습관을 판단하는 단계는,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 상기 딥러닝 기반의 행동 판단 모델과 운전 습관을 연동한 행동인식 데이터베이스를 이용하여 상기 운전자의 운전 습관을 판단하는 단계

를 포함하는 운전 습관 판단 방법.

청구항 4

제3항에 있어서,

상기 운전자의 운전 습관을 판단하는 단계는,

상기 딥러닝 기반의 행동 판단 모델을 이용하여 출력된 분류 점수와 행동인식 데이터베이스를 통해 검출한 운전자의 행동을 비교하여 가장 점수가 높은 행동을 운전자의 행동으로 출력하는 단계

를 포함하는 운전 습관 판단 방법.

청구항 5

제1항에 있어서,

상기 딥러닝 기반의 행동 판단 모델은,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 이용하여 행동 인식 기반의 차량 경로를 생성하고, 상기 생성된 행동 인식 기반의 차량 경로에 기초하여 운전자의 행동 특성을 분석하여 0부터 100사이의 실수로 정량화된 운전 성향 지수를 결과 데이터로서 출력하도록 학습된 것을 특징으로 하는 운전 습관 판단 방법.

청구항 6

제5항에 있어서,

상기 생성된 행동 인식 기반의 차량 경로는,

전처리 필터를 이용하여 잡음이 제거되는 것을 특징으로 하는 운전 습관 판단 방법.

청구항 7

제1항에 있어서,

상기 운전 습관 판단 방법은,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 이용하여 상기 딥러닝 기반의 행동 판단 모델의 학습을 위해 데이터 정규화, 데이터 동기화, 데이터 구획화를 포함하는 데이터 전처리 과정과, 데이터 라벨링 과정을 수행하는 단계를 더 포함하는 운전 습관 판단 방법.

청구항 8

제1항에 있어서,

상기 딥러닝 기반의 행동 판단 모델은, CNN 과 RNN 기반으로 구성되어,

CNN을 이용하여 상기 생성된 차량 정보로부터 특징 벡터를 추출하고, 상기 추출된 특징 벡터에 대하여 RNN을 이용하여 운전자의 행동 분석 결과를 0과 100사이의 운전 성향 지수로 출력하도록 학습된 것을 특징으로 하는 운전 습관 판단 방법.

청구항 9

컴퓨터 장치에 있어서,

메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서

를 포함하고,

상기 적어도 하나의 프로세서는,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델에 입력받고,

상기 딥러닝 기반의 행동 판단 모델을 이용하여 상기 차량 정보로부터 운전자의 운전 습관을 판단하는

것을 특징으로 하는 컴퓨터 장치.

청구항 10

제9항에 있어서,

상기 딥러닝 기반의 행동 판단 모델은,

상기 관성 측정 장치의 6축 데이터를 쿼터니언(quaternion)으로 변환하고, 상기 변환된 쿼터니언의 값을 일정 시간만큼 추출하여 컨볼루션을 수행하고, 상기 컨볼루션, 맥스 폴링 및 소프트맥스를 통과한 분류 점수가 임계치를 초과할 경우, 운전 행동 후보로 출력하도록 학습된 것을 특징으로 하는 컴퓨터 장치.

청구항 11

제9항에 있어서,

상기 적어도 하나의 프로세서는,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 상기 딥러닝 기반의 행동 판단 모델과 운전 습관을 연동한 행동인식 데이터베이스를 이용하여 상기 운전자의 운전 습관을 판단하는 것을 특징으로 하는 컴퓨터 장치.

청구항 12

제11항에 있어서,

상기 적어도 하나의 프로세서는,

상기 딥러닝 기반의 행동 판단 모델을 이용하여 출력된 분류 점수와 행동인식 데이터베이스를 통해 검출한 운전자의 행동을 비교하여 가장 점수가 높은 행동을 운전자의 행동으로 출력하는 것을 특징으로 하는 컴퓨터 장치.

청구항 13

제9항에 있어서,

상기 딥러닝 기반의 행동 판단 모델은,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 이용하여 행동 인식 기반의 차량 경로를 생성하고, 상기 생성된 행동 인식 기반의 차량 경로에 기초하여 운전자의 행동 특성을 분석하여 0부터 100사이의 실수로 정량화된 운전 성향 지수를 결과 데이터로서 출력하도록 학습된 것을 특징으로 하는 컴퓨터 장치.

청구항 14

제13항에 있어서,

상기 생성된 행동 인식 기반의 차량 경로는,

전처리 필터를 이용하여 잡음이 제거되는 것을 특징으로 하는 컴퓨터 장치.

청구항 15

제9항에 있어서,

GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 이용하여 상기 딥러닝 기반의 행동 판단 모델의 학습을 위해 데이터 정규화, 데이터 동기화, 데이터 구획화를 포함하는 데이터 전처리 과정과, 데이터 라벨링 과정을 수행하는 것을 더 포함하는 것을 특징으로 하는 컴퓨터 장치.

청구항 16

제9항에 있어서,

상기 딥러닝 기반의 행동 판단 모델은, CNN 과 RNN 기반으로 구성되어,

CNN을 이용하여 상기 생성된 차량 정보로부터 특징 벡터를 추출하고, 상기 추출된 특징 벡터에 대하여 RNN을 이용하여 운전자의 행동 분석 결과를 0과 100사이의 운전 성향 지수로 출력하도록 학습된 것을 포함하는 것을 특징으로 하는 컴퓨터 장치.

발명의 설명

기 술 분 야

[0001] 아래의 설명은 딥러닝 모델을 이용하여 운전자의 운전 습관을 판단하는 기술에 관한 것이다.

배 경 기 술

[0003] 최근 센서가 측정한 센싱 정보를 처리하는 기술과 카메라에 의해 촬영된 영상 정보를 처리하는 기술이 발전하면서 다양한 분야에서 이를 응용하고 있다. 특히, 차량 정보 제공 시스템에 있어서, 차량에 다중 카메라와 레이더 센서를 부착하여 카메라와 센서로부터 얻어진 정보들을 이용하여 운전자에게 제공함으로써, 운전자가 운전 중 볼 수 없거나 놓칠 수 있는 차량 주변의 정보들을 제공함으로써 안전 운전을 유도하고 있다. 하지만 기존의 차량 정보 제공 시스템은 차량 정보를 생성함에 있어서, 장애물이나 센서로 인한 오차 발생으로 인하여 정확도가 떨어진다는 문제점이 있었다.

[0004] 종래 자율주행항법기술(AUTONOMOUS DRIVING NAVIGATION SYSTEM)은 일반적인 주행 상황에서 목적지까지 경로상 부분 자동화 또는 완전 자율주행이 가능한 기술을 의미한다. 종래의 일반적인 자율주행기술은 자율주행 차량의 단기 경로를 생성하는 단계에서 주변 환경 정보를 기반으로 여러 개의 경로 중 최적으로 판단되는 경로를 추정하는 형태로 구성된다. 하지만, 종래 기술의 경우 운전자의 성향에 적합한 자율주행 경로 또는 자율주행서비스를 제공하지 못하여, 운전자에게 불편함을 유발하는 한계가 있다. 또한, 종래 자율주행 기술의 경우 자율주행 경로를 결정함에 있어 운전자의 운전 성향, 주행 이력에 대한 정보가 제대로 반영되지 못하여 운전자에 심리적으로 답답함, 불편함을 유발하는 경향이 커서 주행 이질감을 야기하는 한계가 있었다.

발명의 내용

해결하려는 과제

[0006] 딥러닝을 기반으로 운전자의 운전습관을 판단하기 위해 GNSS-IMU 센서를 이용하여 차량 정보를 생성하는 방법 및 시스템을 제공할 수 있다.

[0007] GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델을 통해 운전자의 운전습관을 판단할 수 있다.

과제의 해결 수단

[0009] 컴퓨터 장치에 의해 수행되는 운전 습관 판단 방법은, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델에 입력받는 단계; 및 상기 딥러닝 기반의 행동 판단 모델을 이용하여 상기 차량 정보로부터 운전자의 운전 습관을 판단하는 단계를 포함할 수 있다.

[0010] 상기 딥러닝 기반의 행동 판단 모델은, 상기 관성 측정 장치의 6축 데이터를 쿼터니언(quaternion)으로 변환하고, 상기 변환된 쿼터니언의 값을 일정 시간만큼 추출하여 컨볼루션을 수행하고, 상기 컨볼루션, 맥스 풀링 및 소프트맥스를 통과한 분류 점수가 임계치를 초과할 경우, 운전 행동 후보로 출력하도록 학습된 것일 수 있다.

[0011] 상기 운전자의 운전 습관을 판단하는 단계는, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 상기 딥러닝 기반의 행동 판단 모델과 운전 습관을 연동한 행동인식 데이터베이스를 이용하여 상기 운전자의 운전 습관을 판단하는 단계를 포함할 수 있다.

[0012] 상기 운전자의 운전 습관을 판단하는 단계는, 상기 딥러닝 기반의 행동 판단 모델을 이용하여 출력된 분류 점수와 행동인식 데이터베이스를 통해 검출한 운전자의 행동을 비교하여 가장 점수가 높은 행동을 운전자의 행동으로 출력하는 단계를 포함할 수 있다.

[0013] 상기 딥러닝 기반의 행동 판단 모델은, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 이용하여 행동 인식 기반의 차량 경로를 생성하고, 상기 생성된 행동 인식 기반의 차량 경로에 기초하여 운전자의 행동 특성을 분석하여 0부터 100사이의 실수로 정량화된 운전 성향 지수를 결과 데이터로서 출력하도록 학습된 것일 수 있다.

[0014] 상기 생성된 행동 인식 기반의 차량 경로는, 전처리 필터를 이용하여 잡음이 제거되는 것일 수 있다.

[0015] 상기 운전 습관 판단 방법은, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 이용하여 상기 딥러닝 기반의 행동 판단 모델의 학습을 위해 데이터 정규화, 데이터 동기화, 데이터 구획화를 포함하는 데이터 전처리 과정과, 데이터 라벨링 과정을 수행하는 단계를 더 포함할 수 있다.

[0016] 상기 딥러닝 기반의 행동 판단 모델은, CNN 과 RNN 기반으로 구성되어, CNN을 이용하여 상기 생성된 차량 정보로부터 특징 벡터를 추출하고, 상기 추출된 특징 벡터에 대하여 RNN을 이용하여 운전자의 행동 분석 결과를 0과 100사이의 운전 성향 지수로 출력하도록 학습된 것일 수 있다.

[0017] 컴퓨터 장치는, 메모리에 포함된 컴퓨터 판독가능한 명령들을 실행하도록 구성된 적어도 하나의 프로세서를 포함하고, 상기 적어도 하나의 프로세서는, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의

센서 퓨전을 이용하여 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델에 입력받고, 상기 딥러닝 기반의 행동 판단 모델을 이용하여 상기 차량 정보로부터 운전자의 운전 습관을 판단할 수 있다.

발명의 효과

[0019] 딥러닝을 기반으로 운전자의 운전습관을 판단하기 위해 GNSS-IMU 센서를 이용하여 특정 축(예를 들면, 헤딩)을 몰라도 정확하게 차량 정보를 생성할 수 있다.

[0020] GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델을 통해 운전자의 운전습관을 판단함으로써 계산량을 감소시킬 수 있다.

도면의 간단한 설명

[0022] 도 1은 GNSS을 설명하기 위한 도면이다.

도 2는 관성 측정 장치를 설명하기 위한 도면이다.

도 3은 각축에 대한 회전 오차를 설명하기 위한 도면이다.

도 4는 칼만 필터의 원리를 설명하기 위한 도면이다.

도 5는 일 실시예에 있어서, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)를 이용한 운전자의 행동 인식 동작을 설명하기 위한 도면이다.

도 6 및 도 7은 일 실시예에 있어서, 축 정렬이 수행되지 않은 상태를 설명하기 위한 도면이다.

도 8 및 도 9는 일 실시예에 있어서, 레벨링 알고리즘을 설명하기 위한 도면이다.

도 10은 일 실시예에 있어서, 레벨링 이후의 축 정렬 상태를 나타낸 도면이다.

도 11은 일 실시예에 있어서, 헤딩 정렬 알고리즘을 설명하기 위한 구성도이다.

도 12는 일 실시예에 있어서, 헤딩 값만큼 정렬 오차가 있는 차량을 나타낸 도면이다.

도 13은 일 실시예에 있어서, 헤딩 정렬 알고리즘의 순서도를 설명하기 위한 흐름도이다.

도 14는 일 실시예에 있어서, v-frame에 정렬 후의 축 정렬 상태를 나타낸 도면이다.

도 15는 일 실시예에 있어서, 웨이블릿 분해를 이용한 관성 측정 장치의 노이즈 필터링 동작을 설명하기 위한 도면이다.

도 16은 일 실시예에 있어서, 약결합을 이용한 GNSS-관성 측정 장치의 퓨전용 칼만 필터 및 기계화 결합 동작을 설명하기 위한 도면이다.

도 17은 일 실시예에 있어서, 기계화 식을 설명하기 위한 도면이다.

도 18은 일 실시예에 있어서, 딥러닝 기반의 운전습관 추정 동작을 설명하기 위한 도면이다.

도 19는 일 실시예에 있어서, MLP 구조를 설명하기 위한 도면이다.

도 20은 일 실시예에 있어서, 관성 측정 장치의 가속도계 CNN 구조를 나타낸 도면이다.

도 21은 일 실시예에 있어서, 하이브리드 운전습관 판단 동작을 설명하기 위한 도면이다.

도 22는 일 실시예에 있어서, 차량의 이동 경로를 시계열 데이터로 저장한 프로토타입 경로를 나타낸 도면이다.

도 23은 일 실시예에 있어서, 데이터 구획화를 설명하기 위한 도면이다.

도 24는 일 실시예에 있어서, 운전습관 판단 동작을 설명하기 위한 또 다른 도면이다.

도 25는 일 실시예에 있어서, 관성 측정 장치의 데이터를 시각화하기 위한 평가 도구를 설명하기 위한 도면이다.

도 26은 일 실시예에 따른 컴퓨터 장치의 구성을 설명하기 위한 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0023] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.

[0025] 도 5는 일 실시예에 있어서, GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)를 이용한 운전자의 행동 인식 동작을 설명하기 위한 도면이다.

[0026] 컴퓨터 장치는 GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 차량과 관련된 센서 정보를 획득할 수 있다. 컴퓨터 장치는 획득된 차량과 관련된 센서 정보에 기초하여 딥러닝을 기반으로 운전자의 운전습관을 판단하기 위한 차량 정보를 생성할 수 있다. 컴퓨터 장치는 딥러닝 기반의 행동 판단 모델을 이용하여 차량 정보로부터 운전자의 운전습관을 판단할 수 있다.

[0027] 실시예에서 차량은 운전자에 의해 운전되는 차량(승용차, 트럭, 버스, SUV 등), 자율주행 차량일 수 있다.

[0028] 우선적으로, GNSS와 관성 측정 장치(IMU)에 대하여 설명하기로 한다. 도 1을 참고하면, GNSS을 설명하기 위한 도면이다. 차량 위치 측위 시스템은 절대적인 위치 정보를 제공하는 위치 정보를 제공하는 위성항법 기술의 GNSS를 기반으로 한다. GNSS는 우주궤도를 돌고 있는 인공위성에서 발신하는 전파를 이용해 지구 전역에서 움직이는 물체의 위치, 고도, 속도를 계산하는 위성항법시스템이다. 국내에서는 미국에서 운용하는 GPS(Global Positioning System) 시스템에 의존하고 있으며, 차량의 절대적인 위치정보를 제공하기 때문에 차량용 항법 장치에 주로 사용된다. 주행하고 있는 차량에서 정확하고 연속적인 위치 정보를 획득하기 위하여 여러 센서 기술들이 통합되고 있다. 차량용 모션 센서 기술은 관성 센서의 가속도, 각속도를 적분하여 현재 위치를 추정하는 방법으로, 연속적인 정보를 제공하는 장점이 있으나 측정 데이터의 적분으로 인해 오차가 발생한다. 도 1은 GNSS에서의 차량 방향을 나타낸 것이다. 2DOF Bicycle Kinematic model, Dynamic model, 2 Wheel model과 같은 차량 모델을 통해 현재 위치를 추정하는 방법이 있다. INS는 이미 알고 있는 출발점에 대한 차량의 위치와 속도, 진행방향을 결정하는 시스템으로 3차원 공간정보를 측정하기 위해 자이로스코프(Gyroscope)와 가속도(Accelerometer) 각각 3개씩 내장한다. GPS와 INS를 융합하는 통합 항법 알고리즘은 약결합, 강결합, 초강결합 등의 방법이 있으며, 초강결합 방법은 수신기 내부의 신호추적 루프를 제어할 수 있는 환경에서 가능하고 약결합 방식은 GPS수신기에서 나오는 위치 결과를 바탕으로 IMU 측정치와의 통합이 이루어지는 형태로서, 계산량이 적은 장점이 있으나, 칼만 필터 구성 시에 준최적성 발생 문제가 존재한다.

[0029] 도 2를 참고하면, 관성 측정 장치를 설명하기 위한 도면이다. 관성 측정 장치(IMU)는 가속도계(acceleromter)와 자이로계(gyroscope)를 이용하여 가속도, 각속도 및 신체를 둘러싼 자기장을 측정하는 전자 장치이다. 일반적으로 3축 가속도, 3축 자이로 조합으로 각각의 센서의 값을 출력한다. 일부의 관성 측정 장치에는 3-축 지자기계(magnetometer) 센서가 포함되어 있다. 자이로스코프는 각속도 (rad/s)를 측정하고 회전각(degree)은 각속도의 적분으로 계산된다. 가속도계는 가속로 (m/s^2)를 측정하고, 중력 가속도를 분해하여 기울임의 정도 측정에 사용된다. 지자기계는 지자기를 측정하여 진북 기준, 상대적인 회전을 측정한다. 도 2의 왼쪽 도면은 3축 가속도계, 도 2의 오른쪽 도면은 3축 자이로계를 나타낸 것이다. 관성 측정 장치는 물체에 작용하는 가속력과 진동력, 충격력 등 동적 힘이 발생했을 때, 움직임의 변화에 따른 가속도의 변화를 감지한다는 점에서, 짧은 기간에 대한 오차는 작으나, 센서의 바이어스(bias) 및 드리프트(drift)가 포함된 가속도 값을 적분하므로 시간에 따라 오차가 누적되는 한계가 있다.

[0030] 이에, 정확한 자세 추정을 위해서는 차량과 관성 측정 장치가 동일한 방향으로 정렬되어야 한다. 부정확한 정렬은 노이즈의 누적을 극대화한다. 물리적인 정렬이 불가할 경우, 또는 관성 측정 장치의 부착 위치가 불명확할 경우에는 시스템 캘리브레이션을 진행해야 한다. 도 3을 참고하면, 각축에 대한 회전 오차를 설명하기 위한 도면이다. 관성 측정 장치의 정렬 오차는 Ψ_{IMU}와 이를 제자리에 고정하는 시스템 오차 Ψ_{SYS}를 이용하여 $\Psi_T = \sqrt{\Psi_{IMU}^2 + \Psi_{SYS}^2}$ 가 계산된다. 두 개의 오차는 일반적으로 분리가 되어 있어 시스템 전체의 정렬 오차는 두 오차의 제곱근의 합으로 표현이 가능하다. 높이 변동은 x축과 y축의 장착 스큐를 유발할 수 있는 기계적 변동중 하나이다.

[0031] 도 4는 칼만 필터의 원리를 설명하기 위한 도면이다. 칼만 필터는 1960년대 초 루돌프 칼만이 개발한 알고리즘으로 NASA의 아폴로 프로젝트에서 네비게이션 개발에 사용되었다. 일반적인 칼만 필터는 측정 모델이 선형성이 있고 가우시안 분포를 따를 때만 사용이 가능하다. 칼만 필터는 상태 예측과 측정 업데이트로 두 가지 상태를 지속적으로 반복함으로 물체의 상태를 추종한다. 상태 예측은 측정 업데이트에서 계산한 확률 분포와 물체 이동에 관한 확률 분포를 이용하여 현재 상태의 분포를 예측한다. 확률 분포의 평균은 두 평균의 합으로 표현이 되며, 분포의 분산은 두 개의 분산의 합이다. 일반적인 칼만 필터는 측정 모델의 선형성을 가정을 하며, 비선형 구조를 갖는 시스템에서는 선형성 가정을 완화한 확장 칼만 필터(Extended Kalman Filter)를 사용한다. 확장 칼만 필터는 모델의 선형성 대신에 상태 변화의 미분 가능성을 가정한다. 관찰 모델 (예측과 업데이트)와 상태 천이가 매우 비선형일 경우 무향 칼만 필터 (Unscented Kalman Filter)를 사용한다. 확장 칼만 필터는 평균과 근사치를 하나로 계산하는 반면에 무향 칼만 필터는 시그마 포인트로 불리는 여러 값의 평균과 근사치를 이용함으로 더욱 정밀한 예측이 가능하다.

[0032] 컴퓨터 장치는 관성 측정 장치로부터 가속도계와 자이로계를 이용하여 3축 가속도, 3축 각속도를 측정하고, 측정된 가속도에 대한 필터링 알고리즘을 이용하여 노이즈가 보정된 3축 가속도 및 3축 각속도를 포함하는 차량과 관련된 센서 정보를 획득할 수 있다.

[0033] 컴퓨터 장치는 GNSS로부터 측정 프로세서와 측위 프로세서를 이용하여 차량의 속도 정보, 위도 정보, 경도 정보 및 고도 정보를 포함하는 차량과 관련된 센서 정보를 획득할 수 있다.

[0034] 컴퓨터 장치는 관성 측정 장치로부터 획득된 3축 가속도 및 3축 각속도를 포함하는 차량과 관련된 센서 정보와, GNSS를 이용하여 획득된 속도 정보, 위도 정보, 경도 정보 및 고도 정보를 포함하는 차량과 관련된 센서 정보에 대해 자동 초기화 및 정렬 알고리즘을 실행할 수 있다.

[0035] 컴퓨터 장치는 정밀한 차량의 위치 정보, 자세 정보, 속도 정보를 추정하기 위하여 관성 측정 장치와 차량의 프레임 정렬을 수행할 수 있다. 프레임 정렬은 레벨링(leveling)과 헤딩 정렬로 구성될 수 있다. 도 8을 참고하면, 레벨링 알고리즘의 구성도이다. 레벨링은 지면을 기준으로 관성 측정 장치 및 차량 프레임의 자세 행렬 추종을 목적으로 하며, 헤딩 정렬은 차량의 진행 방향과 관성 측정 장치의 x-축 (앞)과 정렬을 목표로 한다. 이때, 레벨링 과정은 중력 가속도를 이용함으로 차량이 고른 평면에 정차가 되었을 시에만 진행될 수 있다. 차량 정차와 지면 검출은 GNSS의 속도, 고도로 측정을 한다. 헤딩 정렬은 차량 헤딩과 관성 측정 장치의 x-축 (앞)을 정렬 시키는 것을 목표로 하며 앞선 레벨링 과정이 요구된다. 정렬 과정은 $a^V = C_S^V a^S$ 으로 표현이 가능하며(a^V 는 IMU(b-frame) 기준 3축 가속 벡터, a^S 는 차량(v-frame) 기준 3축 가속 벡터, C_S^V는 좌표 변환 행렬) 두 프레임의 변환 행렬인 C_S^V 를 구하는 것을 목표로 한다.

[0036] 도 6 및 도 7은 일 실시예에 있어서, 축 정렬이 수행되지 않은 상태를 설명하기 위한 도면이다. 도 6은 축 정렬이 수행되지 않은 차량을 나타낸 것으로, 도 6의 빨간색은 차량의 축(v-frame)을 의미하며, 파란색은 장착 관성 측정 장치의 축(b-frame)을 의미한다. 도 7은 정렬 전 초기 상태를 나타낸 것으로, v-frame에서의 차량과 관성 측정 장치를 나타낸 것이다.

[0037] 도 9를 참고하면, 레벨링 알고리즘의 순서도이다. 정렬은 중력 가속도를 이용하여 정차된 상태에서의 초기화가 요구된다. 컴퓨터 장치는 GNSS로부터 차량의 속도 정보를 취득할 수 있다(910). 컴퓨터 장치는 취득된 차량의 속도 정보에 기초하여 차량이 정차되었는지 여부를 판단할 수 있다(920). 컴퓨터 장치는 차량이 정차되어 있는 것으로 판단됨에 따라 GNSS로부터 획득된 고도 정보와 GNSS로부터 획득된 이전의 고도 정보와의 고도 비교를 통해 고도차가 없을 경우, 평평한 지면에 정차된 것으로 인식하여 레벨링을 진행할 수 있다(930, 940, 950). 컴퓨터 장치는 이전 GNSS 측정치의 고도 비교를 통해 고도차가 없을 경우 균일한 지면으로 가정하며 레벨링을 진행할 수 있다. 정차되어 있는 차량에 적용하는 가속력은 중력이며, 지면에 향한다. 차량과 관성 측정 장치는 attitude matrix인 C_N^S는 $f_{iS}^S = -C_S^N g_S^N$ 을 품으로 취득이 가능하다(g_S^N은 관성 측정 장치 프레임의 중력 벡

터). C_N^S를 오일러 각으로 배열을 하면 $\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} \sin\theta \\ -\cos\theta\sin\phi \\ -\cos\theta\cos\phi \end{pmatrix} g_d$ 이며, g_d는 b-frame의 중력 가속도를 나타낸다(θ, ϕ는 각각 피치, 롤(pitch, roll)). Strapdown 알고리즘의 오차 방정식의 $V,\ \delta V^e,\ \omega,\ \delta\omega,\ \delta g$ (속도, 속도 오차, 각속도, 각속도 에러, 중력 바이어스)는 확장 칼만으로 필터링될 수 있다. 고른 지면에서 정차가 되면, 벡터는 $\begin{pmatrix} f_{xk} \\ f_{yk} \\ f_{zk} \end{pmatrix} = a \begin{pmatrix} a_{rx} \\ a_{ry} \\ a_{rz} \end{pmatrix} + (1-\alpha) \begin{pmatrix} f_{xk} \\ f_{yk} \\ f_{zk} \end{pmatrix}$ 로 표현될 수 있다(f_{ik}는 I 번째 축의 k 반복). 추가적으로 b-frame의 $\begin{pmatrix} f_x \\ f_y \\ f_z \end{pmatrix} = \begin{pmatrix} \sin\theta \\ -\cos\theta\sin\phi \\ -\cos\theta\cos\phi \end{pmatrix} g_d$ 로 정렬함으로써 g_d의 정보없이 θ, ϕ (RP)를 구할 수 있다. 즉,

$$\theta = \arctan\frac{-f_x}{\sqrt{(fy)^2 + (f_z^2)}},\ \phi = atan2(-f_{y,}, -f_z)$$

이다. 도 10을 참고하면, 레벨링 이후의 축 정렬 상태를 나타낸 도면이다. 롤(Roll)과 피치(pitch)의 값은 보정되어 있지만 차량의 헤딩과는 일치하지 않는 것을 확인할 수 있다.

[0038] 도 11을 참고하면, 헤딩 정렬 알고리즘을 설명하기 위한 구성도이다. 헤딩 정렬 알고리즘은 차량의 프레임(v-frame)과 관성 측정 장치의 프레임(b-frame)의 헤딩(heading)을 정렬을 목표로 한다. 헤딩 정렬은 차량이 직선 주행할 때에만 진행이 되며, 노이즈, 바이어스 필터링을 위해 1-D 칼만 필터를 사용한다. 임베디드 플랫폼에 최적화된 경량화 칼만 필터를 이용하여 헤딩 보정이 수행될 수 있다. 앞선 레벨링 알고리즘이 최소 1회 반복의 진행이 필요하며, 일부 좌표 변환이 된 데이터를 기반으로 정렬이 진행될 수 있다. 차량의 진행 방향 (직선 주행 확인)은 정렬된 자이로계의 z-축을 이용하며 차량의 회전이 0(즉, z-축 각속도가 0)인 경우에 GNSS상의 속도를 확인하고, 주행이 맞을 시 정렬을 진행할 수 있다. 이때, 자이로계, 가속도계는 같은 b-frame으로 정렬이 되어 있으므로 축 정렬은 동시에 진행될 수 있다. GNSS 상의 종방향 가속은 차량 헤딩의 x, y 분해로 가정이 가능하며, $\psi = \begin{cases} atan2(a_{y,}a_x) \\ \pi - atan2(a_{y,}a_x) \end{cases}$ (a_x, a_y는 정렬된 관성 측정 장치의 변화)로 표현될 수 있다. 도 12는 헤딩 값만큼 정렬 오차가 있는 차량을 나타낸 도면이다. 도 13은 일 실시예에 있어서, 헤딩 정렬 알고리즘의 순서도를 설명하기 위한 흐름도이다. 컴퓨터 장치는 레벨링이 진행됨에 따라 관성 측정 장치로부터 획득된 차량의 각속도 정보를 이용하여 차량 운행 방향을 확인할 수 있다. 컴퓨터 장치는 관성 측정 장치의 자이로계를 입력할 수 있다(1310). 컴퓨터 장치는 z-축 각속도를 획득할 수 있다(1320). 컴퓨터 장치는 차량이 직선 방향으로 운행 중으로 판단될 경우, GNSS로부터 차량의 속도 정보를 취득할 수 있다(1340). 컴퓨터 장치는 취득된 차량의 속도 정보를 이용하여 차량의 xy축으로 가속도를 취득할 수 있다(1350). 컴퓨터 장치는 취득된 가속도가 임계치를 초과할 경우, 차량의 프레임과 관성 측정 장치의 프레임의 헤딩 정렬을 수행할 수 있다(1360, 1370). 도 14를 참고하면, v-frame에 정렬 후의 축 정렬 상태를 나타낸 도면이다. 차량의 진행 방향(직선 주행 확인)은 정렬된 자이로계의 z-축을 이용하며 차량의 회전이 0 (즉, z-축 각속도가 0)인 경우에 GNSS상의 속도를 확인하고, 주행이 맞을 시 정렬을 진행할 수 있다. 이때, 자이로계, 가속도계는 같은 b-frame으로 정렬이 되어 있으므로 축 정렬은 동시에 진행될 수 있다. 이와 같은 방법으로 헤딩 정렬을 할 경우 2가지 단점이 존재한다. 이전 반복의 보정 정보가 없어 측정 오류가 존재할 경우 잘못된 값을 출력할 수 있으며, 노이즈/간섭 및 바이어스에 대한 보정이 불가하다. 이에, 컴퓨터 장치는 1-D 칼만 필터를 통해 단점을 해결할 수 있다. 이때, 1-D

칼만 필터 외에도 다양한 필터가 존재하나, 1-D 칼만 필터는 계산량이 적으므로 임베디드 플랫폼에 적절하다. 차량과 관성 측정 장치는 v-frame에 고정이 되어있어 공분산 행렬은 $R = \alpha_1.\beta + a_2.\gamma + a_3.(\psi_{k1} - \psi_k)$ 로 표현 가능하다. 칼만 필터의 입력값은 앞에서 계산된 차량의 헤딩 값(ψ)이며, 헤딩의 업데이트는 GNSS 기반의 탄젠트로 계산이 된다. GNSS의 음영 구역에서는 3장의 GNSS-IMU 센서 퓨전된 요(yaw) 값을 이용하여 업데이트를 진행할 수 있다. 상태 천이 행렬, 칼만 이득, 상태 전이 행렬 전파는 각각 다음과 같다.

$$P = \phi P^{+}\phi^{T} + Q,$$

$$K = PH^{T}(HPH^{T} + R)$$

$$\widehat{\psi^{+}} = \hat{\psi} + K(\psi - H\hat{\psi})$$

$$P^{+} = (1 - KH)P$$

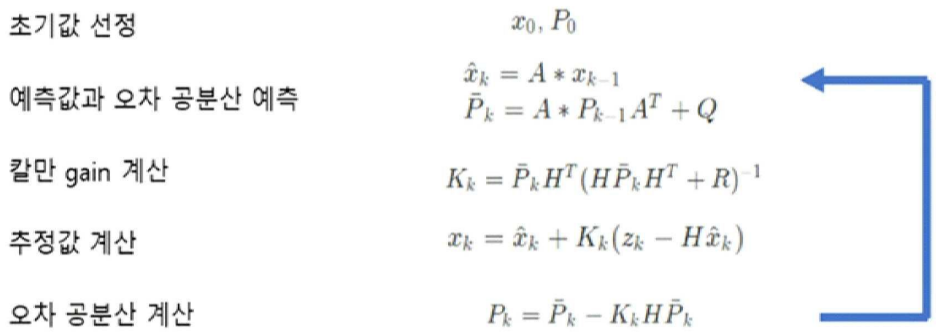
[0039]

[0040] 도 15를 참고하면, 웨이블릿 분해를 이용한 관성 측정 장치의 노이즈 필터링 동작을 설명하기 위한 도면이다. 컴퓨터 장치는 관성 측정 장치의 3축 가속도 및 3축 각속도에 대한 웨이블릿 분해(Wavelet Decomposition)를 이용하여 관성 측정 장치의 노이즈를 제거할 수 있다. 저가 MEMS 관성 측정 장치는 다양한 노이즈와 바이어스에 노출이 되어있다. 자이로계는 일반적으로 저주파 영역에서 드리프트 에러(drift error)가 있으며, 가속도계는 고주파 영역에서 많은 노이즈가 발생한다. 일반적으로 상보 필터(complementary filter)를 이용하여 이 둘의 특성을 역 이용하여 보정이 수행될 수 있다. 웨이블릿(Wavelet)은 시간 및 주파수 도메인(domain)에서 노이즈를 관찰하기 용이하다. 프리 필터링(Pre-filtering) 없이 INS 기계화(mechanization) 알고리즘에 입력이 들어가게 되면, 부정확한 자세가 출력될 수 있다. 필터링의 가장 중요한 특성은, 노이즈 최소화와 동시에 원 신호의 출력이다.

[0041] 웨이블릿 필터링은 다음과 같은 순서로 진행될 수 있다. 1)가속계, 자이로계 센서 데이터를 각 축으로 wavelet multi-resolution analysis(WMRA)를 진행(시간 도메인에서의 노이즈는 일정한 패턴이 있음)하고, 2)임계값을 정하여, 일정 임계치를 넘는 신호를 제거하고, 3)제거된 신호중에 잔여 노이즈 분석 및 원신호 복원을 진행하고, 4)노이즈는 다양한 주파수 대역에서 나타나기 때문에, DWT(discrete wavelet transform)을 이용하여 노이즈 분석을 하고, 원 신호를 출력한다.

[0042] 도 16을 참고하면, 약결합을 이용한 GNSS-관성 측정 장치의 퓨전용 칼만 필터 및 기계화 결합 동작을 설명하기 위한 도면이다. 컴퓨터 장치는 MEMS 관성 측정 장치에서 출력되는 $\Delta\Theta, \Delta v$ (3축 가속도, 3축 각속도)를 6-dof 기계화 식(mechanization equation)를 이용하여 차량의 위치 정보, 속도 정보, 자세 정보로 변환할 수 있다(*b*-frame을 *ecef*-frame로 변환). 컴퓨터 장치는 프리 필터링된 센서값을 이용하여 INS 기계화를 진행할 수 있다. 컴퓨터 장치는 GPS의 위치, 속도를 이용하여 상기 단계에서 도출된 차량의 위치 정보, 속도 정보, 자세 정보를 업데이트할 수 있다. 도 17을 참고하면, 기계화 식을 설명하기 위한 도면이다. 가속도 계는 $\mathrm{f}^{\mathrm{b}} = \mathrm{f}^{\mathrm{b}} + \quad (\mathrm{f}^{\mathrm{b}})$, 자이로 계는 $\omega_{ib}^{b} = \omega_{ib}^{b} + \nabla w_{ib}^{b}$ $^{*}\nabla$ 이고, ω 값은 rate 벡터 값이다(ω 값은 system alignment로 보강이 가능하다). $l - frame([\Phi, \lambda, \gamma])$ 속도 벡터는 $v^{n} = [v_n, v_e, v_d]$ 로 표현될 수 있다. l-frame 에서의 차량의 높이는 다음과 같이 표현을 하여 PRY로 출력될 수 있다.

[0043] 다음은 전체 프로세스의 요약을 나타낸 것이다.



[0044]

[0045] 컴퓨터 장치는 GNSS의 위치 이동을 이용하여 헤딩을 구할 수 있다. 컴퓨터 장치는 진북 데이터와 관성 측정 장치의 퓨전을 위하여 관성 측정 장치의 자이로계로부터 각속도를 입력 받아 적분을 진행할 수 있다. 컴퓨터 장치는 자이로계의 단순 적분으로 인한 요 드리프트(yaw drift)와 헤딩(heading) 정렬 오차에 대해 칼만 필터 및 기계화 알고리즘 기반의 웨이블릿 필터를 사용할 수 있다. 컴퓨터 장치는 초기 상태를 차량의 헤딩(heading)으로 정하며, 관성 측정 장치의 각속도 적분, bicycle 모델을 이용한 (가속도계 기반) 방향 벡터와 GNSS의 위치 변화에서 나온 헤딩으로 업데이트 할 수 있다. 컴퓨터 장치는 터널과 같은 신호 음영 구역을 HDOF, VDOF로 인지하여 칼만 이득을 강제적으로 수정할 수 있다.

[0046] 이때, 초기값은 위치와 자세로 설정하며, 칼만 필터의 추정 상태와 공분산(EKF)은 다음과 같이 표현될 수 있다.

$$\hat{\boldsymbol{x}}_{k|k-1} = f(\hat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_k)$$
$$\boldsymbol{P}_{k|k-1} = \boldsymbol{F}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{F}_k^\top + \boldsymbol{Q}_k$$

[0047]

[0048] 혁신 및 혁신 공분산은 다음과 같이 표현될 수 있다.

$$\tilde{\boldsymbol{y}}_k = \boldsymbol{z}_k - h(\hat{\boldsymbol{x}}_{k|k-1})$$
$$\boldsymbol{S}_k = \boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top + \boldsymbol{R}_k$$

[0049]

[0050] 칼만 이득 (GNSS의 상태에 따라 칼만 이득을 강제적으로 수정을 하며 GNSS 상태는 PDOP $pdop = \sqrt{hdop^2 + vdop^2}$ 와 SNR을 사용한다.

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^\top \boldsymbol{S}_k^{-1}$$

[0051]

[0052] 업데이트된 상태 추정 및 추정 공분산이 도출될 수 있다(NHC 특성을 이용). 추정 PVA가 출력될 수 있다.

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \tilde{\boldsymbol{y}}_k$$
$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1}$$

[0053]

[0054] 도 18은 일 실시예에 있어서, 딥러닝 기반의 운전습관 추정 동작을 설명하기 위한 도면이다.

[0055] 컴퓨터 장치는 운전 행동 분류(Driving behavior classification)를 이산형 데이터로 분류하기 위한 딥러닝용 딥 뉴럴 네트워크를 구현할 수 있다. 이때, 다양한 딥뉴럴 네트워크 적용을 통한 최적 네트워크 선정 및 구현될 수 있으며, 임베디드 시스템 탑재를 위한 경량화 모델이 적용될 수 있다. 컴퓨터 장치는 차량 정보(센서 정보)의 분석을 통한 최적 학습 데이터를 구축할 수 있다. 또한, 컴퓨터 장치는 운전습관 추청 인식률 향상을 위한 휴리스틱 판단기법 하이브리드 모델을 제공할 수 있다. 또한, 컴퓨터 장치는 노이즈에 강인한 딥 뉴럴 네트워크를 구현할 수 있다.

[0056] 도 19를 참고하면, MLP 구조를 설명하기 위한 도면이다. 딥 러닝은 머신 러닝의 한 방법으로, 학습 과정이 필

요하며, 학습 과정에서 얻은 가중치로 추론을 진행한다. 딥러닝은 특정 형식의 인공 신경망(ANN)을 사용하며, 우선 샘플 데이터를 통한 학습이 필요하다. 인공신경망(ANN)은 서로 연결된 "뉴런" 계층으로 구성을 하며 계층은 크게 입력, 은닉, 출력 계층으로 구성된다. 입력 및 출력을 위한 가시적인 계층 사이에 존재하는 수백 개의 추가적인 은닉층은 가중치 및 바이어스를 갖고 있다. 단층 퍼셉트론의 형태에서 은닉층이 1개 이상 추가된 신경망을 다층 퍼셉트론(MLP)이라고 한다. 다층 퍼셉트론은 피드 포워드 신경망(Feed Forward Neural Network, FFNN)의 가장 기본적인 형태이다. MLP에서는 인접한 층의 퍼셉트론 간의 연결은 있어도 같은 층의 퍼셉트론끼리의 연결은 없다. 특히 순방향으로 이동함으로 이전 계층에 다시 연결되는 피드백(feedback)이 없다. 일례로, CNN(Convolutional Neural Network)는 컨볼루션(convolution)이 포함된 NN을 의미한다. 컨볼루션은 임의의 필터를 이용하여 사진과 같은 데이터에 필터를 옮겨가며 데이터를 추출하는 과정이다. CNN은 이미지 데이터를 평면화 과정없이 컨볼루션을 진행함으로 공간 정보가 유지가 되고, 이는 효율적이고 높은 정확도를 보인다. MLP는 일반적으로 행렬 데이터에 사용이 되며, 계산량이 CNN에 비해 적고 CNN은 일반적으로 사진 데이터에 사용이 되며, 정확도가 높은 대신 계산량이 많다. 경량화 시스템에 MLP를 많이 사용하며, 단순 구조의 CNN도 사용된다.

[0057] 컴퓨터 장치는 필터링과 정렬이 된 데이터를 수집 및 상황별로 가공하고, 이를 활용한 경량화된 운전 행동 분류 네트워크로 구성된 딥러닝 기반의 행동 판단 모델을 개발할 수 있다. 이때, 딥러닝 기반의 행동 판단 모델에 3축 가속도, 자이로계의 RPY 및 이동 위치가 입력 데이터로 입력될 수 있다. 다시 말해서, 입력 데이터는 관성 측정 장치의 3축 가속도계, 3축 자이로계, 일정 시간의 위치 이동 데이터($\Sigma_{n-x}^{n}(p_{t_n}-p_{t_{n-1}})$)이다. 딥러닝 기반의 행동 판단 모델은 종단간 학습을 수행하며, 네트워크의 손실은 운전 행동이고, 인코딩은 원-핫 인코딩을 수행할 수 있다. 딥러닝 기반의 행동 판단 모델은 데이터 입력층을 입력+쿼터니언(quaternion)으로 변환하고, 각 축의 데이터와 행렬곱 및 바이어스를 추가하고, 입력 데이터에 은닉층을 적용하고, 출력층의 소프트맥스를 통한 운전 행동을 구분하도록 구성될 수 있다.

[0058] 이때, 위치 이동 데이터의 인코딩은 현 시점(t)부터 n 시간 전(t-n) 사이의 위치 샘플을 추출하고, LLA를 ENU 좌표로 변환한 후, 2-D Spline 보간법을 이용하여 cuve fitting 후 spline 변수를 네트워크에 입력할 수 있다.

[0059] 컴퓨터 장치는 딥러닝용 CNN 기반 종단간 네트워크를 구성할 수 있다. 컴퓨터 장치는 시계열 데이터의 계산량을 고려하여 소형 임베디드 플랫폼에 적합한 2D 입력 CNN을 구성할 수 있다. 이때, CNN 이외에도 RNN, LSTM 등 대체 가능하다. 컴퓨터 장치는 CNN 기반 딥뉴럴 네트워크에 적용 가능한 2D 시계열 변환 알고리즘을 적용할 수 있다. 도 20을 참고하면, 관성 측정 장치의 가속도계 CNN 구조를 나타낸 도면이다. 컴퓨터 장치는 MEMS 관성 측정 장치의 6-축 데이터 (NHC 시스템으로 일부 데이터를 제외)를 쿼터니언(quaternion)으로 변환할 수 있다. 이때, Gimbal lock외에 Euler각도로 표현을 하게 되면 물리적으로 작은 변화가 수치상 크게 나타난다. 컴퓨터 장치는 변환된 값을 일정 (Data x T) 시간 T 만큼 추출을 하여 컨볼루션을 수행할 수 있다. 컴퓨터 장치는 맥스 풀링(Max pooling) 후 일정 크기의 은닉층 및 소프트 맥스(softmax)를 통과함에 따라 출력된 결과 데이터가 일정 임계값을 초과할 경우, 운전 행동 후보로 출력할 수 있다. 이때, Sensor alignment의 초기화 상태를 확인을 위하여 스프트맥스 전의 회귀(regression) 값이 저장될 수 있다.

[0060] 도 21을 참고하면, 하이브리드 운전습관 판단 동작을 설명하기 위한 도면이다. 컴퓨터 장치는 운전습관 추정 인식률을 향상시키기 위하여 휴리스틱 판단을 위한 하이브리드 운전습관 행동 판단 모델을 구성할 수 있다. 컴퓨터 장치는 특정 관성 측정 장치-GNSS 데이터와 운전습관을 연동한 행동인식 DB를 제작하여 규칙 기반의 행동 인식 알고리즘을 실행할 수 있다. 컴퓨터 장치는 하이브리드 운전습관 행동 판단 모델에서 출력 계층의 소프트맥스를 통과함에 따라 출력된 분류 점수와 행동인식 데이터베이스를 통해 검출한 운전자의 행동을 비교하여 가장 점수가 높은 행동을 운전자의 행동으로 출력할 수 있다. 이때, 임계치는 미세한 센서 노이즈를 제거하는 high-pass 필터 역할과 계산량이 많은 MLP 또는 CNN의 사용을 줄임으로써 평균 계산량을 감소시킬 수 있다.

[0061] 이에, 컴퓨터 장치는 운전습관 인식 향을 위한 NN 보조 판단 기법 모델을 이용하여 시스템의 효율을 높이고 탐지 정확도를 향상시킬 수 있다. 또한, 특정 임계치를 설정하여, 해당 임계치를 초과할 경우에만 NN을 통과시킬 수 있다.

[0062] 운전자 행동 인식은 다양한 주행 환경 속에서 운전자의 운전 특성 및 행동을 파악하는 것으로 운전 경로뿐만 아니라, 각 구간에서 차량의 이동속도 및 가감속, 회전에 대한 정보를 기반으로 수행된다. 도 22는 차량의 이동 경로를 시계열 데이터로 저장한 프로토타입(Prototype) 경로를 나타낸 것이다. 프로토타입 경로는 시계열 데이

터로 차량의 이동 속도 및 가감속, 회전에 대한 정보를 포함할 수 있다. 컴퓨터 장치는 차량의 이동 속도 및 가감속, 회전에 대한 정보를 딥러닝 기반의 행동 판단 모델의 입력 데이터로 사용함에 따라 운전자의 행동 특성을 분석하고, 0부터 100사이의 실수로 정량화된 운전 성향 지수를 출력으로 반환할 수 있다.

[0063] 컴퓨터 장치는 관성 측정 장치와 GNSS를 이용한 행동 인식 프로토타입 경로를 생성할 수 있다. 컴퓨터 장치는 운전자 데이터를 관성 측정 장치와 GNSS를 이용하여 취득하여 행동 인식 프로토타입 경로를 생성할 수 있다. 이때, 생성된 프로토타입 경로는 노면, 자동차의 상태, 통신장비들 사이에서 발생하는 레이턴시(latency) 등의 외부 요소에 영향을 받기 때문에 잡음이 포함될 수 있다. 외부요인의 영향은 운전자의 행동 인식을 분석하는 모델을 학습시키는 데에 방해가 되며, 정확한 분석을 방해할 수 있기 때문에 Moving Gaussian과 같은 전처리 필터를 생성된 프로토타입 경로에서 잡음을 제거하는 작업이 선행되어야 한다.

[0064] 컴퓨터 장치는 데이터 전처리 및 라벨링을 수행할 수 있다. 컴퓨터 장치는 데이터 전처리 과정으로 데이터 정규화(Data Normalization), 데이터 동기화(Data Synchronization), 데이터 구획화(Data Segmentation)를 수행할 수 있다. 프로토타입 경로는 시계열 데이터의 형태를 가지고 있으며, 딥러닝 모델의 학습에 알맞게 변형하기 위해 데이터 정규화, 데이터 동기화, 데이터 구획화를 거쳐 전체 데이터를 단위 시간으로 나누어야 한다.

[0065] 데이터 정규화는 수집된 프로토타입 경로의 각 센서 데이터를 일정 구간의 실수로 맵핑하는 작업으로, 각 센서 데이터가 손상되지 않도록 설계되어야 한다. 컴퓨터 장치는 여러가지 센서의 특징을 각자 반영하기 위해, 전체 센서에 대한 정규화가 아닌, 개별 센서 단위에서의 정규화를 수행하여 데이터 값을 정규화할 수 있다. 정규화 이전 데이터는 모든 포인트의 정보를 그대로 가지고 있기 때문에 전체 데이터의 특성을 판단하기 어려운 반면, 정규화 이후 데이터는 전체 데이터에 대한 포인트의 정보를 가지고 있어 전체 데이터 대비 특성을 파악하기 용이하다. 센서 데이터의 값을 특정 영역으로 구성하기 위해 평균과 표준편차를 구하고 이를 이용하여 모든 데이터를 정규화할 수 있다.

[0066] 데이터 동기화는 수집된 모든 센서 신호의 시간 도메인을 일치시키는 작업이다. 센서 데이터는 각기 다른 샘플링 레이트(Sampling Rate), 타임 스탬프(Time Stamp)를 가지게 되는데 이와 같이 서로 다른 시간 도메인을 가진 정보를 그대로 입력에 사용할 경우, 분석 과정에서 같은 상황에 대해 잘못된 시간 영역으로 해석하게 되는 문제가 상존하게 된다. 동기화 이전 데이터는 시간 영역이 서로 다르기 때문에 같은 포인트에 대해 서로 다른 시간의 정보를 가지고 있으나, 동기화 이후 데이터는 시간 영역이 모두 동일하기 때문에 같은 포인트에 대해 같은 시간의 정보를 가지고 있다. 컴퓨터 장치는 각기 다른 신호들을 모두 동일한 시간 간격을 갖도록 샘플링하는 과정을 통해 센서들의 시간 도메인을 동기화할 수 있다. 이때, 시간 간격은 수집된 센서 데이터 중 샘플링 레이트가 가장 작은 센서를 기준으로 수행하여, 시간 데이터의 손실이 발생하는 것을 막도록 구성될 수 있으며, 설정된 시간 간격이 너무 작은 경우, 딥러닝 모델이 처리해야할 데이터의 규모가 너무 방대해져 전반적인 특성을 파악하지 못하고 지역적 특성(Locality)에 과적합되는 문제를 발생시킬 수 있으며 시간 간격이 너무 큰 경우, 실제로 분석해야 할 사건이 분석 시간 간격 사이에 발생하여 정밀한 분석을 방해하는 요인으로 작동할 수 있기 때문에 딥러닝 모델에 적절한 시간 간격을 설정하는 과정이 필요하다.

[0067] 도 23을 참고하면, 데이터 구획화는 시계열 데이터를 포함한 연속형 데이터 분석에서 많이 사용되는 전처리 기법으로 데이터를 일정 크기로 나누는 작업으로 나뉘어진 각 데이터를 세그멘트(Segment)라고 부른다. 구획화 작업은 오버래핑(overlapping) 유무에 따라 2가지 방법이 존재하게 되며 오버래핑을 포함한다. 데이터 구획화 방식과 포함하지 않고 분리하는 구획화 방식이 존재한다. 도 23의 왼쪽 도면은 분석 세그먼트, 오른쪽 도면은 오버랩을 포함한 데이터 구획화를 나타낸 것이다. 오버랩을 포함하는 방식은 딥러닝 모델에서 사용할 수 있는 데이터의 양을 증가시켜주며, 이전 세그멘트의 정보를 다음 세그멘트 분석에 사용할 수 있어 시간 단위로 변화하는 데이터의 특성을 더 잘 반영할 수 있게 해준다. 컴퓨터 장치는 특정 영역의 정보뿐만 아니라 이전 시간 세그먼트 대비 변화가 중요시되기 때문에 오버랩을 포함하여 데이터를 구획화를 수행할 수 있다.

[0068] 표 1은 세그멘트 결정 수식들을 나타낸 것이다.

$$n = \frac{l - N}{m + 1}$$

n: 세그먼트의 수; l: 데이터 길이; N: 데이터 포인트 수; m: 이동 간격

$$n = (N-1) \times (L_{seg} - L_{overlap}) + L_{seg}$$

n: 세그먼트의 수; $L_{overlap}$: overlapping 길이;

L_{seg}: 세그먼트 길이; N: 세그먼트 개수

$$x_k(i) = x[i + (h-1)M],$$
$$\text{for } i = 1,2,\ldots,k; h = 1,2,\ldots,H$$

h: h번째 세그먼트; M: 세그먼트간 overlapping부분의 데이터 포인트 수;
k: 세그먼트 길이; H: 세그먼트 개수

[0069]

[0070] 컴퓨터 장치는 적절한 분석 단위 시간을 선정하고 오버랩을 수행할 시프트(Shift)의 크기를 선정하여 전체 시계열 데이터를 딥러닝 모델에 사용할 수 있는 형태로 구획화하여 구성할 수 있다.

[0071] 다음으로, 컴퓨터 장치는 데이터 라벨링(Data Labeling)을 수행할 수 있다. 운전자 행동에 대한 정량적인 라벨링을 생성하기 위해, 주행 시 전문 평가자가 함께 탑승하여 5점척도법을 활용한 설문기반 운전 행동을 수치화할 수 있다. 평가자의 설문 문항을 구성하기 위해, 주행 방식 및 운전자의 행동 양식을 표현할 수 있는 척도가 제시되며 모든 문항 배점의 합의 가중평균을 활용하여 점수를 생성할 수 있다. 생성된 점수는 전체 문항의 최대 점수합 대비 주행자의 점수를 이용해 0부터 100사이의 실수로 정규화되고 이는 설정한 운전 성향 지수와 동일한 정보를 가지게 된다.

[0072] 도 24를 참고하면, 운전습관 판단 동작을 설명하기 위한 또 다른 도면이다. 컴퓨터 장치는 딥러닝 기반의 행동 판단 모델에 대해 컨볼루션 뉴럴 네트워크(CNN; Convolution Neural Network)와 순환 신경망(RNN; Recurrent Neural Network)를 복합적으로 활용할 수 있다. CNN은 영상 처리에 작업에 사용되기 위해 고안된 모델로 입력된 데이터의 형태를 분석하고 이를 지역적 특성(Locality)로 반영할 수 있는 특징을 가지는데, 실시예의 신호들은 서로 다른 파형을 가지고 있고 이 파형을 분석하기 위해 CNN 구조를 활용한다. RNN은 시계열 분석 작업에 사용되기 위해 고안된 모델로 입력 데이터의 시간 세그멘트 뿐만 아니라 이전 분기의 데이터를 복합적으로 고려하여 전체 시간 흐름에 대한 데이터의 변화인 전역적 특성(Globality)를 반영할 수 있다는 특징을 가진다.

[0073] CNN을 활용한 파형 분석에 대하여 설명하기로 한다. 컴퓨터 장치는 전체 데이터를 시간 세그멘트 단위로 구획화하여 입력으로 사용할 수 있다. CNN은 각 세그멘트 단위로 입력된 센서 데이터의 파형을 분석하기 위해 컨볼루션 연산과 맥스풀링 연산을 반복적으로 수행할 수 있다. 입력된 센서 데이터는 CNN을 거쳐 특징 공간(Feature Space)로 맵핑되며, 맵핑된 데이터는 센서 데이터에 내재된 운전자의 특성 정보들을 가지고 있다.

[0074] RNN을 활용한 시계열 특성 분석에 대하여 설명하기로 한다. CNN을 통해 특징 공간으로 맵핑된 센서 데이터를 특징 벡터라고 하며, 이를 그대로 분석할 경우 시계열성을 반영할 수 없다는 단점이 존재한다. 컴퓨터 장치는 CNN을 통해 맵핑된 특징 벡터를 RNN의 입력으로 하여 시계열성을 반영한 운전자의 행동 분석을 수행할 수 있다. RNN은 특정 시간 영역, t 분기의 특징 벡터를 입력받아 해당 시간의 운전자의 행동을 분석하는데, 이때, 재귀적인 구조를 통해 이전 시간의 특징 벡터의 정보를 활용할 수 있다.

[0075] 컴퓨터 장치는 CNN을 통해 세그멘트 단위로 입력되는 센서 데이터를 압축하여 특징 벡터를 생성하고, RNN을 통해 특징 벡터 사이의 시계열성을 고려하여 운전자의 행동 분석 결과를 0과 100사이의 실수인 운전 성향 지수로 출력할 수 있다.

[0076] 도 25를 참고하면, 관성 측정 장치의 데이터를 시각화하기 위한 평가 도구를 설명하기 위한 도면이다. 컴퓨터 장치는 노이즈 제거 필터링 알고리즘 검증을 위한 시각화하기 위해 실제 관성 측정 장치의 센서 데이터를 분석하기 위해 관성 측정 장치의 센서 데이터가 저장된 데이터베이스와 통신하는 데이터 통신 프로세스를 구성할 수 있다. 컴퓨터 장치는 관성 측정 장치의 센서 데이터를 시각화하기 위한 평가 도구를 구성할 수 있다. 컴퓨터

장치는 관성 측정 장치의 센서 데이터를 활용한 운전자 행동 분석 딥러닝 모델의 결과를 시각화할 수 있다. 컴퓨터 장치는 관성 측정 장치의 센서 모듈로부터 시간별로 측정된 센서값을 저장할 데이터베이스를 설계하고 데이터 통신을 관리할 수 있다. 컴퓨터 장치는 관성 측정 장치의 센서 데이터를 저장한 데이터베이스로부터 관성 측정 장치의 센서값을 전송받아 데이터를 시각화 알고리즘을 통하여 시각화할 수 있다. 컴퓨터 장치는 노이즈 및 보정 필터 적용 전과 적용 후 데이터를 시각화를 통해 비교할 수 있다. 컴퓨터 장치는 데이터 딥러닝 연산을 위한 데이터 전처리 결과의 시각화를 통하여 그래프로 표현할 수 있다. 컴퓨터 장치는 관성 측정 장치의 센서 데이터를 딥러닝 기반의 행동 판단 모델을 통하여 분석 후 결과로 나오는 운동자 행동 점수(Driving Behavior Score)를 시간별로 시각화할 수 있다.

[0077] 도 26은 일 실시예에 따른 컴퓨터 장치의 구성을 설명하기 위한 블록도이다.

[0078] 컴퓨터 장치(100)는 GNSS(Global Navigation Satellite System)와 관성 측정 장치(IMU)의 센서 퓨전을 이용하여 차량과 관련된 센서 정보를 획득하고, 획득된 차량과 관련된 센서 정보에 기초하여 딥러닝을 기반으로 운전자의 운전습관을 판단하기 위한 차량 정보를 생성하고, 생성된 차량 정보를 딥러닝 기반의 행동 판단 모델을 이용하여 차량 정보로부터 운전자의 운전습관을 판단하는 동작을 수행할 수 있다. 컴퓨터 장치(100)는 메모리(2610) 및 프로세서(2620)를 포함할 수 있다.

[0079] 메모리(2610)는 컴퓨터 판독가능 명령어, 데이터 구조, 프로그램 모듈 또는 기타 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현된 휘발성 및 비휘발성, 분리형 및 비분리형 매체를 모두 포함하여 구성될 수 있으며, 컴퓨터 장치(100)의 적어도 하나의 다른 구성요소에 관계된 명령어 또는 데이터를 저장한다.

[0080] 프로세서(2620)는 중앙처리장치, 애플리케이션 프로세서, 또는 커뮤니케이션 프로세서 중 하나 또는 그 이상을 포함할 수 있다. 예를 들면, 프로세서(2620)는 컴퓨터 장치(100)의 적어도 하나의 다른 구성요소들의 제어 및/또는 통신에 관한 연산이나 데이터 처리를 실행할 수 있다.

[0081] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0082] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0083] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를

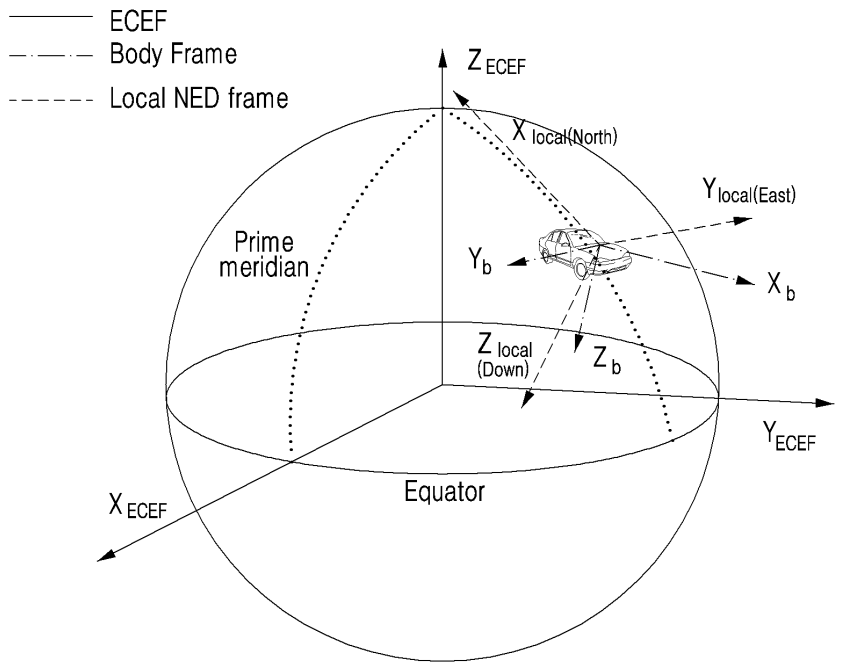
포함한다.

[0084] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

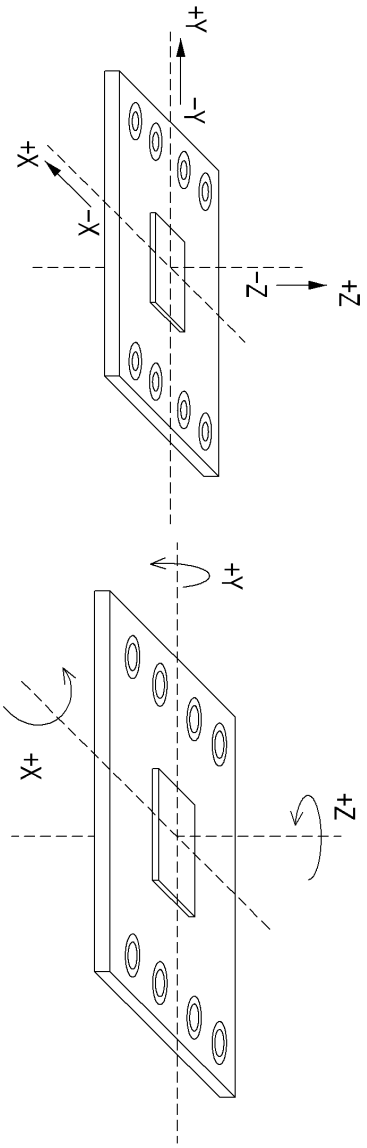
[0085] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

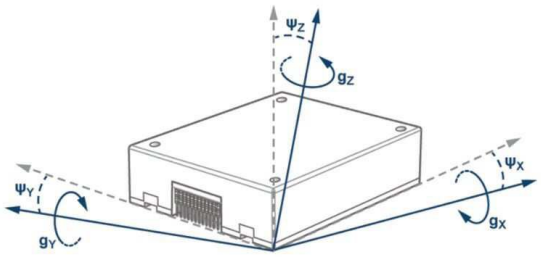
도면1



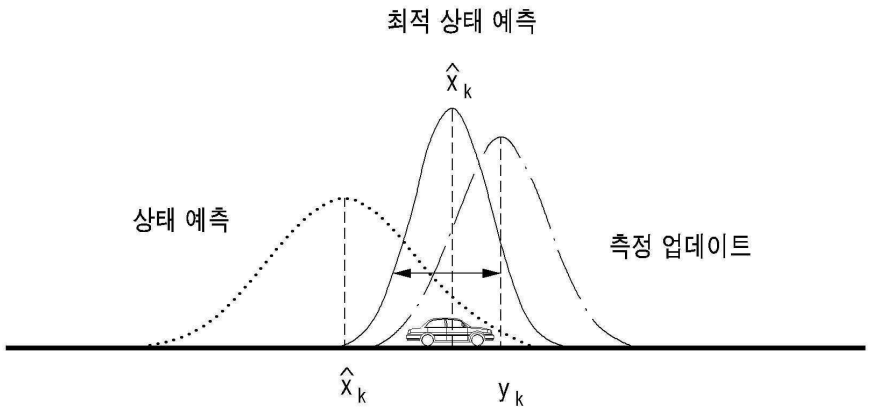
도면2



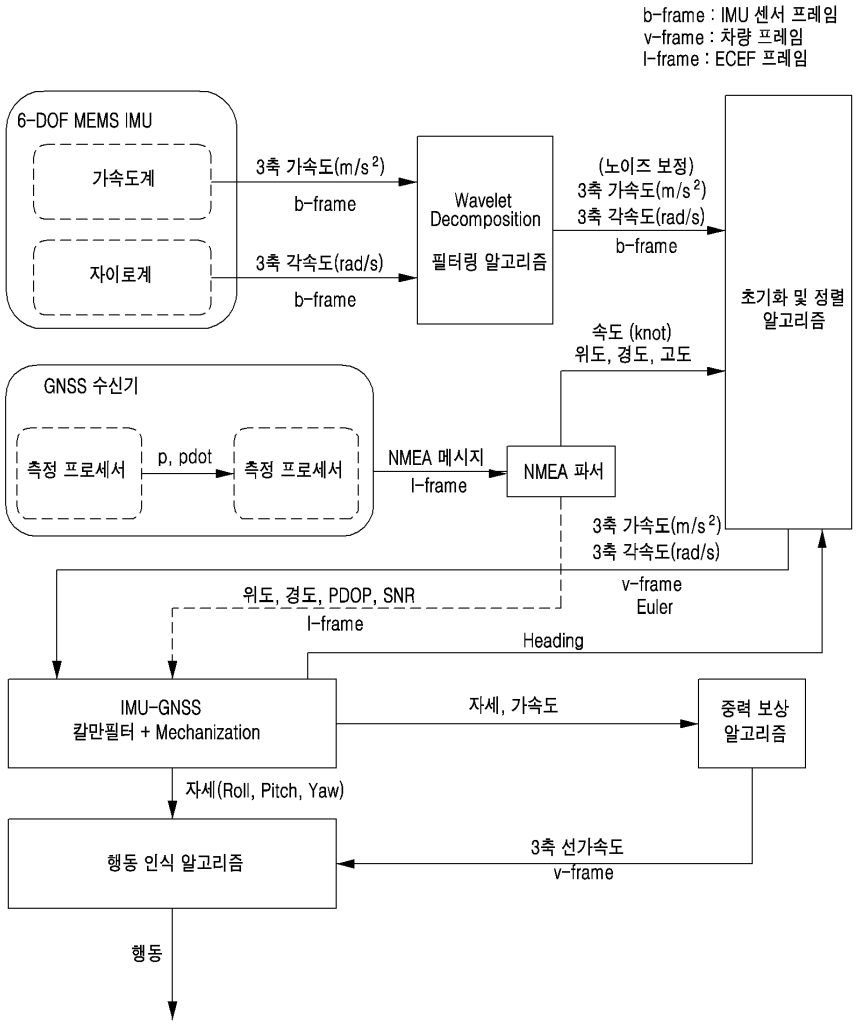
도면3



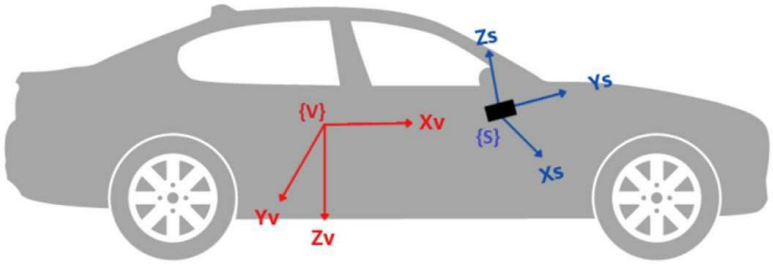
도면4


최적 상태 예측
$\hat{x}_k$
상태 예측
측정 업데이트
$\hat{x}_k$
y_k

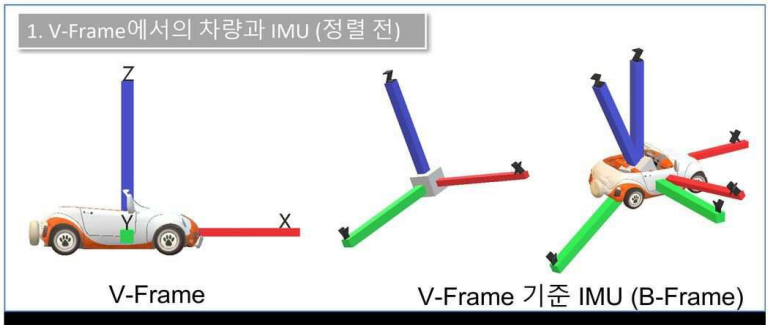
도면5


b-frame : IMU 센서 프레임
v-frame : 차량 프레임
l-frame : ECEF 프레임
6-DOF MEMS IMU
가속도계
자이로계
3축 가속도(m/s²)
b-frame
3축 각속도(rad/s)
b-frame
Wavelet Decomposition
필터링 알고리즘
(노이즈 보정)
3축 가속도(m/s²)
3축 각속도(rad/s)
b-frame
초기화 및 정렬 알고리즘
속도 (knot)
위도, 경도, 고도
GNSS 수신기
측정 프로세서
p, pdot
측정 프로세서
NMEA 메시지
l-frame
NMEA 파서
3축 가속도(m/s²)
3축 각속도(rad/s)
v-frame
Euler
위도, 경도, PDOP, SNR
l-frame
Heading
IMU-GNSS
칼만필터 + Mechanization
자세, 가속도
중력 보상 알고리즘
자세(Roll, Pitch, Yaw)
행동 인식 알고리즘
3축 선가속도
v-frame
행동

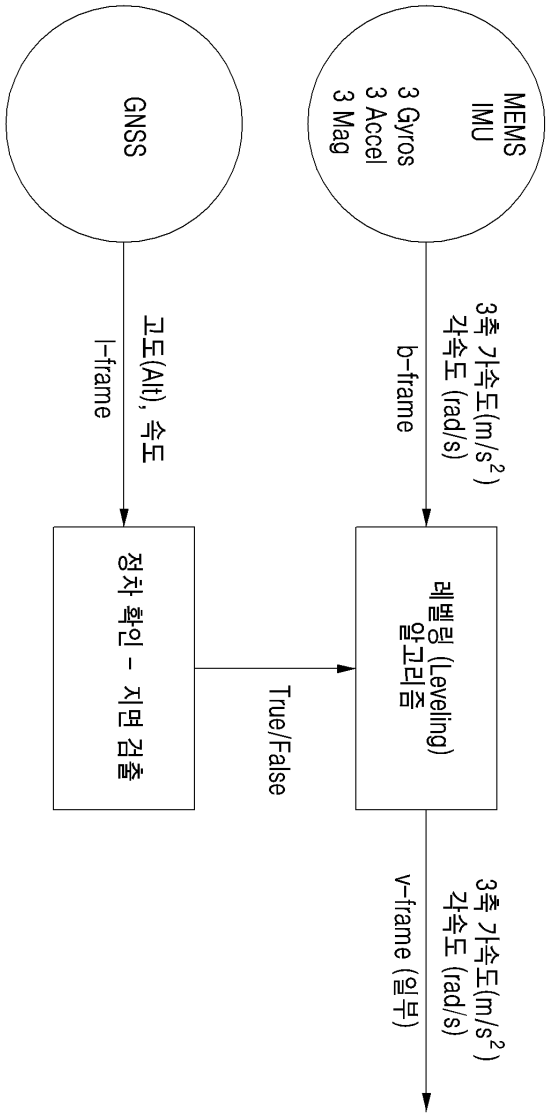
도면6


Zs
Ys
{V}
Xv
{S}
Xs
Yv
Zv

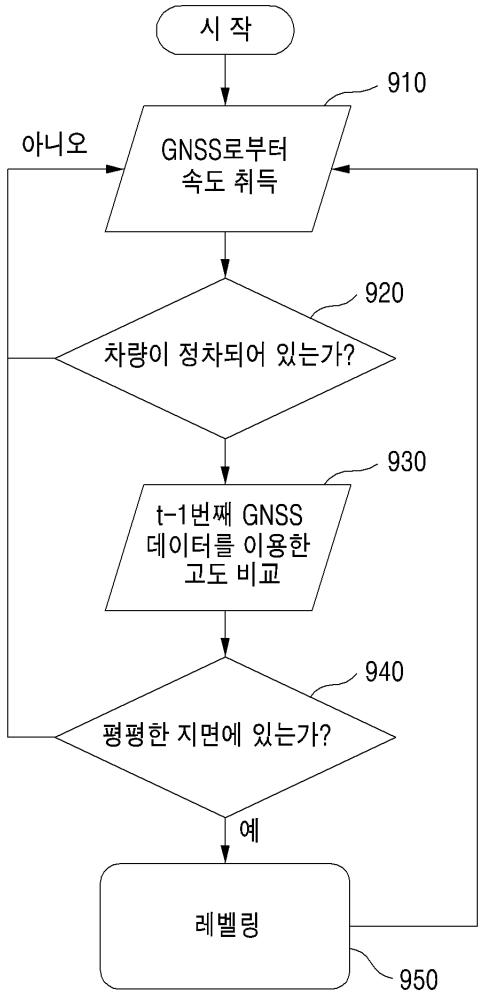
도면7


1. V-Frame에서의 차량과 IMU (정렬 전)
Z
Y
X
V-Frame
V-Frame 기준 IMU (B-Frame)

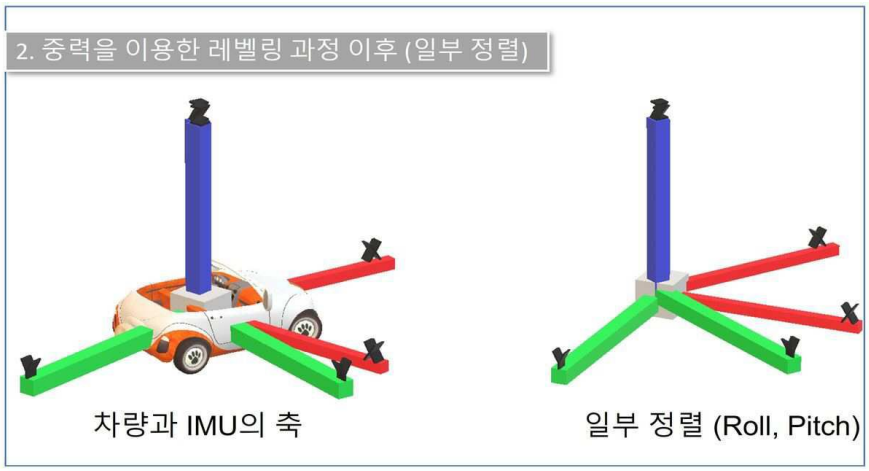
도면8


MEMS
IMU
3 Gyros
3 Accel
3 Mag
GNSS
3축 가속도(m/s²)
각속도 (rad/s)
b-frame
고도(Alt), 속도
l-frame
레벨링 (Leveling)
알고리즘
정차 확인 - 지면 검출
True/False
3축 가속도(m/s²)
각속도 (rad/s)
v-frame (일부)

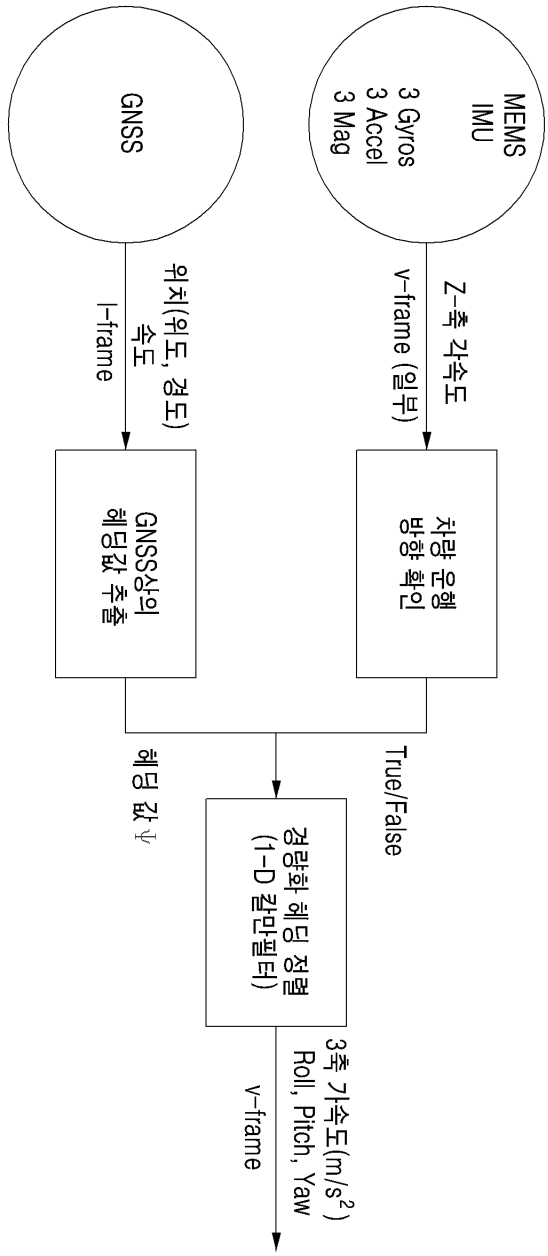
도면9


시 작
910
GNSS로부터
속도 취득
아니오
920
차량이 정차되어 있는가?
930
t-1번째 GNSS
데이터를 이용한
고도 비교
940
평평한 지면에 있는가?
예
레벨링
950

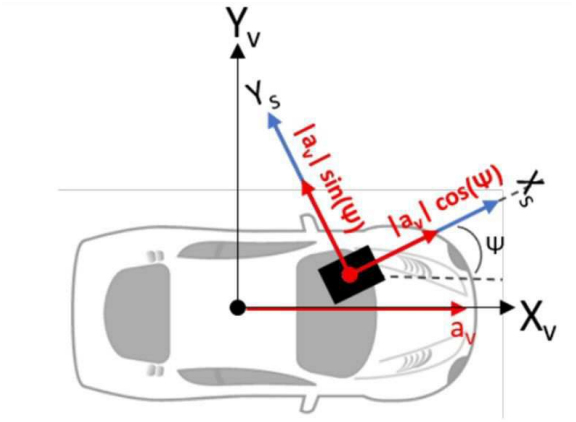
도면10


2. 중력을 이용한 레벨링 과정 이후 (일부 정렬)
차량과 IMU의 축
일부 정렬 (Roll, Pitch)

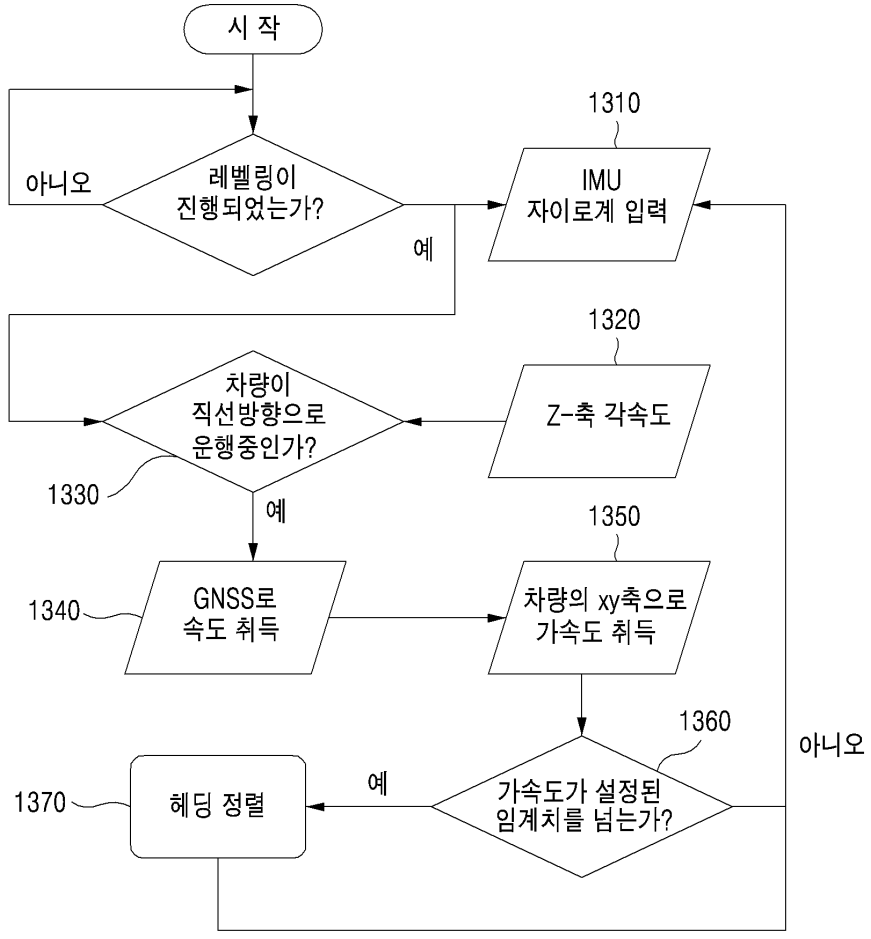
도면11


GNSS
MEMS
IMU
3 Gyros
3 Accel
3 Mag
위치(위도, 경도)
속도
l-frame
v-frame (일부)
Z-축 각속도
GNSS상의
헤딩값 추출
차량 운행
방향 확인
헤딩 값
True/False
경량화 헤딩 정렬
(1-D 칼만필터)
3축 가속도(m/s²)
Roll, Pitch, Yaw
v-frame

도면12


Yv
Ys
|av| sin(ψ)
|av| cos(ψ)
Ys
ψ
Xv
av

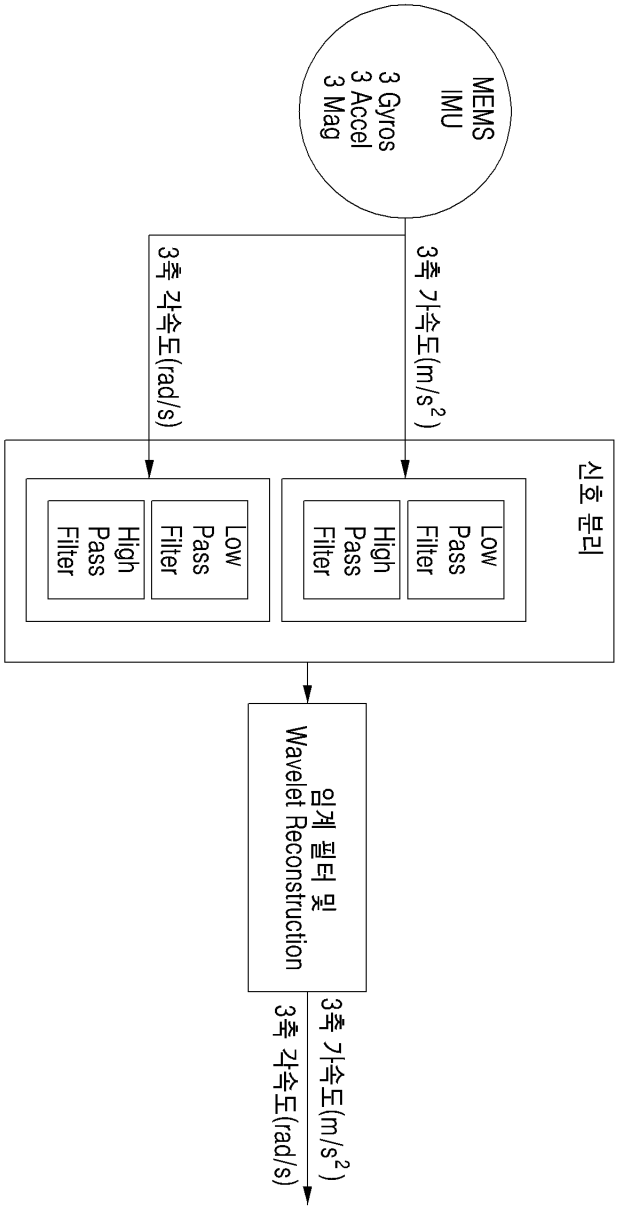
도면13


시 작
1310
아니오
레벨링이
진행되었는가?
IMU
자이로계 입력
예
1320
차량이
직선방향으로
운행중인가?
Z-축 각속도
1330
예
1350
1340
GNSS로
속도 취득
차량의 xy축으로
가속도 취득
1360
아니오
1370
헤딩 정렬
예
가속도가 설정된
임계치를 넘는가?

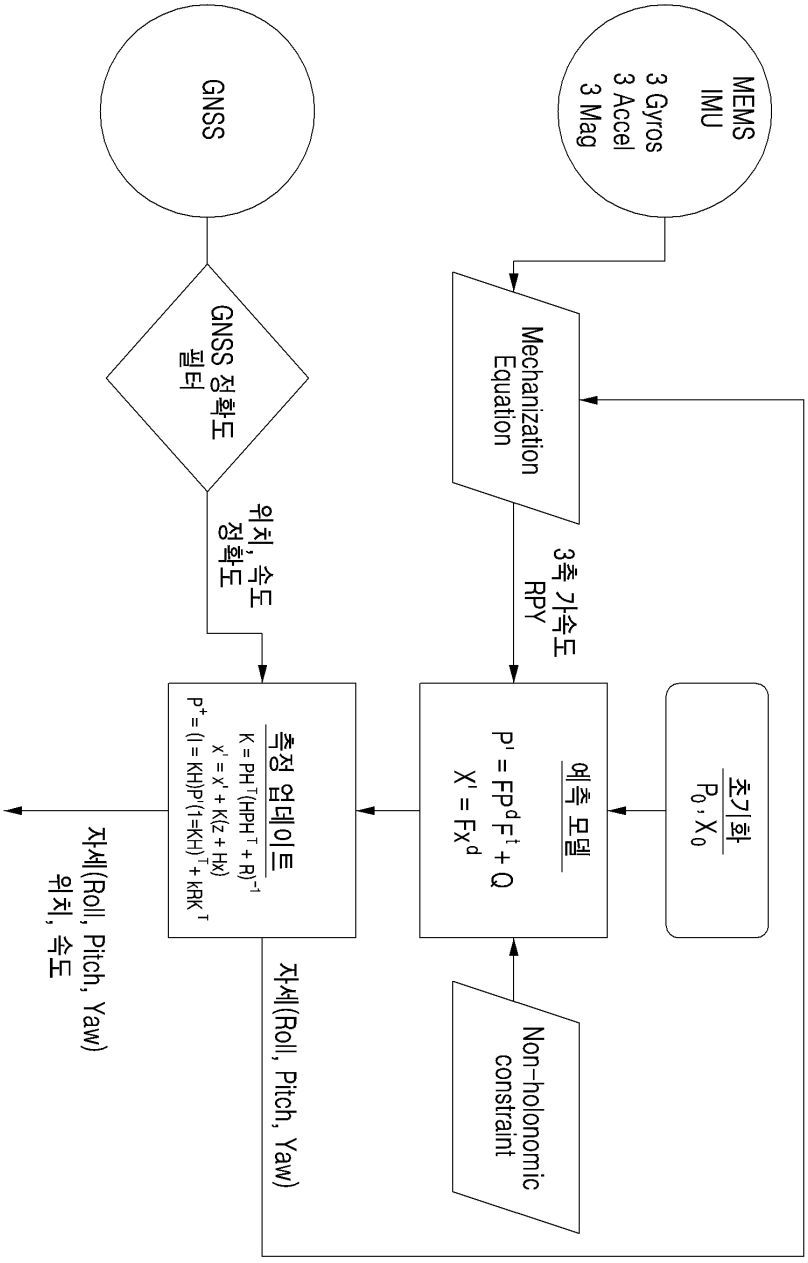
도면14



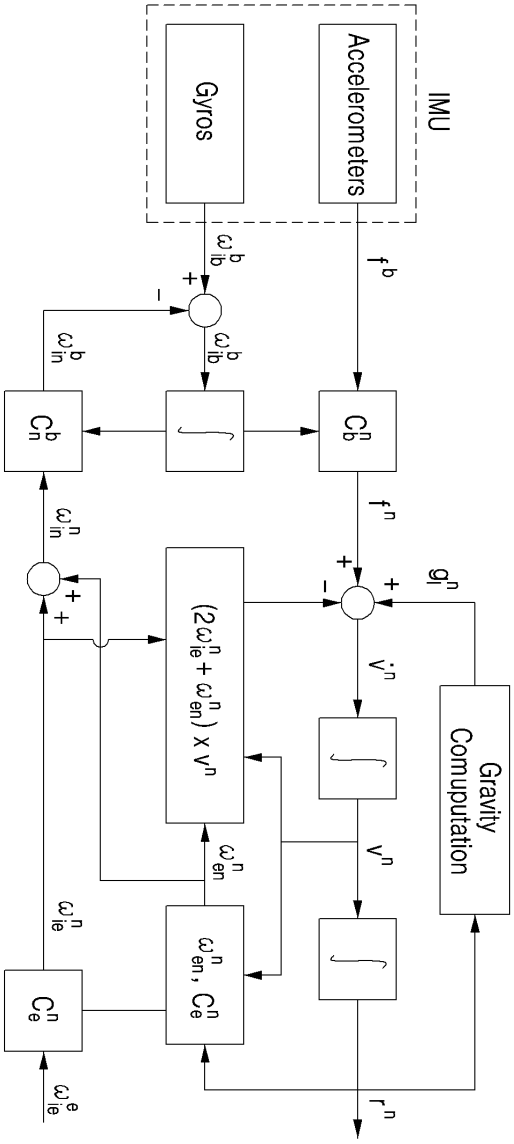
도면15


MEMS
IMU
3 Gyros
3 Accel
3 Mag
3축 각속도(rad/s)
3축 가속도(m/s²)
신호 분리
Low Pass Filter
High Pass Filter
Low Pass Filter
High Pass Filter
임계 필터 및 Wavelet Reconstruction
3축 가속도(m/s²)
3축 각속도(rad/s)

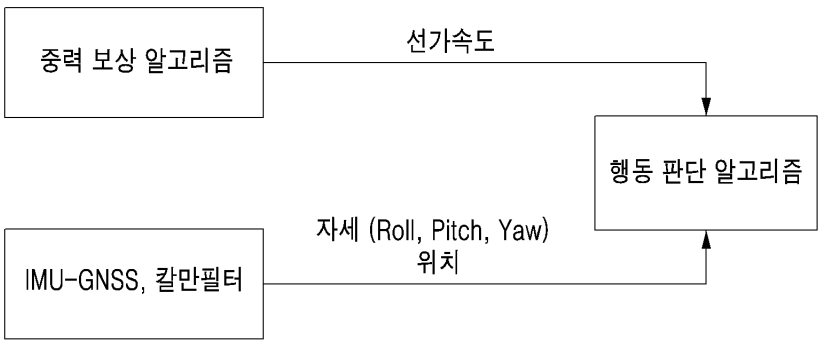
도면16


GNSS
MEMS
IMU
3 Gyros
3 Accel
3 Mag
GNSS 정확도
필터
Mechanization
Equation
위치, 속도
정확도
3축 가속도
RPY
초기화
P_0, X_0
예측 모델
$P' = FP^dF^t + Q$
$X' = Fx^d$
측정 업데이트
$K = PH^T(HPH^T + R)^{-1}$
$X' = X' + K(z + Hx)$
$P^+ = (I = KH)P'(1=KH)^T + KRK^T$
Non-holonomic
constraint
자세(Roll, Pitch, Yaw)
위치, 속도
자세(Roll, Pitch, Yaw)

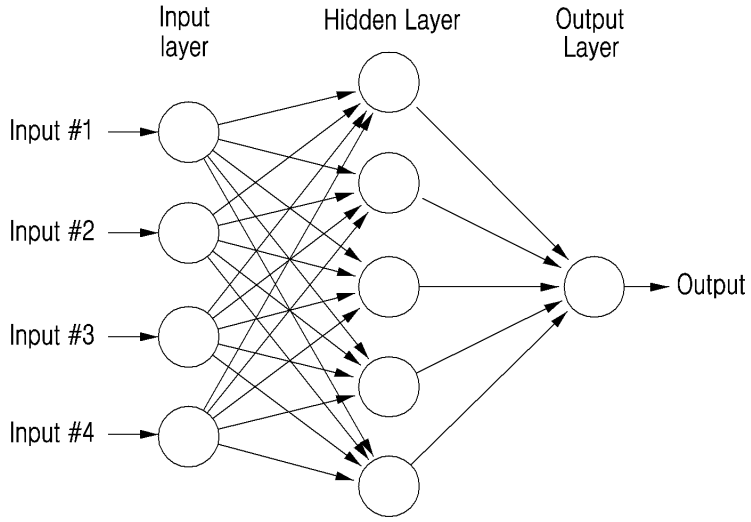
도면17


IMU
Gyros
Accelerometers
Gravity Comupation

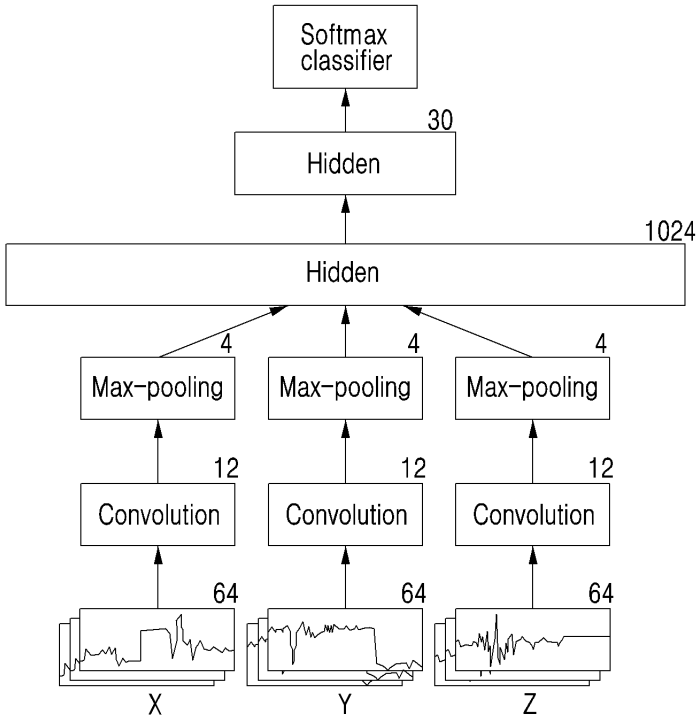
도면18


중력 보상 알고리즘
선가속도
행동 판단 알고리즘
자세 (Roll, Pitch, Yaw)
위치
IMU-GNSS, 칼만필터

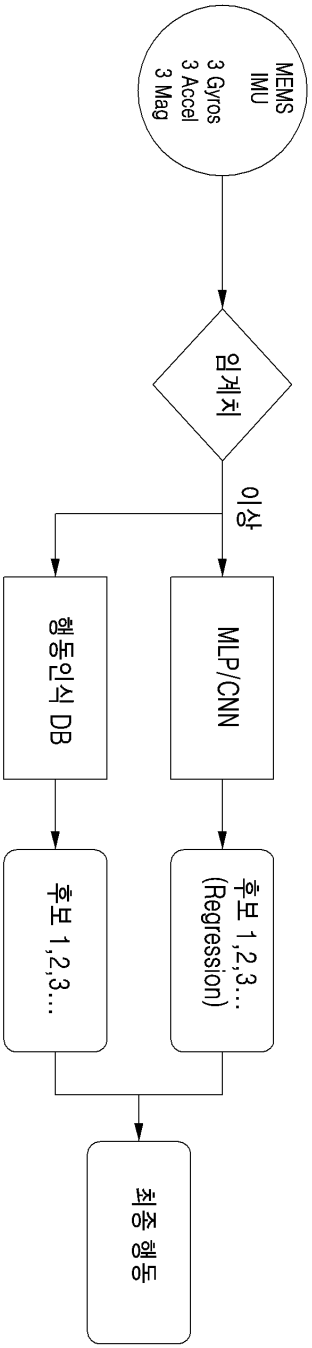
도면19


Input layer
Hidden Layer
Output Layer
Input #1
Input #2
Input #3
Input #4
Output

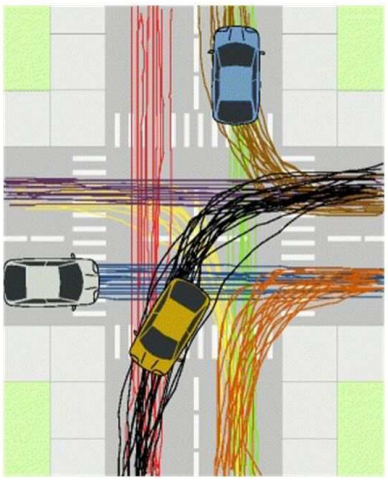
도면20


Softmax classifier
30
Hidden
1024
Hidden
4
4
4
Max-pooling
Max-pooling
Max-pooling
12
12
12
Convolution
Convolution
Convolution
64
64
64
X
Y
Z

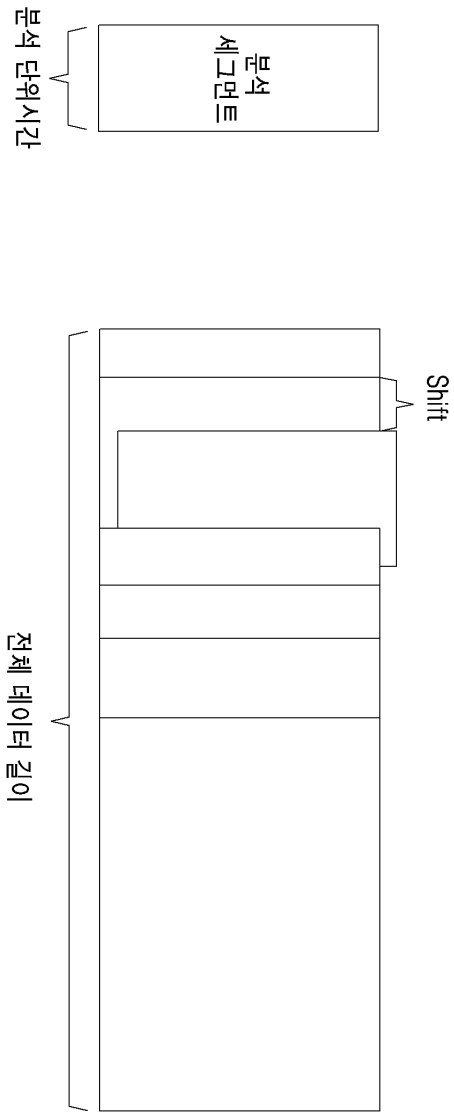
도면21


MEMS
IMU
3 Gyros
3 Accel
3 Mag
임계치
이상
행동인식 DB
MLP/CNN
후보 1,2,3...
후보 1,2,3...
(Regression)
최종 행동

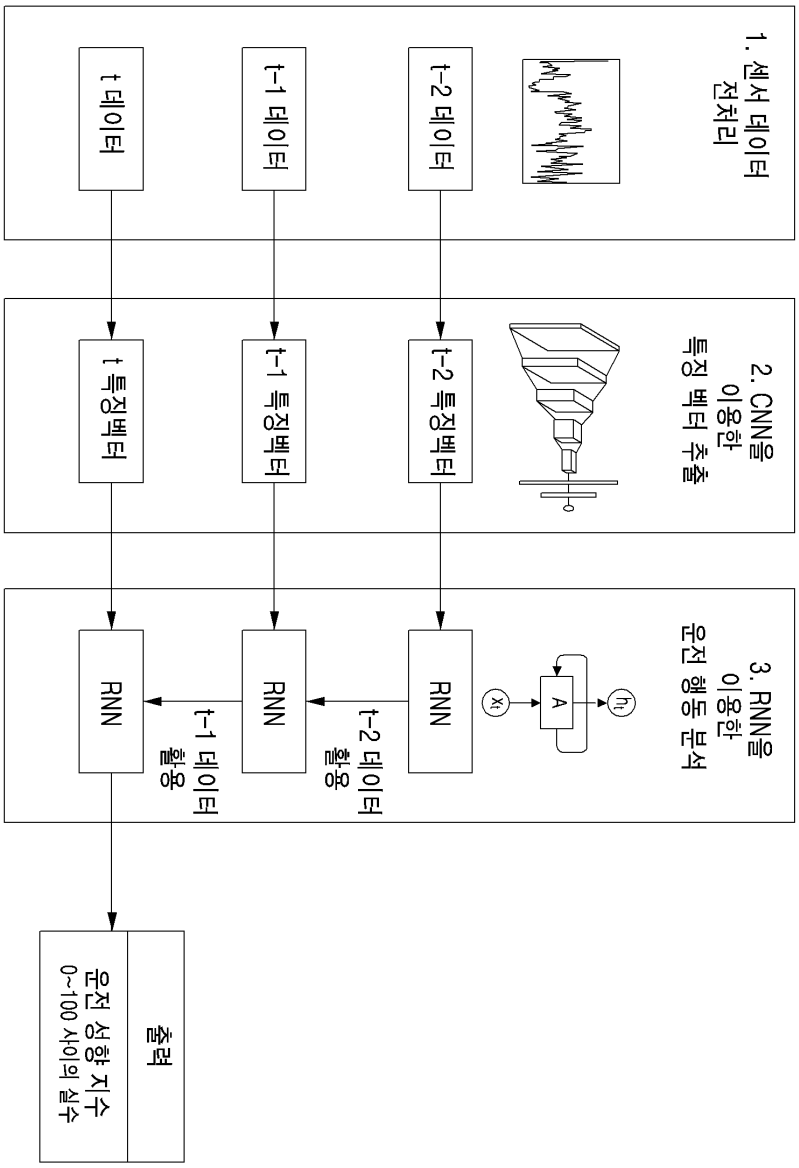
도면22



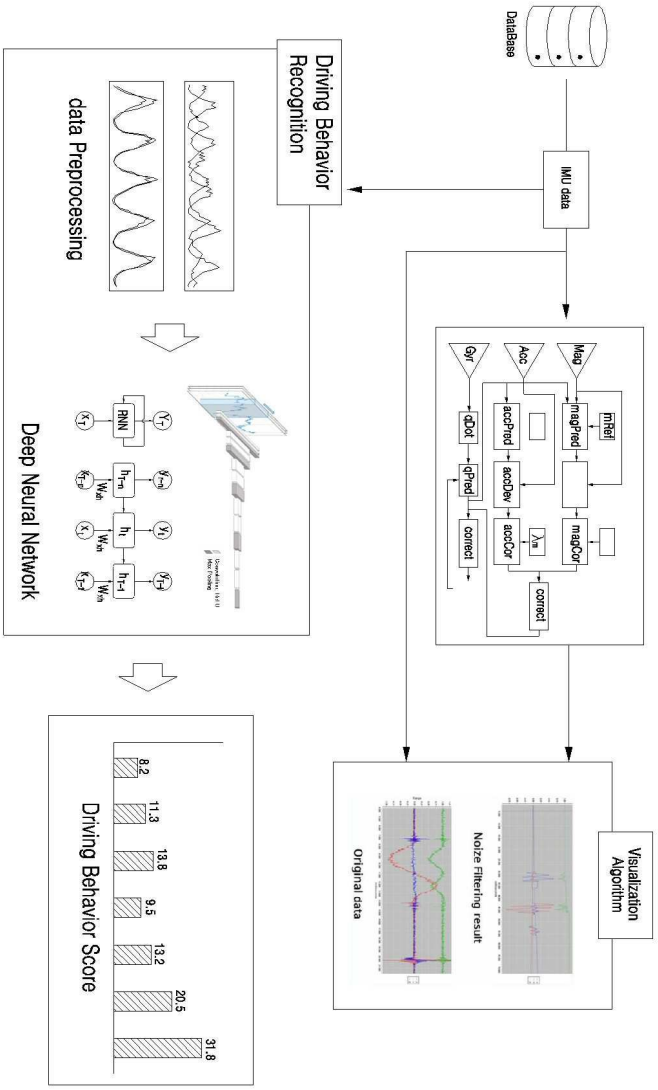
도면23



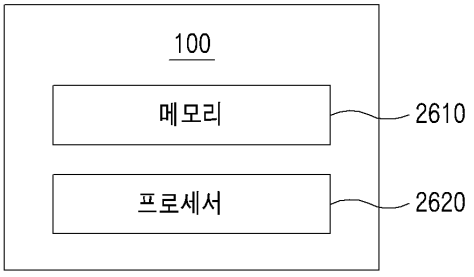
도면24


1. 센서 데이터 전처리
t 데이터
t-1 데이터
t-2 데이터
2. CNN을 이용한 특징 벡터 추출
t 특징벡터
t-1 특징벡터
t-2 특징벡터
3. RNN을 이용한 운전 행동 분석
RNN
RNN
RNN
t-1 데이터 활용
t-2 데이터 활용
A
출력
운전 성향 지수 0~100 사이의 실수

도면25


DataBase
IMU data
Driving Behavior Recognition
data Preprocessing
Deep Neural Network
RNN
Gyr
Acc
Mag
qDot
qPred
accPred
accDev
accCor
magPred
magCor
mRef
correct
Visualization Algorithm
Original data
Noize Filtering result
Driving Behavior Score
8.2
11.3
13.8
9.5
13.2
20.5
31.8

도면26


100
메모리
2610
프로세서
2620