

GPS 시스템 적용을 위한 맵 정보와 매칭 알고리즘의 구현

구진모*, 송의석*, 성영락**, 오하령**

*국민대학교 전자공학과

**국민대학교 전자정보통신공학부 교수

서울시 성북구 정릉동 861-1 국민대학교

e-mail:jin_dog@hanmail.net

Implementation of Map Information and Matching Algorithm for GPS System Application

Jin-Mo Koo*, Eu-Suk Song*, Young-Rak Sung**, Ha-Ryung Oh**

*Electronics engineering, Kookmin University

**Electronics engineering, Kookmin University, Professor

요 약

ATMEL사의 GPS(Global Positioning System) 칩셋을 이용하여 실제 GPS 응용분야에서 사용될 수 있는 GPS 시스템을 구현하였다. 또한, 맵 정보들을 플래쉬 메모리에 추가와 삭제를 가능토록 하기위한 데이터베이스 구조를 설계 하고 GPS의 오차와 실제 도로환경을 고려하여 매칭 알고리즘을 구현하였다. 본 논문에서 설계한 데이터베이스 구조에 42500개의 맵 정보가 정상적으로 플래쉬 메모리에 추가와 삭제되는 것을 확인하였으며, 실험을 통하여 구현한 매칭 알고리즘이 정상적으로 동작함을 확인하였다.

1. 서론

GPS(Global Positioning System)는 1970년 초 미국 국방부가 지구상에 있는 물체의 위치를 측정하기 위해 60억불을 들여 만든 군사 목적의 시스템이다. 그러나 오늘날에는 일부를 민간에게 개방하는 것을 전제로 미 의회에서 승인되어 민간에서도 사용되고 있다. GPS는 주로 비행기, 선박, 차량의 항법장치에 전자 지도(GIS)와 함께 GPS가 사용되고 있으며, 사람들이나 차량 등 이동체의 위치를 파악하는 데에도 사용된다. 또한 개인휴대용 GPS 수신기가 개발되어 미지 탐사나, 군 작전시 자기 위치파악에 이용되고 있다[1,2,3]. 사용자의 위치정보를 알 수 있는 GPS는 차량항법시스템, 항공기착륙 유도 시스템 등 많은 응용분야에서 발전해 왔다. 이러한 GPS를 차내에 설치하여 운전자가 초행길인 곳의 정보를 얻을 수 있다면 운전자의 안전운행에 도움이 될 것이다.

본 논문에서 구현한 GPS 시스템은 GPS를 차내에 설치하여 운전자에게 주변의 정보를 미리 알려주는

시스템을 구축하여 운전자의 안전운행에 도움이 되는 시스템을 구현하는 것이다. 이러한 시스템을 구현하기 위해 먼저 ATMEL사의 GPS 칩셋을 사용하여 GPS를 구현하였고, GPS 내에 맵 정보를 효율적인 데이터베이스 구조로 구현하여, 맵 정보와 차량의 위치를 매칭하여 실시간으로 운전자에게 정보를 제공할 수 있는 매칭 알고리즘을 구현 하였다.

2. 소프트웨어

2.1 개발환경

본 논문에서 구현하고자 하는 GPS 시스템을 구현하기 위해 필요한 개발환경은 첫째 GPS 시스템의 성능을 측정하기 위한 u-centerAE, 둘째 구현된 GPS 시스템이 원하는 값을 출력하는지 알 수 있는 모니터 프로그램, 셋째 GPS 시스템 내에 구현된 API를 이용하여 ARM7TDMI 코어에 프로그래밍 할 수 있는 컴파일러 ADS 1.2이다. 그림 1은 개발 환경을 그림으로 나타낸 것이다.

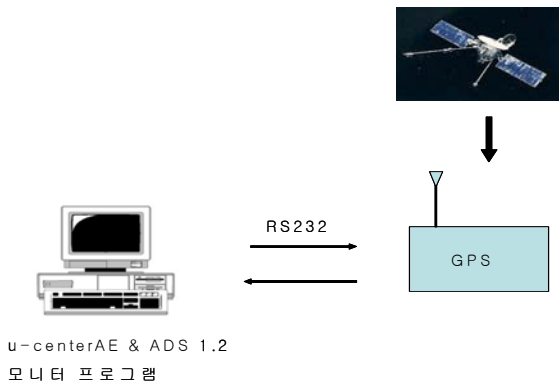


그림1 개발 환경

2.2 셀

본 논문에서는 모든 맵 정보들을 현재 사용자의 위치와 1초마다 비교한다는 것은 무리가 있기 때문에 다음과 같은 방법을 사용하였다. 셀이라는 단위를 두어 현재 사용자의 위치가 포함된 셀을 안다면 그 셀안에 있는 맵 정보들만 사용자의 위치와 비교하면 된다. 셀의 크기는 다음과 같이 정하였다.

실제로 대한민국의 모든 곳의 맵 정보를 다 수집할 수 없기 때문에, 가로 세로 각각 16Km지역에서 64개의 맵 정보를 수집하고 수집한 맵 정보를 바탕으로 모두 42500개의 맵 정보를 만들었다. GPS시스템의 CPU는 실험결과 1초에 약 6000개의 맵 정보를 비교 검색할 수 있었다. 매칭 알고리즘을 구현하면서 한번에 최대 4개의 셀을 동시에 비교 검색해야 하는 경우가 생겼고, CPU에서 다른 기능을 수행할 수 있는 여유를 주기 위해 하나의 셀에 최대 1000개의 맵 정보를 포함하도록 하였으며, 셀에 맵 정보를 추가하는 경우가 생기므로 여유를 주기위해 셀당 400개의 맵 정보가 포함되도록 가로, 세로 각각 40Km로 셀의 크기를 정하였다.

2.3 맵 정보의 구조

맵 정보의 구조는 다음과 같다.

```
typedef struct
{
    float latitude; // 위도
    float gradient; // 경도
    short event; // 종류
    short valid; // valid/invalid
} DB_DATA;
```

하나의 맵 정보는 위치 정보인 위도, 경도를 가지고 있어야 하고 맵 정보의 종류 그리고 맵 정보가 유효한 것인지 아닌지의 판별이 가능하도록 설정 할 수

있어야 한다. 위도와 경도는 도,분,초의 포맷형식이 아닌 RMC형식인 도,분 형식으로 하여, 실제로 GPS 시스템에서 사용자의 위치에 대한 위도,경도 값과 맵 정보를 비교 하였을때, 위도와 경도의 포맷을 맞추는 작업이 없도록 하였다.

2.4 블록의 구조

맵 정보들을 추가, 삭제 하기위해 제안된 데이터베이스 구조는 블록이라는 단위를 두어 그 안에 맵 정보들을 저장한다. 블록의 구조는 그림 2와 같다.

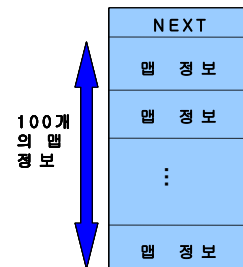


그림2 맵블록의 구조

하나의 블록에는 100개의 맵 정보들이 저장될 수 있도록 하였다. NEXT는 하나의 블록에 100개의 맵 정보들이 모두 저장되었을 경우 빈 블록을 찾아 빈 블록의 번호를 저장하고, 나머지 부분은 맵 정보들을 저장한다. 하나의 블록의 크기는 NEXT가 2Byte와 맵 정보 X 100인 1200Byte 총 1202Byte가 된다. 하나의 블록크기를 1202Byte로 하였을때 플래쉬 메모리에는 798개의 블록이 존재하게된다. 실제의 맵 정보들은 전 지역에 고르게 분포된 것이 아니고, 차량의 소통이 많은 곳에 집중되어 있기 때문에 하나의 블록에 저장되는 맵 정보의 수가 많게 되면 맵 정보가 거의 없는 셀에 할당된 블록에는 많은 공간이 남게 되어 플래쉬 메모리에 낭비되는 공간이 많이 생기게 된다. 그러나 본 논문에서는 가상의 맵 정보들이 어느정도 고르게 분포되어 있으므로, 임의로 하나의 블록안에 저장되는 맵 정보의 수를 100개로 하였다.

하나의 블록에는 100개의 맵 정보가 저장되고 블록에 저장되는 맵 정보가 100개가 넘을 시에 새로운 블록과 기존의 블록을 연결되도록 하였다. 맵 정보가 100개가 되어 하나의 블록이 모두 찼을 경우 빈 블록의 위치를 알아야 하기 때문에 빈 블록들인 Free블록들의 시작점을 가리키고 있는 Free블록 포인터를 두어 빈 블록을 할당시킨 뒤 Free블록 포인터를

한 블록의 크기만큼 이동시켜 Free블록들의 시작점을 가리키도록 하였다. Free블록은 하나의 셀에 하나씩 블록을 할당하기 때문에 셀에 할당된 블록을 제외한 698개가 존재하게 된다. 각 셀별로 하나의 블록씩 총 100개의 블록에 맵 정보들을 저장할 수 있게 하였으며 나머지 698개의 블록은 Free블록으로 두어 셀당 할당된 블록이 가득 차게 되면 Free블록에서 빈 블록을 찾아 셀에 할당하도록 하였다.

3. 매칭 알고리즘

본 논문에서 구현하려는 GPS시스템은 실시간으로 현재 사용자의 위치 정보가 어느 셀에 속하는지를 판별하고 그 셀 안의 맵 정보들과 비교한다. 그러나 사용자의 위치정보와 사용자의 위치정보가 속한 셀의 맵 정보만 비교하게 된다면 다음과 같은 문제점이 생기게 된다. 맵 정보가 근접 셀과의 거리가 GPS의 오차이내인 곳에 존재하고, 사용자의 위치가 근접 셀과 GPS의 오차이내에 위치하였을 경우 오차로 인해 사용자가 속한 셀이 바뀌게 되는 일이 발생할 수도 있다. 이러한 문제점은 다음과 같이 해결 하였다. 현재 사용자의 위치와 근접 셀의 거리가 GPS오차 범위 내라면, 근접 셀의 맵 정보들도 사용자의 위치와 비교하게 하였다. 사용자의 위치에 따라 그림3처럼 구분하였다

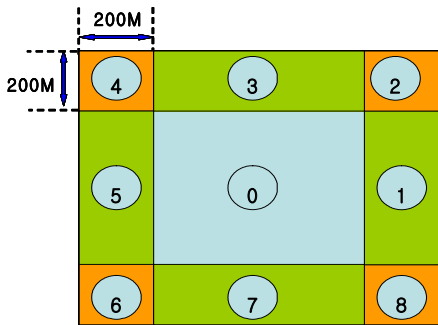


그림3 매칭 알고리즘의 보기

사용자 위치는 셀 안에서의 위치에 따라 0~8번으로 분류 된다. 사용자의 위치가 0번인 경우는 사용자가 속한 셀의 맵 정보만 비교 한다. 1번, 3번, 5번, 7번의 경우는 자신이 속한 셀의 맵 정보와 이웃한 하나의 셀의 맵 정보를 비교하고, 2번, 4번, 6번, 8번의 위치일 경우 자신이 속한 셀과 이웃한 3개의 셀의 맵 정보까지 비교하여 총 4개의 셀의 맵 정보들을 비교하는 방법을 사용 하였다.

본 논문에서는 맵 정보의 검색 외에도 CPU에서 다

른 기능을 수행할 수 있는 여유를 주기 위해 하나의 셀당 1000개로 하였다. 맵 정보와 사용자의 위치가 일치하는가의 판별은 GPS오차를 감안 하여 맵 정보 주위 200M안으로 사용자의 위치가 들어왔을 경우 일치하는 것으로 하였다.

도로는 일방통행이 아닌 상행선과 하행선으로 대부분 이루어져있기 때문에 오차범위를 감안한 알고리즘을 쓰지 않는다면 현재 사용자가 진행하고 있는 방향의 반대편의 맵 정보도 사용자의 위치와 일치하는 것으로 인식하는 문제점이 생기게 된다. 이러한 문제점을 해결 하기 위해 각 맵 정보마다 예비 신호를 주는 맵 정보를 만들었다. 사용자는 맵 정보가 있는 곳을 가기 전에 예비 신호를 주는 맵 정보를 먼저 만나게 된다. 예비 맵 정보는 각 맵 정보의 종류에 따라 다른 종류를 가지게 되고, 매칭 알고리즘은 사용자가 예비 맵 정보를 만난 뒤 해당 맵 정보를 만나면 맵 정보와 사용자의 위치가 일치 한다고 인식한다.

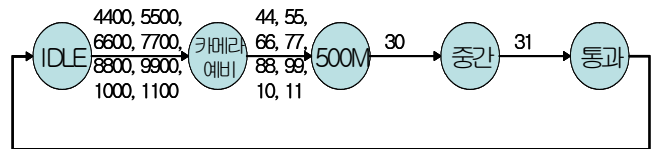


그림4 속도감지 카메라의 유한상태도

그림4는 속도감지 카메라에 대한 유한상태도를 나타내고 있다. 카메라의 맵 정보의 종류는 40Km, 50Km, 60Km, 70Km, 80Km, 90Km, 100Km, 110Km 로 모두 8개지만 카메라의 경우 하나의 카메라가 통과 되지 않은 상태에서 다른 종류의 카메라가 나오지 않으므로, 카메라의 예비 상태는 하나로 하였다. 카메라 예비 상태는 4400,5500,6600, 7700, 8800, 9900, 1000, 1100의 종류를 가진 카메라 예비 맵 정보 이후에 카메라 예비 상태로 되고, 500M 전 상태는 카메라 예비 맵 종류에 따라 맞는 맵 정보 이후에 된다. 중간 상태와 통과 상태도 중간구간임을 나타내는 맵 정보 30이후에 되고, 통과 상태는 중간상태에서 카메라를 통과했음을 알리는 맵 정보 31이후에 된다. 통과 상태가 되면 다시 IDLE상태로 가게 된다.

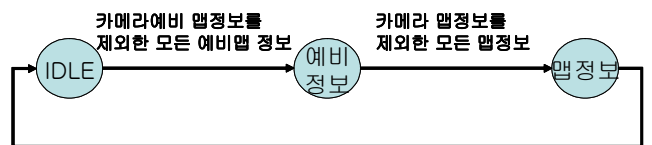


그림5 맵 정보의 유한상태도

그림5는 카메라 맵 정보를 제외한 나머지 맵 정보에

