

Car Navigation System에서 GPS와 추측항법을 결합한 위치오차의 최소화에 관한 연구

Minimizing Position Error in a Car Navigation System by
fusing GPS and Dead-Reckoning.

이 혁 중* 이 창 호* 김 광 익**
Lee, Hyuck Joong Lee, Chang Ho Kim, Kwang Ik

要 旨

CNS(Car Navigation System)는 국내외적으로 자율주행차량(Autonomous Land Vehicle)연구분야에서 주행계획 용도 보다는 일반차량에서 운전자 보조 시스템으로 더 많이 쓰이고 있다. 본 논문에서는 CNS에 한 연구분야인 Global Path Planning의 새로운 위치추적 알고리즘 결과를 개발하였다. 일본과 같은 경우에는 벌써 이러한 CNS의 연구가 활발히 진행되어서 현재는 상품화 되어 연간 40~50만개 정도가 팔리고 있고 미국및 유럽각국(EC)에서도 CNS의 개발및 상품화가 활발히 진행되고 있다.[1][14] 우리 나라의 경우엔 아직 주행중인 차량의 현재위치를 구하고 이를 Digital Map상에 실시간으로 구현하는 제 1세대 CNS가 연구 중에 있으나 아직은 초기 단계에 있다. 이에 본 논문은 현재까지 우리 나라에서 연구중인 인공위성과 범지구적 위치결정체계(Global Positioning System)만을 이용한 차량 위치 추정 방법에 추측항법체계(Dead Reckoning)라는 다른 방법을 결합하여 기존의 차량 위치 추정 오차를 줄여주는 새로운 알고리즘을 제시하고 실제차량에서의 실험결과를 보인다.

ABSTRACT

The CNS(Car Navigation System) is used more generally in driver aid system than ALV(Auto nomous Land Vehicle) research area. In this paper we developed a new position tracking algorithm for the Global Path Planning in the CNS. In japan, CNS is already well developed and, thesedays they sell CNS products about 400~500 thousands per year, and USA and European Communications(EC), too. In Korea, studies of the first generation CNS, which finds current location of a navigating vehicle and displays its location in a Digital-Map with real-time are progressing but still in the beginning step. Therefore a new position tracking algorithm is presented, which reduces vehicle position error dramatically by fusing GPS and dead-reckoning sensors. And the validity of our algorithm is demonstrated by the experimental results with the real car.

* 포항공과 대학교 정보통신 대학원 정보통신학과 석사과정

** 포항공과 대학교 수학과 교수/정보통신 연구소 지능 주행연구실 실장

1. 서론

차량의 정확한 현재위치를 GPS만을 가지고 얻는 것은 GPS가 받는 위성 신호 오차 및 CNS주위 환경의 영향으로 그 위치 오차가 상당히 크다(약 60~100미터). 기존의 주행 방법(piloting, celestial 등)들에 비해서 GPS를 이용한 CNS는 약 20,200km상의 고도 위에 있는 위성에서 받는 위치 신호 덕분에 기존 방법의 가장 큰 문제점이었던 누적오차를 줄이기는 하였지만, 위성에서 위치 신호를 받지 못하는 지역(예를 들면 터널 내부, 높은 빌딩들 사이, 나무가 너무 울창한 숲 등)이 생기게 된다.[8][17] 또, GPS위성의 HDOP(Horizontal Dilution Of Precision)나 주기적으로 바뀌는 위성의 궤도 때문에 위치를 연산하는데 문제점이 발생하게 된다. 그리고, GPS위성이 미국의 '위성이기 때문에 전에 있었던 "쿠웨이트-이라크戰" 같은 경우 임의로 그 위성의 위치를 전술상의 이유 때문에 바꾸는 경우도 생긴다. 이러한 문제점을 고치려고, 본 논문에서는 Dead Reckoning이라는 방법을 GPS와 함께 결합해서 쓰게 되었다. 즉, 위성에서의 신호를 수신 중에 있을 때 GPS 위성에서 송신되는 신호의 상태인 HDOP가 너무 높은 경우나, 수신되는 위성의 수가 3개 미만일 때에 GPS의 위치 오차를 최소화하기 위하여는 Dead Reckoning이라는 방법을 선택해야 한다. 본 알고리즘은 일단 차량의 출발점과 목적지를 선택하여 최단 경로를 찾게끔 한 뒤에 현재의 차량 위치를 찾아내게끔 하였다. 알고리즘의 정확도를 실험하기 위하여 포항공과 대학교내에서 차량에 Trimble Navigation사의 Svec Six GPS수신기, Murata사의 Gyro, 그리고 차량의 counter를 사용하여 실험하였다.

2. 본 문

2.1. 주행 경로 관리 시스템

자율 주행 차량이 어디로 가야하는지 어느 방향에서

회전을 해야하는지를 결정하는 것은 Dijkstra's Algorithm[3]을 이용하여 알게 되는데, 우선 출발지와 도착지를 선택하여 여러 갈래의 길 중 각 교차로마다 위치정보와 회전정보(좌회전, 우회전, 직진)를 주어서 교차로지역에 접근했을 때 실제 차량의 주행 알고리즘인 Vision이나 Neural Net module에게 적절한 방향을 알려주게 되어있다. 즉, 주행 알고리즘인 Neural Net이나 Vision Based Algorithm은 도로의 양쪽 선만을 보고 가는데 교차로 지역에 갔을 때 CNS에서 회전 정보를 주행알고리즘에 주면 CCD카메라를 이용하여 회전할 위치를 찾게 된다. 한편 여기서 Dijkstra's Algorithm은 제약조건인 노드와 노드 사이의 거리를 사용하게 된다. Dijkstra's Algorithm을 설명하면 아래와 같다.

- ① 먼저 초기 점에서 다음으로 갈 수 있는 node와 그 node사이의 거리를 구한다.
- ② 각각의 node에서 또 다음으로 연결된 node사이의 거리를 구하여 ①에서 구한 node와 연결, 그 거리를 알아낸다.
- ③ 이렇게 계속 loop를 돌다가 목적지까지 더 이상 node가 연결이 안되면 그때까지 연결된 node를 제거한다.
- ④ 제거 안되고 남아 있던 node는 계속 looping을 돌려 최종 목적지까지 연결되어 나온 경로 중 가장 거리가 짧은 길을 골라 최종적으로 최단 경로를 설정한다.

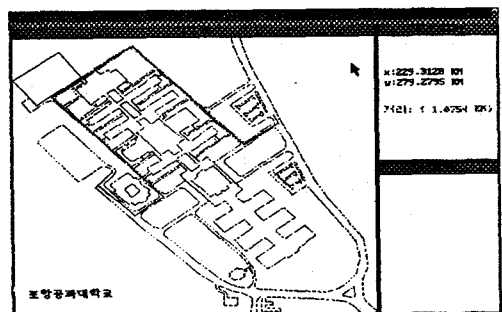


그림 1. Dijkstra 알고리즘을 이용하여 구현된 최단경로

2.2. WGS84좌표계의 변환과 Digital Map 제작.

GPS(NAVSTAR)위성은 미 국방성에서 만든 WGS 84라는 좌표를 이용한다. WGS84는 미 국방성 (Department Of Defense)에서 전 세계에 대하여 하나의 통일된 좌표계를 이용하게 하기 위하여 제작한 지심좌표를 기준으로 하는 준거회전타원체이다. WGS84는 지심좌표를 기준으로한 회전 타원체에 투영한 회전타원체에 투영한 경위도 좌표이기 때문에 우리나라에서 제작된 TM좌표를 사용하기 위해서는 WGS 84 좌표에서 동경원점을 기준(우리 나라 포함)으로 한 Bessel 경위도 좌표로 변환한 다음 우리 나라의 TM 좌표계로 변환해야 한다.[5][7]

본 논문에서는 WGS84를 미국의 DMA (Defence Mapping Agency)에서 제공되어지는 MRE좌표변환 방법으로써 구해진 Df, Di를 WGS84에서 구한 위도, 경도에 대해서 한국을 포함한 Tokyo Datum으로 나타내게끔 위도, 경도를 구한다. 이렇게 나온 위도, 경도에 우리가 이용하고 있는 지도가 xy직교 좌표계로 표시되어 있기 때문에 정확한 x,y좌표를 얻기 위해서 아래와 같은 변환공식을 이용하여 위도 경도를 xy직교 좌표계로 바꾸었다.

$$qb = latitude * 0.017453292$$

$$ql = longitude * 0.017453292$$

$$lat = 38 * 0.017453292$$

$$rc = \frac{6377397.155}{\sqrt{1 - 0.006674372231315 \times \sin(qb) \times \sin(qb)}}$$

$$etta = \frac{0.0816968312 * \cos(qb)}{\sqrt{1 - 0.006674392231315}} \times \frac{0.0816968312 * \cos(qb)}{\sqrt{1 - 0.006674392231315}}$$

$$x_0 = 6377397.15500 * (1 - 0.006674372231315) * (1.005037306048555 * lat - 0.5 * 0.0050478492403 * \sin(2 * lat) + 0.25 * 0.000010563786831 * \sin(4 * lat) - 1 / 6 * 0.000000020633321650 * \sin(6 * lat) + 0.125 * 0.00000000038853603630 * \sin(8 * lat) - 0.1 * 0.0000000000070028597050 * \sin(10 * lat))$$

$$b = 6377397.15500 * (1 - 0.006674372231315) * \{ (1.005037306048555 * qb - 0.5 * 0.0050478492403 * \sin(2 * qb) + 0.25 * 0.000010563786831 * \sin(4 * qb) - 1/6 * 0.000000020633321650 * \sin(6 * qb) + 0.125 * 0.00000000038853603630 * \sin(8 * qb) - 0.1 * 0.0000000000070028597050 * \sin(10 * qb) \}$$

$$x = b + rc * \sin(qb) * (ql - (129 + 10.405 / 3600) * 0.017453292) / 2 * \cos(qb) + (ql - (129 + 10.405 / 3600) * 0.017453292) / 24 * \cos(qb)^3 * (5 - (\tan(qb))^6) * (5 - (\tan(qb))^2 + 9 * etta + 4 * etta^2 + (ql - (129 + 10.405 / 3600) * 0.017453292) / 720 * (\cos(qb))^5 * (61 - 58 * (\tan(qb))^4 + etta * (270 - 330 * (\tan(qb))^2))$$

$$y = rc * \{ (ql - (129 + 10.405 / 3600) * 0.017453292) * \cos(qb) + (ql - (129 + 10.405 / 3600) * 0.017453292) / 6 * (\cos(qb))^3 * (1 - (\tan(qb))^2 + etta) + (ql - (129 + 10.405 / 3600) * 0.017453292) / 120 * (\cos(qb))^5 * (5 - 18 * (\tan(qb))^2 + (\tan(qb))^4) + etta * (14 - 58 * (\tan(qb))^2) \}$$

$$BX = (y + 200000) / 1000$$

$$BY = (x - x_0 + 500000) / 1000$$

latitude, longitude는 MRE방식에 의해서 나온 좌표이고, BX, BY는 우리가 구하고자 하는 XY직교좌표이다.

그리고, 본 논문에서 사용한 지도데이터는 1990년 12월에 포항제철에서 만들어진 1/5000비율의 포항시 지역 단지중 포항공과대학교 지역만을 Autocad로 digitizing하여 사용하였다. 여기에 최단경로를 찾기위

해서 도로의 정보만을 따로 넣은 간단한 데이터 베이스를 만들었다.(그림 1. 참조)[10]

2.3. 차량 위치 결정 시스템

차량의 위치를 결정하는 방법중 가장 많이 쓰이고 있는 것은 GPS와 Dead_Reckoning이다.[2][9][14] Dead_Reckoning은 방향 각의 vector 변화량과 주행한 거리를 연속적으로 찾아서 현재의 위치를 추정하는 방법으로 일명 “추측항법장치”라고도 한다. 이 방법의 장점은 GPS에 비해서 그 장비를 만드는 비용이 저렴하다는 것이고 또, 단거리에서는 상당히 정확한 위치를 찾을 수 있다는 점이다. 그에 반해 단점은 위치오차가 계속 누적된다는데 있다. 반대로 GPS수신기의 경우에 있어서는 장점이 오차가 축적이 되지 않는 것이고 단점은 위성에서 보내오는 위치신호의 상태에 따라 부정확한 데이터가 들어올 수 있다는 점이다. GPS 위성은 두 가지의 형태의 신호를 보낸다. 그 하나는 주파수를 10.23Mhz를 가진 군사용 코드라고 하는 P-code인데 이 코드는 3~5미터의 오차가 발생한다. 다른 하나는 1.023 Mhz의 주파수를 가지는 C/A code이고 대략 60~100미터의 오차가 난다. 일반적으로 P-code는 아무나 사용할 수 없도록 미국 국방성에서 암호를 걸어 놓고 있다. 그리고 C/A code에 대해서는 SA(Selective Availability)를 이용하여 일부러 위치 오차를 섞어넣고 있는 실정이다. 그래서 수신 받은 신호(C/A code)는 필연적으로 propagation delay가 있고 건물사이나 터널안, 그리고 숲속 같은 곳은 신호의 제약을 받아서, GPS만 가지고 정확한 위치를 찾아 내기에는 문제점이 많다. [4][6][8][16][17]

2.3.1. GPS수신기의 수신원리

GPS 수신기가 그 위치를 알아내는 기본적인 좌표계산의 원리는 삼각측량법과 동일하다. 즉 위치를 측정하려는 임의의 지점에서 위치를 알고 있는 3개의 GPS위성 지점까지 거리를 안다면, 그 지점의 공간좌표(X,Y,Z)를 구할 수 있게 된다. GPS수신기의 주된

역할은 위성과 수신기 간의 거리를 위성신호의 전달 시간 또는 주기(cycle)수를 측정하여 구하는 것이다. 수신기는 각 GPS위성에서 보내는 Data stream을 이용하여 현재 위성의 고도, 위도, 경도, 그리고 위성에서 수신기에 도달하는 시간을 알아내고 수신기 내의 program으로 수신기의 위치를 계산하게 된다.[17]

GPS수신기는 이와 같이 위성에서 위치를 알아내고 그 밖에 위성에서 보내는 신호를 받기 위해서 아래와 같이 세 가지 segment로 구성 되어있다.[6]

(1) Space Segment

18개의 위성과 예비위성으로 총24개의 위성으로 구성되어 위성으로부터 항상 전파가 송신되어지고 있으며 지상의 기지국으로 부터는 하루에 한번씩 신호(시간, 위치)를 보정받는다. Space Segment에서 하는 일은 수신기에서 필요로 하는 수신기의 위치, 속도, 그리고 위성에서 수신기에 도달하는 시간을 알아내기 위한 정확한 timing pulse, 위성의 위치 측정 계수를 보내는 것이다. 위치 측정표(ephemeris)는 16개의 상수로 구성되어 있는데 그 상수가 하는 일은 Navstar 수신기에게 어떤 위성이 현재 timing pulse를 보내는 지를 알게 해주는 것이다. 위성의 안테나는 2개의 부분으로 구성되어 있는데 이는 L밴드 향해 안테나(User Segment와의 수신용)와 S밴드 안테나(Control Segment와의 수신용)이다.

그밖에 위성에서 수신기까지 거리를 측정하는 방법은 아래와 같다. 위성은 미리 수학적으로 계산된 random화 된 pulse를 발생시키고 수신기는 위성에서 받은 전파를 program으로 계산하여 pulse가 얼마만큼의 시간이 delay이 되는가를 알게 된다. 그래서 그 matching되는 곳까지의 shift가 수신기까지 도착하는 시간이 된다. 여기에 전파가 전달되는 속도인 광속을 곱하면 수신기와 위성까지의 거리가 된다.

(2) Control Segment

Control Segment가 하는 일은 GPS 위성의 궤도를 추적하여 그 위성의 위치측정표(ephemeris) 상수, 그

리고 clock-bias 오차를 주기적으로 update해서 고쳐 주는 일이다. Navstar 위성은 언제나 위성의 현 위치를 기지국에 송신하기 때문에 기지국서 원하는 인공 위성의 위치 정보를 송신된 신호를 가지고 알아낼 수 있다. 송신국에는 GPS위성의 신호를 처리하기 위하여 5개의 무인 monitor station들이 위치하여 있는데 각 위치들은 Hawaii, Ascension Island, Diego Garcia, Kwajalein, 그리고 Colorado에 있는 Colora do Spring 이다. 기지국들은 위성에서 신호를 받아 보정된 신호를 다시 위성으로 보내는 역할을 하는데 이 중, Colorado Spring에 위치한 기지국은 이 역할 이외에 다른 기지국과 통신하는 매개체 역할을 하는 Master Control Station이다.

(3) User Segment

User Segment는 우리가 사용하는 수신기를 이야기 하는 것으로, 하는 일은 4개나 그 이상의 위성에서(동시나 혹은 순차적으로) 전파가 수신기까지 도달하는데 걸리는 시간과 위성의 위치신호를 이용하여 수신기의 정확한 위치와 속도, 그리고 위성과 수신기 사이의 거리를 알아내는 것이다. 일반적인 수신기의 구조는 안테나, 신호를 사용자가 쓰기 위한 데이터로 바꾸는 processor부분, 그리고 데이터를 display하는 부분의 3가지로 구성된다.

2.4 위치오차의 최소화를 위한 알고리즘

위에서 설명한 것처럼 GPS가 지나는 본질적인 오차 때문에 본 논문에서는 Dead_reckoning과 GPS를 결합해서 상호간의 단점을 보완할 수 있도록 하고 있다. 본 알고리즘은 회기분석곡선에 따라 GPS의 신호가 안좋을때는 Dead_reckoning의 방법을 쓰고, 그렇지 않으면 표 1]에 따라 Dead_Reckoning과 GPS를 결합해서 Dead_Reckoning의 오차를 줄이고 있다.

표1]에서 언급한 조건에 만족하게끔 GPS 데이터와 Dead_reckoning의 방법을 조합하여 직선 구간을 달릴

1	GPS의 HDOP data가 결정된 상수 값보다 작을경우. (본 program에서는 HDOP를 5로 결정했다.)
2	수신되는 위성의 수가 최소한 3개 이상일 경우.
3	GPS에 의해서 얻어진 두 지점간의 거리와 Counter에 의해서 얻어져 계산되어진 두지점간의 거리의 차이가 1.5m가 안될 경우.
4	GPS에 의해 처음으로 수신되어진 위치와 바로 전에 수신되어진 위치간의 거리에 GPS 에 의해서 얻어진 두 지점간의 거리를 더한 값에서 GPS에 의해 처음으로 수신되어진 곳에서 위와 현재위치간의 거리를 빼 값이 1.5m 가 안될 경우.
5	GPS에 의해 처음으로 수신되어진 위치와 현재 위치간의 거리와 본 알고리즘으로 구한 처음의 위치와 바로 전의 위치간의 거리의 차가 1.5m 가 안될 경우.

표 1] GPS 데이터를 선택하는 5가지 경우

경우에는 아래와 같은 회기곡선을 써서 차량의 위치를 구했다. 이를 만족하는 회기곡선의 식은 아래와 같다.

$$y = \alpha x + \beta,$$

$$\alpha = \frac{n \sum_{i=1}^n s_i t_i - (\sum_{i=1}^n s_i) (\sum_{i=1}^n t_i)}{n \sum_{i=1}^n s_i^2 - (\sum_{i=1}^n s_i)^2},$$

$$\beta = \frac{\sum_{i=1}^n t_i}{n} - \alpha \frac{\sum_{i=1}^n s_i}{n}$$

차량의 방향 D는 아래 식에 의해서 구할 수 있다.

$$D = \tan^{-1} \left(\frac{\delta y}{\delta x} \right),$$

$\delta x = \text{direction}_x - \text{starting point } x,$

$\delta y = \text{direction}_y - \text{starting point } y.$

이렇게 나온 데이터에서 최종적으로는 차량의 위치를 아래와 같이 구한다.

$$m = \sqrt{(gx - bx_0)^2 + (gy - by_0)^2},$$

$$l = \sqrt{(dx - bx_0)^2 + (dy - by_0)^2},$$

$$k = \frac{l}{m+l}.$$

bx, by : 최종적으로 구한 차량의 위치,

gx, gy : GPS에 의해 나온 x,y 위치,

dx, dy : Dead_reckoning에 의해 구한 x,y 위치.

최종적인 차량의 위치는 아래와 같은 식을 써서 구한다.

$$bx = k(gx - dx) + dx,$$

$$by = k(gy - dy) + dy.$$

이러한 모든 과정을 순서도로 설명하면 그림2 와 같다.

3. 실험 결과

위치측정 알고리즘을 실제 주행차량에서 GPS수신기만을 사용한 경우와 Dead_Reckoning만을 사용한 경우, 그리고 2.4절에서 설명한 새로운 알고리즘의 결과는 아래의 그림3, 그림4, 그림 5와 같다.

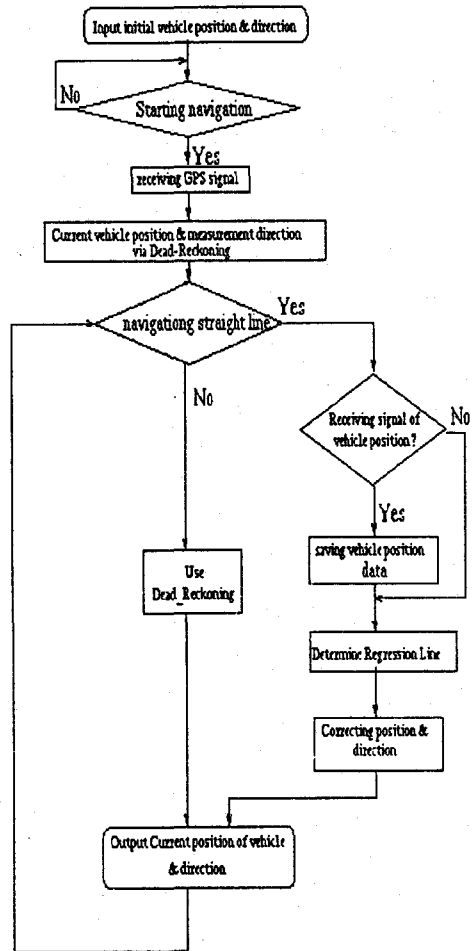


그림 2. 알고리즘 순서도

그림 3에서 GPS 수신기만을 써서 추적된 경로를 보면 출발지에서 도로의 왼쪽으로 벗어나는 것을 알 수 있다. 이것은 GPS의 단점인 좌우에 높은 건물이나 울타리, 1차 서퉁에서 언급한 바와 같이 전파의 교란을 받기 때문에 정확한 삼각측량이 불가능하여 출발점 계산에 오차가 생겼기 때문이다.

그림 4는 Dead_reckoning만을 이용했을 경우인데 이 때는 그림4에서 원으로 표시된 부분에서 경로추적

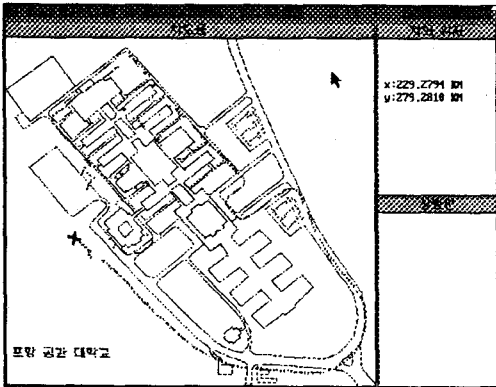


그림 3. GPS수신기만에 의한 주행경로

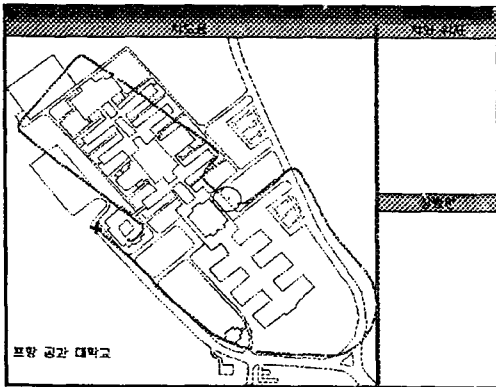


그림 4. Dead_reckoning만의 데이터에 의한 주행경로

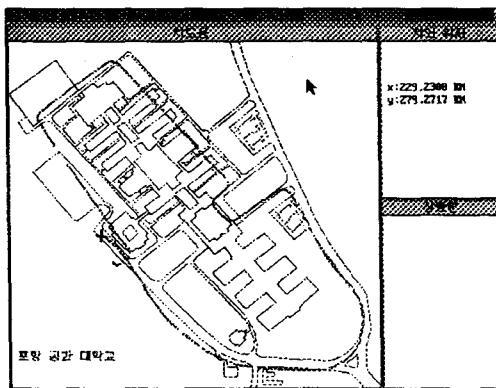


그림 5. 본 알고리즘에 의한 주행경로

연결이 안되어있다. 이러한 경우는 Dead Reckoning 데이터만을 사용할 때에 때때로 발생되는데 그 이유는 차량의 흔들림(위아래로)에 의해서 부정확한 Gyro 값 및 counter값을 읽게 되기 때문이다. 연결이 안된 위치(원으로 표시된 부분) 이후의 주행 추적경로는 왼쪽으로 돌아가 있는데 (그림4) 이 추적경로는 오른쪽으로 조금 돌리면 원래의 지도와 도로와 주행경로가 잘 맞게된다. 물론 시작점과 끝점은 그림3과 같이 X라고 표시된 부분에서 시작하여 다시 그 주위까지 돌아오는 경로이다.

그림5는 2.4절에서 설명한 알고리즘을 쓴 결과로 전체적인 데이터가 안정적이고 정확하다.

4. 결론 및 앞으로의 연구방향

실험환경에 대해서 언급하면 포항 공과 대학교의 도로가 좁고 높은 건물이 많아 일반적인 경우보다 열악한 조건이어서 GPS데이터가 안 좋게 수신된다. 그림 5에서 오차가 많이 나는 것처럼 보이는 이유는 보통의 경우 경로추적 데이터는 한 화면을 최소 2km x 2km정도로 사용하고, 차량의(수신기의) 크기가 대략 반경 30m 정도로 화면에 표시되기 때문에 대부분의 오차는 눈으로 보이지 않게 된다. 그러나 본 논문의 실험 데이터는 0.6km x 0.6 km의 화면에서 차량의 크기는 반경 1.5m가 되기 때문에 절대적 오차는 작으나 가시적인 상대적 오차는 커 보이게 된다. 이외에 오차를 발생시키는 요인을 열거하면 우리나라 특성상 일반인이 정확한 지도를 얻는다는 것이 불가능하고 지도를 컴퓨터에 입력하는 작업에서 오차가 발생하기 때문이다. 본 논문에서 제시한 알고리즘으로 실험했을 때 최대 오차는 15m 였다. GPS수신기만으로 수신했을 경우의 절대오차는 60m에서 100m정도로 상대적인 위치 오차가 75%에서 85%로 줄어 들었다. 근본적인 문제점은 아직도 우리나라에서는 정확한 지도가 없기 때문에 빠른 시일 내에 항공사진과 함께 정확한 측량을 하여 컴퓨터에 입력된 지도자체의 오차를 줄였으면 한다. 그리고 차후의 연구 방향은 본 위치 추적 알

고리즘에 Map-matching 기술을 접합시킴으로써 더 많은 오차를 줄이는 것으로 현재 연구를 진행 중이다. [10][11][13][14][18]

5. 감사의 글

본 연구는 93년도 국방과학연구소의 연구비 지원(과제 번호 UD930037BD)에 의한 연구의 일부로서, 이런 결과가 나올수 있도록 지원해 주신 국방과학 연구소 여러분께 깊이 감사드립니다.

6. 참고문헌

1. Toruitoh, Atsushi Watanabe, and Hiroshi Kishi, "Development of a Voice Navigation System", 자동차 기술 Vol. 47, No. 8, 1993. pp.18-23.
2. Hiroaki Tsuji, Hiroyuki Maeda, Akihito Shibata, and Fuminori Morisue, "Evaluation of Location System Combining a GPS Receiver With Inertial Sensor" '91 Vehicle Navigation and Information Systems Conference Processings Vol. 2, 1991, pp.645-649.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Introduction to Algorithms, 1992, pp. 527-532.
4. R. J. Milliken, C. J. Zoller, "Principle of Operation of NAVSTAR and System Characteristics", Global Positioning System, Vol. 1, 1980, pp.3-14.
5. 박필호, 한인우, 이영창, 강준목, "GPS 실용화를 위한 우리 나라 측지계와 WGS-84의 좌표변환 (1)", 한국측지학회, 제 16권 제 1호, 1992, pp. 41-50.
6. Tom Logsdon, The Navstar Global Positioning System, VAN NOSTRAND REINHOLD, 1992
7. 장용구, 김기홍, 강인준, "Car-navigation을 위한 수치지도 활용에 관한 연구", 한국지형공간정보학회 집, 1994, pp37-41
8. Trimble Navigation, 1992, SVeeSix 6-channel GPS sensor Specification and User's Manual, Revision C.
9. Wei-Wen Kao, "Intergration of GPS and Dead-Reckoning Navigation Systems", SAE paper 912808, 1991
10. W. Dwayne DeBardelaben, "An Algorithm for Controlling a Vehicular Navigation System's Map Display", SAE paper 900472, 1990
11. Jiro Tanaka, Kazuo Hirano, Toshiyoki Itoh, Hiroaki Nobuta and Seiichi Tsunooda, "Navigation System with Map-Matching Method", SAE paper 900471, 1990
12. Richard J. Weiland, "Standards for Navigation Database: A Progress Report", SAE paper 912752, 1991
13. K. Ito, "Past, Present and Future of In-Vehicle Communication" SAE paper 921597, 1997
14. James L. Buxton, Stanley K. Honey, Wadym E. Suchowerskyj and Alfred Tempelhof, "The Travelpilot: A Second-Generation Automotive Navigation System", IEEE Transaction on Vehicular Technology, Vol.40, No.1, February 1991
15. Donald F. Cooke, "Digital Map Database for IVHS", SAE paper 92C046, 1992
16. Earl G. Blackwell, 1985, "Overview of Differential GPS Methods", Global Positioning System, Vol.III, The Institute of Navigation, pp89.-100
17. 김동현, "Vehicle Navigation System", '93GIS Workshop, 한국 GIS학회, pp347-370
18. Toru Hirata, Toshiaki Hara and Atul Kishore, "The Development of a New Multi-AV System Incorporating an On-Boaed Navigation Function", SAE paper 930455, 1993