

博士學位論文

하이브리드 기법을 이용한 LBS기반의 유사궤적 추적시스템



濟州大學校大學院

컴퓨터工學科

韓京福

2006年 6月

하이브리드 기법을 이용한 LBS기반의 유사궤적 추적시스템

指導教授 郭鎬榮

韓 京 福

이 論文을 工學 博士學位 論文으로 提出함



韓京福의 工學 博士學位 論文을 認准함

審 查 委 員 長 _____ 印

審 查 委 員 _____ 印

審 查 委 員 _____ 印

審 查 委 員 _____ 印

審 查 委 員 _____ 印

濟 州 大 學 校 大 學 院

2006年 6月

A LBS-Based Tracking System of Similar Trajectories Using Hybrid Method

Kyoung-Bok Han

(Supervised by professor Ho-Young Kwak)

A thesis submitted in partial fulfillment of the requirement for
the degree of Doctor of Computer Engineering

June 2006



This thesis has been examined and approved.

Thesis director, _____

Thesis director, _____

Thesis director, _____

Thesis director, _____

Thesis director, _____

June 2006

Department of Computer Engineering
Graduate School
Cheju National University

Contents

SUMMARY	iii
I. 서론	1
1. 연구 배경 및 목적	1
2. 연구 내용 및 논문 구성	5
II. 관련 연구	7
1. 시공간 이동객체	7
2. 주기적인 보고에 의한 차량 위치정보 표현	9
3. 차량 이력 데이터 관리 시스템	12
4. 위치검출 방법	14
5. 관련 연구 요약	18
III. 차량 이동객체 정의 및 제약 조건	20
1. 차량 이동객체 정의	20
2. 차량 이동객체의 거리 제약 조건	22
3. 차량 시간 제약 조건	23
4. 차량 이동객체 연산자	24
IV. 차량 유사궤적 및 이동 시퀀스 추출	26
1. 차량 유사궤적	26
2. 유사궤적 방향각 추출	29
3. 유사궤적 추출을 위한 전처리	34
4. 유사궤적 추출을 위한 차량 정보표현	40
5. 차량 이동 시퀀스 추출	46
6. 빈발 이동 패턴 마이닝	50
7. 유사궤적 데이터베이스 설계	53
8. 차량 위치정보 불확실성 처리	56

V.	분석 방법 및 성능 평가	64
	1. 분석 방법	64
	2. 분석 환경과 조건	65
	3. 분석 시스템과 도구	66
	4. 구현과 분석	69
VI.	결론	147
	참 고 문 헌	153
	국 문 초 록	157
	감 사 의 글	159



SUMMARY

A LBS-Based Tracking System of Similar Trajectories Using Hybrid Method

Kyoung-Bok, Han
Department of Computer Engineering
Graduate School of Cheju National University

In this study, the hybrid methods are suggested, which use the direction angle information to present running trajectory and track the past locations through a small amount of vehicle's location information.

In order to prove the effectiveness of the new technique suggested here, vehicle's location information are collected by running the vehicles moving objects under various conditions. Using the location informations and direction angle information collected with time intervals, the vehicle's location information is abstracted, compared and analyzed.

When the similar trajectory generated by the new techniques is compared with GPS Track Maker and Difference Image Techniques, the running trajectory almost identical to the current situation was able to be presented rather than ones by the time interval.

In addition, having analyzed the consistency of the running trajectory, it is shown the running trajectory is more consistent than

one by time intervals.

When the sizes of files generated by running in real time were compared with the file made by the suggestion above, it was quantitatively proved that it was able to save 80 percent of storage space. The record number saved in database showed same results.

The method of a proposition generated location information rather than the direction angle techniques . However, it showed running trajectory track closer to real time than the direction angle one in presenting the vehicle's running trajectory.

Finally, from the results of comparing the vehicle's running distances, the suggested techniques and hybrid method can express the distances almost identical to the real time running distances which proved the effectiveness.

List of Figures

Fig. 1	A Spatiotemporal moving point object.	8
Fig. 2	Moving region object.	9
Fig. 3	Location of time T_0	10
Fig. 4	Predicted location of time T_2	11
Fig. 5	Predicted location of time T_2	11
Fig. 6	Query processing on euclidean space.	12
Fig. 7	Min_dist application example using minimum distances critical.	23
Fig. 8	Vehicle moving paths.	27
Fig. 9	Vehicle similar trajectories.	28
Fig. 10	Dynamic base line.	30
Fig. 11	Fixing base line.	31
Fig. 12	Extraction algorithm for vehicle moving action information.	32
Fig. 13	Trajectory of vehicle moving object T.	33
Fig. 14	Direction angle abstraction function of vehicle moving object.	34
Fig. 15	GPS receiving format.	35
Fig. 16	Location coordinates received while a vehicle stops. ..	39
Fig. 17	Location change of moving object between t_1 and t_2	41
Fig. 18	Circle equation.	43
Fig. 19	Average speed.	44
Fig. 20	The speed to input steps.	45
Fig. 21	Attribute information of vehicle moving object running sections.	48
Fig. 22	Moving sequence abstraction algorithm.	50
Fig. 23	Candidate sequence creation example.	51
Fig. 24	Moving pattern mining algorithm VMP's whole composition.	53
Fig. 25	Past location estimation algorithm.	61
Fig. 26	Future location estimation algorithm.	62

Fig. 27	Vehicle running trajectory.	68
Fig. 28	GPS TrackMaker execution.	68
Fig. 29	GPS receiving error by vehicle's backward moving. ..	70
Fig. 30	Vehicle running trajectories.	71
Fig. 31	The first time interval running trajectory.	73
Fig. 32	The second time interval running trajectory.	74
Fig. 33	The third time interval running trajectory.	75
Fig. 34	The 4th time interval running trajectory.	76
Fig. 35	The 5th time interval running trajectory.	77
Fig. 36	The first similar trajectory.	78
Fig. 37	The second similar trajectory.	79
Fig. 38	The third similar trajectory.	80
Fig. 39	The 4th similar trajectory.	81
Fig. 40	The 5th similar trajectory.	81
Fig. 41	Hybrid trajectory.(Direction angle $10^\circ+$ Time interval 1min).	83
Fig. 42	Hybrid trajectory.(Direction angle $20^\circ+$ Time interval 1min).	84
Fig. 43	The first real time running trajectory and time interval running trajectory.	86
Fig. 44	The second real time running trajectory and time interval running trajectory.	87
Fig. 45	The third real time running trajectory and time interval running trajectory.	88
Fig. 46	The 4th real time running trajectory and time interval running trajectory.	89
Fig. 47	The 5th real time running trajectory and time interval running trajectory.	90
Fig. 48	The first real time running trajectory and similar trajectory.	91
Fig. 49	The second real time running trajectory and similar trajectory.	92
Fig. 50	The third real time running trajectory and similar trajectory.	93
Fig. 51	The 4th real time running trajectory and similar trajectory.	94
Fig. 52	The 5th real time running trajectory and similar trajectory.	94

Fig. 53	Real time vehicle running trajectory and hybrid trajectory.	96
Fig. 54	Real time vehicle running trajectory and hybrid trajectory.	97
Fig. 55	Trajectory comparison(Hybrid trajectory : direction angle $10^{\circ}+1\text{min}$).	99
Fig. 56	Trajectory comparison.(Hybrid trajectory : direction angle $20^{\circ}+1\text{min}$).	100
Fig. 57	Trajectory comparison.(Hybrid trajectory : direction angle $20^{\circ}+1\text{min}$).	101
Fig. 58	The first norm difference image abstraction.	103
Fig. 59	The first time interval difference image abstraction(1min).	103
Fig. 60	The first time interval difference image abstraction(2min).	103
Fig. 61	The first time interval difference image abstraction(3min).	104
Fig. 62	The first time interval difference image abstraction(4min).	104
Fig. 63	The first time interval difference image abstraction(5min).	104
Fig. 64	The first similar trajectory difference image abstraction(10°).	105
Fig. 65	The first similar trajectory difference image abstraction(20°).	105
Fig. 66	The first similar trajectory difference image abstraction(30°).	106
Fig. 67	The first hybrid trajectory difference image abstraction($10^{\circ}+1\text{min}$).	106
Fig. 68	The first hybrid trajectory difference image abstraction($20^{\circ}+1\text{min}$).	107
Fig. 69	The second time interval difference image abstraction (1min).	108
Fig. 70	The second time interval difference image abstraction (2min).	108
Fig. 71	The second time interval difference image abstraction (3min).	108
Fig. 72	The second time interval difference image abstraction (4min).	109
Fig. 73	The second time interval difference image abstraction (5min).	109
Fig. 74	The second similar trajectory difference image abstraction(10°).	109
Fig. 75	The second similar trajectory difference image abstraction(20°).	110
Fig. 76	The second similar trajectory difference image abstraction(30°).	110

Fig. 77	The second hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).	110
Fig. 78	The second hybrid trajectory difference image abstraction($20^\circ+1\text{min}$).	111
Fig. 79	The third time interval difference image abstraction(1min).	112
Fig. 80	The third time interval difference image abstraction(2min).	112
Fig. 81	The third time interval difference image abstraction(3min).	112
Fig. 82	The third time interval difference image abstraction(4min).	113
Fig. 83	The third time interval difference image abstraction(5min).	113
Fig. 84	The third similar trajectory difference image abstraction(10°).	113
Fig. 85	The third similar trajectory difference image abstraction(20°).	114
Fig. 86	The third similar trajectory difference image abstraction(30°).	114
Fig. 87	The third hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).	114
Fig. 88	The third merge trajectory difference image abstraction($20^\circ+1\text{min}$).	115
Fig. 89	The 4th time interval difference image abstraction(1min).	116
Fig. 90	The 4th time interval difference image abstraction(2min).	116
Fig. 91	The 4th time interval difference image abstraction(3min).	116
Fig. 92	The 4th time interval difference image abstraction(4min).	117
Fig. 93	The 4th time interval difference image abstraction(5min).	117
Fig. 94	The 4th similar trajectory difference image abstraction(10°).	117
Fig. 95	The 4th similar trajectory difference image abstraction(20°).	118
Fig. 96	The 4th similar trajectory difference image abstraction(30°).	118
Fig. 97	The 4th hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).	118
Fig. 98	The 4th hybrid trajectory difference image abstraction($20^\circ+1\text{min}$).	119
Fig. 99	The 5th time interval difference image abstraction(1min).	120
Fig. 100	The 5th time interval difference image abstraction(2min).	120
Fig. 101	The 5th time interval difference image abstraction(3min).	120
Fig. 102	The 5th time interval difference image abstraction(4min).	121
Fig. 103	The 5th time interval difference image abstraction(5min).	121

Fig. 104	The 5th similar trajectory difference image abstraction(10°).	121
Fig. 105	The 5th similar trajectory difference image abstraction(20°).	122
Fig. 106	The 5th similar trajectory difference image abstraction(30°).	122
Fig. 107	The 5th hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).	122
Fig. 108	The 5th hybrid trajectory difference image abstraction($20^\circ+1\text{min}$).	123
Fig. 109	Comparative graph of the number of black pixels.	124
Fig. 110	Comparative graph of the number of white pixels.	124
Fig. 111	Comparative graph of the file sizes between time Interval and similar trajectory.	126
Fig. 112	Comparative graph of the number of records between time Interval and similar trajectory.	127
Fig. 113	Time Interval consistency comparison.	129
Fig. 114	Similar trajectory consistency comparison.	130
Fig. 115	Hybrid trajectory consistency comparison.	130
Fig. 116	GPS location information change by vehicles running.	131
Fig. 117	Vehicle running distance.	133
Fig. 118	Percent of vehicle running distance.	133
Fig. 119	Vehicle running path.	134
Fig. 120	Starting area and running path point.	135
Fig. 121	Finishing area and running path point.	135
Fig. 122	Vehicle running path abstraction.	136
Fig. 123	Real time running trajectory and time interval running trajectory in straightaway.	137
Fig. 124	Real time running trajectory and similar trajectory in straightaway.	137
Fig. 125	Comparison of file sizes between time interval and similar trajectory in straightaway.	138
Fig. 126	Comparison of the number of records between time interval and similar trajectory in straightaway.	139
Fig. 127	Vehicle attribute table structure and search.	139
Fig. 128	Driver attribute table structure and search.	140
Fig. 129	Vehicle history data search result.	140
Fig. 130	Vehicle moving sequence table.	141
Fig. 131	Vehicle frequent sequence table.	141

Fig. 132 Comparative graph of similar trajectory file size 143
Fig. 133 Comparative graph of the number of similar trajectory record. 143
Fig. 134 Similarity between real time and similar trajectory of. 145
Fig. 135 Consistency of similar trajectory. 146



List of Tables

Table 1.	Query for moving point object.	8
Table 2.	Query for moving region object.	9
Table 3.	Positioning rate and distance error.	18
Table 4.	Vehicle location operator.	25
Table 5.	GPRMC field composition.	37
Table 6.	Location coordinate value received by GPS.	38
Table 7.	Moving sequence.	47
Table 8.	Vehicle attribute table.	54
Table 9.	Driver attribute table.	54
Table 10.	Vehicle history table.	55
Table 11.	Vehicle moving sequence table.	55
Table 12.	Frequent sequence table.	56
Table 13.	System and analysis tool for vehicle location information and similar trajectory abstraction.	69
Table 14.	Vehicle running data for vehicles location information abstraction.	70
Table 15.	Comparison of the number of pixels in the first vehicles running trajectories.	107
Table 16.	Comparison of the number of pixels in the second vehicles running trajectories.	111
Table 17.	Comparison of the number of pixels in the third vehicles running trajectories.	115
Table 18.	Comparison of the number of pixels in the 4th vehicles running trajectories.	119
Table 19.	Comparison of the number of pixels in the 5th vehicles running trajectories.	123
Table 20.	Quantitative comparison between time interval and similar trajectories(first).	125
Table 21.	Quantitative comparison between time interval and similar trajectories(second).	125
Table 22.	Quantitative comparison between time interval and similar trajectories(third).	125
Table 23.	Quantitative comparison between time interval and similar trajectories(4th).	126

Table 24.	Quantitative comparison between Time Interval and similar trajectories(5th).	126
Table 25.	Vehicle running distance(units : km).	132
Table 26	Percentage of vehicles running distance(units :%). ·	132
Table 27.	Quantitative comparison between time interval and similar trajectories.	138
Table 28.	The file size of similar trajectory by real time vehicle running(units : %).	142
Table 29.	The number of similar trajectory records by real time vehicle running(units : %).	143



I. 서론

1. 연구 배경 및 목적

최근 GPS(Global Positioning System)측위기술과 이동통신기술의 발달로 위치 기반 서비스(LBS : Location Based Service)의 요구가 증대되고 있으며, 이를 위해, 이동객체들의 위치정보를 효율적으로 검색, 저장하기 위한 이동객체 연구가 이루어지고 있다(배태욱 등 2004).

특히 이동객체 중 차량 이동객체에 대한 연구가 활발히 이루어지고 있으며 차량 위치추적 시스템(Vehicle Location Tracking System)과 지능형 교통시스템(ITS : Intelligent Transport System), 차량 관리 시스템(Vehicle Management System) 등 다양한 연구가 진행 중이다.

차량 위치추적 시스템과 지능형 교통시스템에서는 차량의 위치를 검출하는 측위기술을 포함하는 요소기술들을 필요로 한다. 또한 이동객체 위치 데이터에 대한 효율적인 저장 및 처리 방안을 포괄하는 이동객체 데이터베이스(MODB : Moving Object Databases)에 관한 연구도 필요하다. 이러한 이동객체에 대한 연구는 구체적으로 데이터 모델링, 질의 표현, 인덱스 구조, 불확실성 처리 방법 등이 있다. 뿐만 아니라 차량 위치 추적과 비즈니스에 관련된 응용 시스템들의 개발에 관한 연구가 초기 진행 단계에서 이루어지고 있다.

그리고 텔레매틱스(Telematics) 산업의 성장으로 물류 및 차량 관리, 응급 상황, 차량 도난과 같은 상황 발생시 차량의 현재 위치정보를 빠르게 서비스하기 위한 현재 및 미래 위치 색인에 대한 연구가 이루어지고 있다

(S. Saltenis 등 2000), (Y. Tao 등 2003), (S. Saltenis와 C. S. Jensen 2002), (C. M. Procopiuc 등 2002).

특히 차량 이동객체는 시간의 변화에 따라 그 위치가 연속적으로 변경되는 특성을 가지므로, 이를 효율적으로 저장 및 관리하기 위한 연구가 필요하게 되었다(안윤애 등 2002).

추적 대상이 되는 차량이 시간의 흐름에 따라 그 위치정보가 연속적으로 변경되는 특성을 가짐에도 불구하고 기존의 차량 위치 추적을 위해 개발된 많은 시스템들은 이동객체의 위치 데이터에 대하여 매우 제한된 범위만을 저장하였다. 즉, 개별적인 차량의 가장 최근 데이터만을 대상으로 서비스를 제공하였다(김진석 등 2004).

최근 들어 인터넷(Internet)을 기반으로 하는 관련 기술과 비즈니스 영역이 확장 및 발전함에 따라 이전에 개발된 시스템에 기능 및 성능을 추가하고 보완하는 필요성이 제기되어 왔으며, 이를 해결하기 위한 추가 연구들이 진행되어 왔다.

이동객체 위치추적을 위한 데이터는 갱신 간격이 매우 짧고 빈번하게 발생하는 일종의 시공간 데이터(Spatiotemporal Data)이며 방대한 크기를 지닌다. 이러한 이유로 지금까지 진행된 대부분의 연구에서는 관리의 효율과 성능의 관점에서 이동객체의 위치 데이터에 정보를 제한적으로 관리하고 있다. 즉, 기존의 차량추적 관련 연구들은 가장 최근에 수신된 이동객체 위치 데이터만을 관리하기 때문에, 데이터베이스에 저장되지 않은 과거 및 미래의 차량 위치 추정이 불가능하다. 그리고 효율적인 차량 위치정보를 추출하기 위한 방법도 제한적이다. 차량 현재 위치정보를 추출하기 위한 방법으로 시간간격 보고에 의한 차량 위치정보만을 추출하고 있는 실정이다.

차량 이동객체의 위치정보 보고 방법으로 S. Saltenis(2000)는 속도와

방향이 미리 설정한 범위를 벗어난 경우 서버로 보고가 이루어지도록 하였다. 예를 들어, 차량 이동객체가 마지막으로 보고한 속도와 5km 이상 차이가 날 경우는 클라이언트인 차량 이동객체가 서버로 현재 위치정보를 보고하도록 하였다. 이러한 조건은 실제 도로 환경에 적용시 차량의 속도와 방향의 잦은 변화로 인해 클라이언트인 차량 이동객체와 서버와의 무선 통신비용을 크게 증가시키는 문제가 있다. 또한 차량 이동객체의 수가 많아지면 서버의 병목현상(bottleneck)을 일으키는 문제가 발생한다.

적정한 통신비용을 유지하기 위해서 차량 이동객체의 보고를 특정 주기 즉, 일정 시간간격(Time Interval)마다 보고가 이루어지게 하는 방법이 있다. 일정한 시간간격마다 보고가 이루어지게 되면 차량 이동객체의 속도와 방향이 바뀌어도 서버로의 보고가 이루어지지 않는다. 그러나 마지막에 보고된 위치정보를 사용하여 미래의 차량 위치를 예측함으로써 보고 주기가 길어질 경우 예측 오차가 크게 발생할 수 있다. 즉, 일정한 시간간격으로 보고되는 경우 차량 이동객체의 속도와 방향이 변화에 따른 위치 예측의 불확실성을 반영할 수 없다.

기존 차량 및 이동객체 관리 시스템은 주기적인(Time Interval) 보고에 의한 차량 관리가 이루어지고 있으며, 보고 주기에 의한 여러 가지 문제점들이 발생하고 있다. 보고 주기가 짧으면 방대한 차량 이력 데이터가 발생하게 된다. 이는 클라이언트와 서버간의 통신이 빈번함에 따른 서버의 큰 부하를 발생시켜 시스템 효율을 저해하는 요인이 된다. 그리고 방대한 차량 이력 데이터를 저장하기 위해 대용량의 데이터베이스를 사용하기 때문에 시스템 구축비용이 증가하는 요인이 된다. 반대로 보고 주기가 길면 운행 중인 차량의 방향 전환과 속도 변화에 따른 차량 위치 예측 오차가 커지는 결과를 초래하게 된다.

최근 텔레매틱스 서비스 확대로 차량의 운행 궤적 정보가 데이터마닝

과 같은 응용분야에 필요로 하지만 기존 시스템들은 이동객체의 현재 위치정보만을 저장, 관리하기 때문에 이전에 운행한 궤적 정보를 추출하기 어렵다. 따라서 과거 위치에 대한 위치정보의 저장이 필요하다(양해정 등 2004).

또한 대부분의 연구에서 차량 이동객체의 위치정보는 GPS에서 제공하는 데이터를 가공하지 않고 시간에 따른 위치정보를 그대로 사용하였다.

이동객체의 과거 위치 분석을 위해 GPS 데이터를 그대로 이용할 경우 다음과 같은 문제점들이 발생할 수 있다.

첫째, 이동객체의 궤적 데이터는 시간에 따라 누적되므로, 데이터의 저장 비용은 시간이 증가함에 따라 선형적으로 증가한다. 따라서 누적된 위치정보 처리를 위한 비용 역시 선형적으로 증가한다.

둘째, GPS 데이터는 시간 단위가 증가할 때 마다 새로운 갱신이 발생한다. 따라서 차량 이동객체가 정지한 경우의 위치정보도 새로운 정보로 간주하며, 이를 시스템에 저장하게 된다.

위와 같이 차량 이동객체의 위치정보를 GPS데이터로 저장, 관리하는 경우 비효율적인 요인들이 존재함을 알 수 있다.

따라서 이러한 위치정보를 간결하게 표현하고 효율적으로 처리하는 방법에 대한 연구가 필요하다.

본 연구는 방향각과 시간 간격을 이용한 하이브리드 유사궤적 추적 시스템을 제안하였다.

반대한 차량 위치정보에서 차량 진행 방향을 결정하는 차량 방향각 정보를 이용하여 차량 위치정보를 추출하였으며, 추출된 차량 위치정보를 이용하여 차량 운행궤적을 생성할 수 있는 방법을 제시하였다.

본 연구는 다음과 같은 단계로 수행하였다.

첫 번째, 차량은 시간 변화에 따라 위치정보가 변경되는 시공간 이동객

체이므로 시공간 이동객체에 대한 정의, 종류, 표현 및 질의에 대하여 서술하였다. 또한 주기적인 보고에 의한 차량 이동객체의 문제점과 GPS 측위 기술 및 차량 이력 데이터 관리 시스템 종류에 대해서 서술하였다.

두 번째, 차량은 시간 변화에 따라 차량 이동객체의 위치정보, 방향, 거리 등의 정보를 갖는 객체이다. 따라서 차량 이동객체에 대한 정의와 거리 및 시간에 대한 제약 조건을 서술하였다. 그리고 차량 이동객체 질의를 위한 차량 위치 연산자에 대한 정의도 함께 서술하였다.

세 번째, 효율적으로 차량 운행궤적을 표현할 수 있는 방향각과 시간간격을 혼합한 하이브리드 기법에 대해 제안하였으며, 제안한 기법으로 추출된 차량 위치정보 처리 방법에 대해 서술하였다. 이를 위한 선행 작업으로 차량 위치정보 추출을 위한 전처리 단계를 작성하였다.

네 번째, 제안 기법의 성능을 평가하기 위해 분석 방법을 제시하였으며, 제시된 분석 방법을 통해 제안 기법의 효율성을 검증하였다.

2. 연구 내용 및 논문 구성

본 연구는 차량 운행 중에 발생하는 방대한 차량 위치정보 중 의미 있는 차량 위치정보만을 추출하여, 차량 위치정보량을 최소화 시키고 또한 추출된 위치정보를 이용하여 차량 유사궤적을 생성하는 기법을 제안하였다.

본 연구의 전체 구성 및 각 장의 내용은 다음과 같다.

2장의 내용은 선행 연구 및 관련 연구로서 시공간 이동객체와 종류 그리고 기존 주기적인 차량 위치 보고에 따른 차량 검색 시 문제점을 서술

하였으며, 이동객체 관리를 위한 대표적인 응용 시스템에 대한 연구에 대해 기술하였다. 또한 차량 이동객체의 현재 위치정보를 추출하기 위한 위치검출(Positioning)방법에 대하여 서술하였다.

3장에서는 차량 위치정보를 추출하기 위해 차량 이동객체의 정의 및 표현 방법에 대하여 서술하였다. 그리고 차량 이동객체의 제약 조건과 연산자를 정의하였다.

4장은 본 연구에서 제안한 방향각 정보와 시간 간격을 이용하여 유사궤적을 추적할 수 있는 하이브리드 기법과 이를 통해 추출된 차량 위치정보 처리 방법에 대한 알고리즘과 함수들을 정의하였다.

5장은 제안 기법의 성능 평가를 위해, 실시간 GPS수신 차량 위치정보와 시간간격별 차량 위치정보 그리고 제안 기법을 통해 추출된 차량 위치정보들을 비교 분석할 방법을 제시하였다. 제시된 분석 방법을 통해 제안 기법의 효율성을 검증하였다.

끝으로 6장 결론에서는 본 연구의 결과와 기대효과 그리고 향후 연구 과제에 대하여 기술하였다.

II. 관련 연구

차량은 시간 변화에 따라 위치정보가 변하는 시공간 이동객체 중 이동점(Moving Point) 객체로서 연속적인 공간 정보를 갖는다. 따라서 2장에서는 시공간 이동객체에 대해 설명하고, 차량 위치정보를 관리하기 위한 기존 시스템들의 문제점과 주기적인 보고에 의한 차량 위치정보 표현 방법의 문제점을 서술 하였다.

1. 시공간 이동객체



시공간 이동객체(Martin Erwig 등 1997)(Luca Forlizzi 등 2000)는 시간에 따라 객체의 공간 정보가 연속적으로 변경되는 객체로 크게 이동점 객체(Moving Point Object)와 이동영역 객체(Moving Region Object)로 나눌 수 있다. 이동점 객체는 시간에 따라 객체의 위치가 변하는 것으로, 사람이나 동물, 자동차, 비행기, 배 등이 여기에 속한다. 그리고 이동점 객체의 특정 시점에 대한 질의는 그 시간에 이동점 객체가 존재하는 위치를 표현하는 점(Point)을 반환한다. 이동점에 대한 질의는 시공간 이동객체의 위치에 대한 질의이며 시공간 이동객체가 가지는 면적에 대해서는 질의를 수행하지 않는다. 이동점과 관련한 질의들은 객체의 이동 경로나 방향, 거리 등에 대한 질의들로 Table 1과 같다.

Fig. 1은 시공간 이동객체 중 이동점 객체의 이동을 보여주고 있으며, 시간 변화에 따른 이동점 객체의 이동 경로를 나타내고 있다.

Table 1. Query for moving point object.

이동점	질 의
사 랫	A 지역을 순찰중인 경찰관의 위치는?
동 물	우리나라에 서식하는 겨울 철새인 가마우지의 이동 경로를 보이시오
차	승객이 요구한 위치에서 가장 가까운 택시는?
배	1시간 내에 부산항에 도착 예정인 선박은?

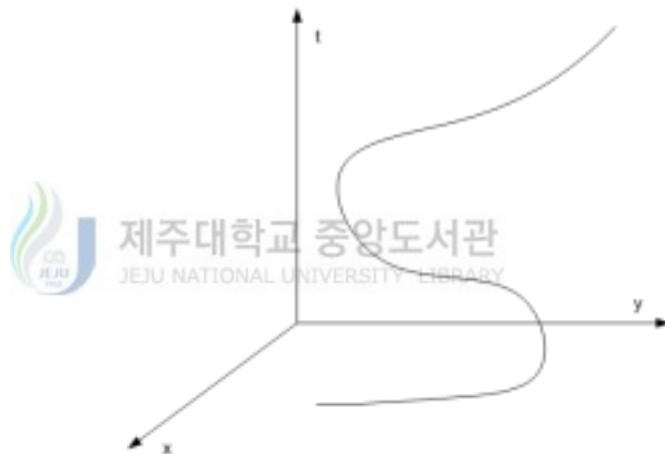


Fig. 1 A Spatiotemporal moving point object.

이동영역 객체는 시간에 따라 객체의 위치뿐만 아니라 모양까지 변하는 것으로 한 국가의 행정 구역이나 폭풍의 영향권, 암세포의 상태 등을 예로 들 수 있다. 이러한 이동영역 객체의 특정 시점에 대한 질의는 그 시점에 이동영역이 존재하는 위치와 모양을 표현하는 영역을 반환한다. 이동 영역에 대한 질의는 특정 시간에 이동객체의 위치나 모양에 대한 질의 일 수 있으며 이와 관련한 질의들은 Table 2와 같다.

Fig. 2는 시공간 이동영역 객체를 보여주고 있으며, 시간 변화에 따라 이동객체의 모양과 위치가 변경되는 것을 볼 수 있다.

Table 2. Query for moving region object.

이동 영역	질 의
국가	고구려의 영토가 가장 넓었을 때의 국가 경계는 어떤 모습이었는가?
폭풍	2000년 6월 우리나라에 영향을 끼쳤던 4호 태풍 '카이탁'의 이동경로 및 영향 지역을 보여라.



Fig. 2 Moving region object.

2. 주기적인 보고에 의한 차량 위치정보 표현

차량은 시간 변화에 따라 연속적으로 변경되는 위치정보를 갖는 객체이므로, 시간이 지남에 따라 방대한 위치데이터가 발생한다. 이를 저장·관리하기 위한 방법으로 일정한 시간간격마다 차량 위치데이터를 저장하고 있다.

주기적인 보고에 따라 차량 이동객체의 위치 보고가 이루어질 경우 차량 이동객체의 마지막 보고 이후 이동 방향이 변경될 경우 이를 반영하지 못하여 실제로 질의 조건을 만족하는 차량 이동객체들을 검색하지 못하는 문제가 있다. Fig. 3은 T_0 시간에 차량 이동객체의 보고가 이루어 졌을 때 차량 이동객체들과 이들을 포함하고 있는 경계사각형을 보여주고 있다. 차량 이동객체 o_1 는 정지 상태이고 o_2 는 오른쪽 방향으로 이동함을 나타낸다. TPR-tree(S. Saltenins 외, 2000)(Y. Tao, D. Papadias 외, 2003)(S. Saltenis and C. S. Jensen, 2002)(C. M. Procopiuc 외, 2002)의 경우 T_2 시간에 해당 지역을 지나갈 수 있는 차량에 대한 질의가 수행되었을 때, Fig. 4와 같이 차량 이동객체가 마지막으로 보고한 방향으로 경계사각형(Bounding Rectangle)을 확장시킨다. 그러나 일정한 시간간격으로 보고가 이루어지게 되면 차량 이동객체가 마지막으로 보고한 이후에 방향이 변할 경우 실제 질의 조건을 만족하더라도 질의 결과에 포함되지 않는 경우가 발생할 수 있다. 또한 차량 이동객체의 마지막 보고 이후에 차량 이동객체의 속도가 변할 경우에도 질의 조건을 만족하는 질의 시 검색되지 않을 수 있다.



Fig. 3 Location of time T_0

Fig. 5는 T_2 시간에 차량 이동객체의 위치를 예측하기 위해 T_0 시간에 보고된 차량 이동객체들의 속도로 경계사각형을 확장시킨 것이다. 그러나 T_0 시간 이후에 o_2 차량 이동객체가 더 빠른 속도로 이동할 경우 T_2 시간에는 질의 영역을 벗어나게 되어 질의 조건을 만족하더라도 이를 반영하지 못하게 된다.

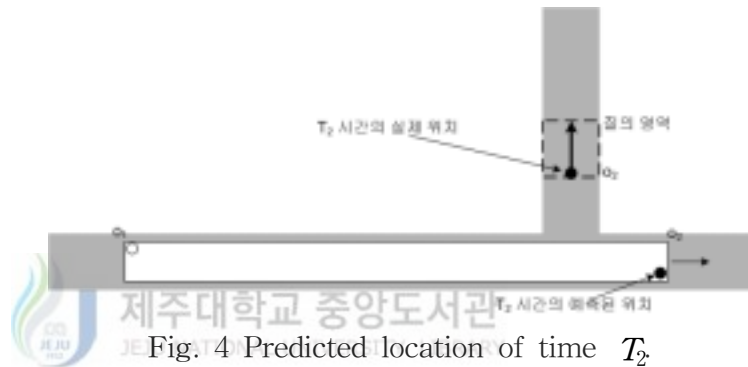


Fig. 4 Predicted location of time T_2

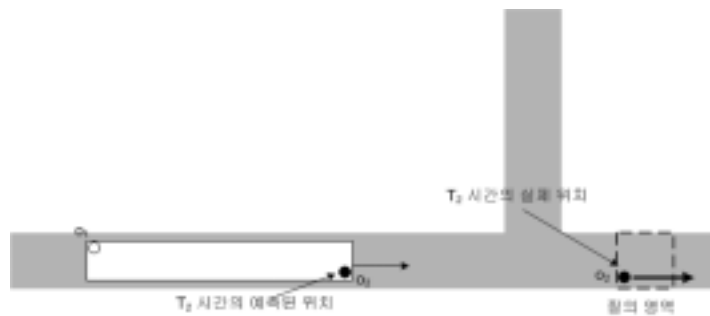


Fig. 5 Predicted location of time T_2

Fig. 6에서 네트워크 거리를 고려하지 않고 유클리디언 공간 질의를 수행했을 경우, 이동객체의 속도와 이동 시간에 의한 실제 이동 가능한 거리 20m라 하면 실제 네트워크 거리가 40m 이상인 곳을 예측할 수 없다. 또한, 시간간격마다 차량 위치정보를 저장하였을 경우, 저장된 차량 위치

정보를 이용하여 차량 운행 궤적을 표현하고자 하는 경우, 정확한 차량 운행궤적을 표현할 수 없다. 그리고 시간간격이 짧은 경우에는 정확한 차량 운행궤적을 표현할 수 있지만, 방대한 차량 위치정보를 발생시키는 계기가 된다.

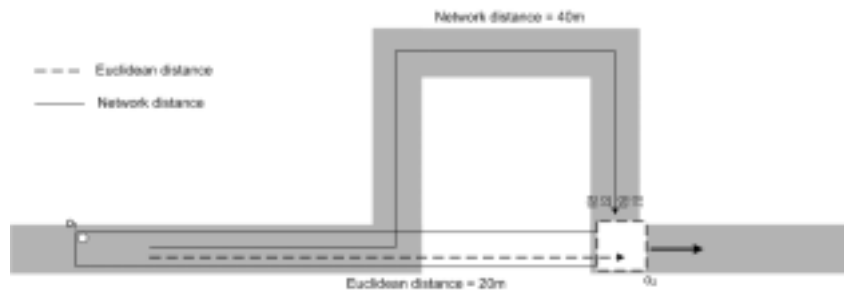


Fig. 6 Query processing on euclidean space.



제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

3. 차량 이력 데이터 관리 시스템

이동객체 관리를 위한 대표적인 응용 시스템 연구에는 DOMINO, CHOROCHRONOS, DEDALE, Battlefield Analysis가 있다.

1) DOMINO

DOMINO(Database fOr MovINg Object)(O. Wolfon, 1998)는 이동객체의 수송에 대한 웹 기반 실시간 궤도 응용의 개발을 촉진하는 이동객체 소프트웨어 도구로서, DBMS기술을 활용하여 만든 실시간 위치 추적 시스템 프로토타입이다. 그러나 DOMINO 프로토타입은 이동객체의 현재

위치, 속도, 방향정보를 이용하여 미래의 이동 위치를 예측하는 방법에 주로 초점을 맞추고 있다. 따라서 과거 시점을 포함하는 이동객체의 완전한 이동 경로인 궤적(Trajectory)을 관리할 수 없는 단점을 가진다.

2) CHOROCHRONOS

CHOROCHRONOS(Martin Erwig 등, 1997)는 시공간 데이터베이스의 특수한 형태인 이동객체에 관한 연구가 집중적으로 수행되었고, 이동객체의 데이터 모델링 및 인덱싱에 관한 연구 결과를 가장 많이 발표하였다. 특히 이 연구는 GPS기반의 수송 관리 시스템과 멀티미디어 시스템에 적용한 응용 시나리오를 제시하였다. 그러나 이동객체 데이터베이스를 활용한 응용 시스템의 모델 및 개발 사례는 제시되지 않고 있으며, 개발 중인 질의처리 시스템에서는 이동객체의 이력 데이터 저장 기법이 제시되지 않고 있다.

3) DEDALE

DEDALE(S. Grumbach 등, 1999)은 제약사항 데이터베이스 모델을 이용하여 시공간 데이터를 모델링하고 질의처리를 하기 위해 개발된 프로토타입으로, 기존의 시공간 데이터의 모델뿐만 아니라 이동객체의 궤적 등과 같은 데이터 모델 및 질의 표현도 제공한다. DEDALE은 고급 수준의 개발 및 최적화를 위한 질의어의 확장을 위해, 공간 질의에 대해 선형 제약사항 추상화를 제공한다. 또한 DEDALE 프로토타입은 데이터베이스에 시간의 변화에 따른 이동객체의 이력 (x, y) 좌표값이 직접 저장되지 않고, 특정 구간의 궤적을 표현하는 선형 제약 사항(Linear Constraint)의 공식이 저장된다. 이로 인해 빈번하게 이동하는 객체의 실시간 위치 추적을 위한 응용 시스템 개발에는 부적합한 특징을 가진다.

4) Battlefield Analysis

Battlefield Analysis(I. B. Oh 등, 2002), (S. S. Park 등, 2001), (M. H. Huh 등 2002)는 모의 전장에서 이동하는 부대 및 탱크들의 움직임을 예측하여 이를 의사결정에 활용할 수 있도록 개발된 전장분석 프로토타입이다. 전장분석 프로토타입은 이동객체 관리기와 추론 엔진을 접목시키고자 하는 데 초점이 맞추어졌다. 특히, 시공간 이동객체의 연산 결과를 추론 엔진에서 활용하는 새로운 이동객체 추론 모델을 제시하였다. 그러나 이동객체 관리 시스템의 실시간 환경이 고려되지 않았으며, 특정 도메인 지식을 활용한 이동객체의 미래의 위치정보만을 예측하고 있다. 따라서 임의의 과거 및 미래 시점에 대한 모든 위치정보를 제공하지 못하고 있다.



4. 위치검출 방법

차량 위치검출은 현재위치를 정확하게 파악하고, 경로탐색이나 경로 유도기술에 필요한 중요 기술이다.

위치검출 방법은 추측항법(Dead-Reckoning)과 맵 매칭(Map-Matching) 기술에 의해, 기본이 되는 (상대)위치를 계산하고, 이에 GPS 항법에 의한 절대위치정보를 결합시켜 최종적인 위치를 계산하는 하이브리드방식이 일반적이다.

본 연구에서는 위치검출 방법 중 GPS 위치검출 방법을 이용하였으며, 다양한 위치검출 방법에 대해 알아보았다.

1) 추측항법 (DR : Dead-Reckoning)

Dead-Reckoning이란 차량 자체에 설치된 거리 및 방위센서에 의해 이동거리와 방위를 검출함으로써 현재위치를 추측하는 자립항법에 의한 계산 방식이다.

Dead-Reckoning에 사용되는 센서에는 속도센서와 방위센서가 있다.

속도센서에는 차륜속형(車輪速型), 스피드형(도플러 레이더형이나 공간 필터형), 가속도형 등이 있지만, 차륜속형이 많이 이용된다. 이는 소형이면서 수명이 길고, 가격이 싸고, 차륜의 회전수를 검출하여 이동거리를 측정하는 방식이다.(G. Siouris, 1993) (김진원 등, 1996)

방위센서에는 절대방위형인 지자기(地磁氣 : 마그네트 컴퍼스)센서 등이 있지만, 각속도(角速度)검출형인 진동형 레이트자이로가 일반적이다.

Dead-Reckoning의 계산 원리는 어느 지점(X_0, Y_0)에서 출발하여 임의의 지점(X_n, Y_n)에 도달했을 때의 좌표를, 방위 센서와 거리센서를 통해 단위시간당 얻어진 데이터에서 각 지점의 좌표(주행거리와 방위)를 구하고, 벡터가산으로 구하는 것이다. 1 단위 시간에 거리와 방위센서에서 16개 데이터를 얻고, 그 중에서 각각 최대, 최소 데이터를 버린 나머지 14 데이터를 채용하여 계산 처리하여, 각 지점의 X, Y 추정 좌표를 구한다.

2) 맵 매칭(Map-Matching)

Dead-Reckoning으로 구해진 추정좌표는 거리나 운전시간이 길어짐에 따라 센서 오차가 많이 발생하고, 또한 주행 조건에 의한 오차가 누적되어 지도상에 실제 위치를 정확하게 표시할 수 없는 경우가 많다.

이 오차를 보정하고, 지도상의 위치에 현위치 좌표를 수정하는 것이 맵 매칭이다.

맵 매칭의 원리는 일반적으로 자동차는 도로상을 주행한다. 바꿔 말하

면 넓은 주차장 같은 곳을 제외하면 보통 자동차는 도로 이외는 주행하지 않는다는 말이다. Dead-Reckoning으로 구해진 추정좌표에 의한 주행궤적과 CD-ROM에서 읽어낸 디지털 지도의 도로 모양을 항상 대조하면서 누적되는 오차를 그때그때 보정하고, 정확한 현재위치를 디지털 지도상에 표시하는 기술이 맵 매칭의 기본적인 원리이다.

3) GPS(Global Positioning System) 방식

GPS(Global Positioning System : 전 지구규모 측위 시스템)은 미 국방성이 개발하여, 미 운수성과 공동으로 운용 관리하고 있는 인공위성에 의한 전 세계적인 전파측위 시스템이다.

GPS 측위 원리는 위성과 이용자(GPS 수신기) 사이의 거리를 측정하여 측위를 한다. 3개의 위성이 있어서 각각의 위치를 이미 알고 있다면, 위성에서 이용자까지의 거리를 알면 위치가 확정된다.

이용자가 측위를 하려고 하는 임의의 순간의 GPS 위성의 위치는 위성에서 방송되고 있는 궤도 정보에 의해 정확히 계산할 수 있다.

위성에서 이용자까지의 거리는 위성에서 시각이 송출되고 있어서 전파가 위성을 나온 시각과 이용자가 수신하는 시각(수신기 내장시계)의 차이에서 전파 전달시간을 알 수 있고, 거기에 광속을 곱하면 거리를 계산할 수 있다.

이용자 수신기 내부시계는 정확한 것은 아니다. 따라서 이렇게 측정된 전파 전달시간은 상당히 부정확하므로, 거리도 정확하지는 않다. 이렇게 하여 구한 거리를 의사 거리(疑似距離)라고 한다.

정확한 거리를 구하기 위한 4번째 위성이 필요하게 된다.

이용자의 수신기 내부시계가 위성 시계와 완벽하게 같다면, 각각의 위

성까지의 의사거리를 반경으로 하는 4개의 구면과 1점에서 교차한다. 이용자의 내부시계가 맞지 않을 때에는, 수신기의 계산기가 4개의 구면이 1점에서 교차하도록 시계를 수정한다. 이렇게 하여 자신(이용자)의 위치(지구상의 절대 위치)를 알 수 있는 것이다.

GPS 수신기에 의한 측위(단독측위)는 동시에 4개의 GPS 위성이 가시 범위에 있어야 한다. 그리고 궤도 정보나 보정 정보 등의 Almanac이나 Ephemeris를 해독, 신호 도달 시각 측정치를 근거로 자신의 위치를 계산한다.

4) 하이브리드(Hybrid) 방식

하이브리드 방식의 위치 취득에는 차속센서와 방위센서 및 GPS가 필요하다. GPS 방식의 단점은 전파가 도달하지 않는 곳에서 측위가 곤란한 점이며, 추측항법에 의한 자립 방식의 결점은 누적오차이다. 이들 결점을 보완하고 각각의 장점을 살리는 방법으로서 양자를 결합한 것이 하이브리드 방식이다.

5) 측위 방식별로 본 측위율 비교

아래 표는 동일 장소에서 GPS 단독방식, 추측항법방식, 하이브리드 방식의 측위율과 거리 오차를 비교한 표이다.

Table 3은 GPS 메이커의 실험 결과에서 발췌하였다.

평균거리오차와 측위율이라는 2가지 요인에서 종합하면, 하이브리드 방식의 안정성을 알 수 있다.

Table 3. Positioning rate and distance error.

구분	측위율(%)	평균거리오차(m)	최대거리오차(m)
GPS	67	13.3	38.6
추측항법	100	90.2	249.5
하이브리드	100	10	36.6

5. 관련 연구의 요약

앞에서 기술한 것과 같이 기존의 차량 관리 시스템들은 방대한 차량 위치정보를 계속적으로 저장·관리 한 것이 아니라, 차량의 최근 위치정보만을 저장·관리했기 때문에 차량의 위치추적과 차량 운행궤적을 표현하는데 문제가 많았다.

주기적인 보고에 의한 차량 위치정보는 시간간격에 따라 차량 위치데이터의 증가와 감소가 심각하게 발생한다. 그리고 시간간격에 따라 차량 운행궤적 모양이 수시로 변경되고 또한 차량 운행궤적이 일관성이 없다.

차량 위치정보를 단순히 GPS에서 수신된 위치 좌표값과 시간만을 저장하기 때문에 위치기반서비스의 질 개선을 위한 자료로 사용할 수 없다.

차량 이력 관리 시스템인 DOMINO는 이동객체의 현재 위치, 속도, 방향 정보를 이용하여 미래의 이동 위치를 예측하는 방법에 주로 초점을 맞추고 있어 과거 시점의 이동객체의 완전한 이동 경로 궤적을 관리할 수 없다.

CHOROCHRONOS는 이동객체 데이터베이스를 활용한 응용 시스템의 모델 및 개발 사례가 제시되지 않았으며, 이동객체의 이력 데이터 저장 기법도 제시되지 않았다. 따라서 과거의 차량 이력 데이터를 검색하지 못

하는 단점이 있다.

DEDALES은 특정 구간의 궤적을 표현하는 선형 제약 사항의 공식이 저장되어 있어 빈번하게 이동하는 객체의 실시간 위치 추적을 위한 응용 시스템 개발에 부적합하다.


Battlefield Analysis는 이동객체의 미래의 위치정보만을 예측하고 있어 임의의 과거시점의 위치정보를 제공하지 못하는 단점이 있다.

본 연구에서는 측위검출 방법은 GPS측위에 의해서만 차량 위치정보를 추출하여 사용하였다.

Ⅲ. 차량 이동객체 정의 및 제약 조건

이 장에서는 이동점 객체인 차량 이동객체에 대한 정의와 차량 이동 행위 그리고 차량 이동 정보를 표현하였다. 또한 차량 이력 데이터 추출을 위한 차량 이동객체의 시간 및 거리 제약 조건과 차량 이동객체 처리를 위한 연산자를 정의하였다.

1. 차량 이동객체 정의

 제주대학교 중앙도서관
본 연구에서는 운행 중인 차량 이동객체의 위치정보만을 관리하기 위해, 이동점을 대상으로 하는 이동객체인 차량 이동객체의 데이터를 다음과 같이 정의 하였다.

[정의 1] (차량 이동객체) 시간의 변화에 따라 차량 객체의 위치 값만 변화되는 차량 이동객체를 말한다. 차량 이동객체 MP 는 시간 속성(Time attribute), 공간 속성(Space attribute), 일반 속성(General attribute)을 가지며, $MP = \langle T_A, S_A, G_A \rangle$ 가 된다.

[정의 2] (시간 속성) MP 의 시간 속성 $T_A = \langle ut_s, ut_e \rangle$ 로 구성되며 ut_s 는 시작 시간, ut_e 는 종료 시간을 나타낸다. 이 때 ut_s 와 ut_e 는 유효시간(Valid time)의 집합 S_{VT} 의 원소가 된다. 유효

시간은 실세계에서 발생한 시간을 나타내며, $S_{VT} = \{t_0, t_1, t_2, \dots, t_{now}\}$ 이고, 각 원소들은 $t_0 < t_1 < t_2 \dots < t_k < \dots < t_{now}$ 의 순서를 가진다. $t_k = t_{k-1} + 1$, $t_k = t_0 + k$, $k \geq 0$ 인 정수로 정의된다. t_{now} 는 현재 시간을 의미하는 시간 함수이다.

[정의 3] (공간 속성) MP 의 공간 속성 $S_A = \langle x, y \rangle$, $x, y \in R$, R 은 실수이다.

[정의 4] (차량 이동객체 데이터베이스) 차량 이동객체 데이터베이스를 구성하는 차량 이동객체들의 집합은 $S_{MP} = \{MP_0, MP_1, \dots, MP_n\}$ 이다. S_{MP} 로 구성된 차량 이동객체 데이터베이스의 차량 이력 집합은 $H_{MP} = \{H_{MP_0}, H_{MP_1}, \dots, H_{MP_n}\}$ 이다. S_{MP} 에 속하는 각각의 MP_i 에 대한 모든 차량 이력 집합은 $H_{MP_i} = \{mp_{i_0}, mp_{i_1}, \dots, mp_{i_n}\}$ 이고, $mp_{i_k} = \langle T_A(mp_{i_k}), S_A(mp_{i_k}), G_A(mp_{i_k}) \rangle$ 이다.

[정의 4]에서 mp_{i_k} 는 MP_i 의 k 번째 차량 이력 정보를 의미하고, $T_A(mp_{i_k})$ 는 MP_i 의 k 번째 시간 속성, $S_A(mp_{i_k})$ 는 k 번째 공간 속성, $G_A(mp_{i_k})$ 는 k 번째 일반 속성을 의미한다.

2. 차량 이동객체의 거리 제약 조건

차량 이동객체의 빈번한 위치 이동과 교통 상황에 따라 많은 차량 위치 정보가 발생한다. 예를 들면, 정체구간에서 차량 속도의 저하로 운행한 거리가 짧아진다. 따라서 최소 이동 거리 제약 조건을 정의하여 차량 이력 데이터를 감소시킬 필요가 있다.

차량 이동객체의 위치정보 요약을 위하여 거리 제약조건을 정의하였다.

[정의 5] 차량 이동객체의 차량 거리 제약 조건은 차량의 최소 움직임을 지정하는 것이다. 차량 이력객체 MHI_i 에 대하여 거리 제약 조건(Distance constraint) min_dist 는 차량 이력 객체 내의 인접한 두 이동 점 P_i, P_{i+1} 간에 만족해야 하는 최소 이동 거리이다.

차량 이동객체의 위치 요약은 인접한 두 이동은 최소 거리 임계치 min_dist 를 적용하여 다음의 식 (1)을 만족하도록 하는 것을 말한다. 여기에서 $dist()$ 함수는 두 이동간의 공간상의 유클리디언 거리를 구하는 함수이다.

$$dist(P_i, P_{i+1}) > min_dist, \text{ for } 1 \leq i < n \quad (1)$$

Fig. 7은 차량 이동객체의 운행 경로를 min_dist 적용하여 차량 운행 경로를 표현하였다.

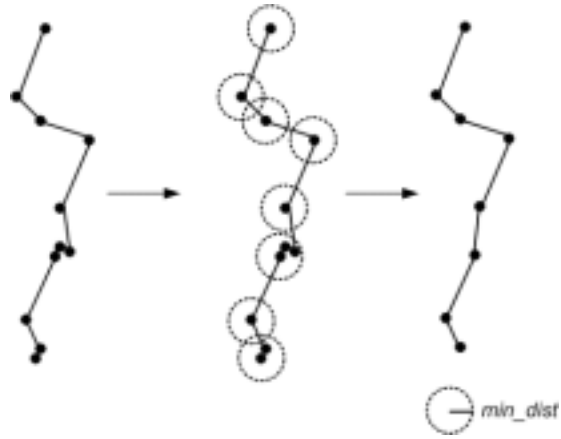


Fig. 7 Min_dist application example using minimum distances critical.

3. 차량 시간 제약 조건



차량 이동객체 데이터 집합 MD 는 차량 객체 식별자 MP 와 차량 객체의 위치 이동의 연속적인 나열로 구성되어 있다.

차량 이동 시퀀스 S 는 위치 이동의 연속적인 모임으로써 $S = \{LC_1, LC_2, \dots, LC_n\}$ 으로 정의한다. 여기서 $LC_i = \langle (x_i, y_i), VT_i \rangle$ 로서 2차원 공간 평면상의 좌표 x_i, y_i 와 이동 시간 VT 로 구성된다. 예를 들어 $S = \{ \langle (100, 95), 10 \rangle, \langle (125, 93), 22 \rangle, \langle (160, 117), 27 \rangle, \langle (180, 120), 32 \rangle \}$ 와 같이 표현된다.

차량 이동 시퀀스는 한 차량 이동객체가 특정한 구간을 운행했을 경우에 운행 궤적을 표현하기 위한 연속된 차량 위치정보들의 집합이다. 따라서 특정 구간의 시작과 종료 시점이 반드시 있어야 한다. 만약 운행 궤적의 시작 시점만 있고 종료 시점이 없이 계속해서 운행을 한다면, 하나의 차량 이동 시퀀스만 생성되고, 중복되는 구간들이 발생하게 된다. 따라서

모든 이동간의 시간간격을 고려하여 차량 이동객체의 시작과 종료 시점을 구분할 필요가 있다. 이를 위하여 시간 제약조건을 정의한다.

[정의 6] 최대 시간간격 max_gap 은 연속하고 있는 두 이동 LC_{i-1}, LC_i 간에 만족해야 하는 시간 제약 조건이다.

[정의 6]의 시간 제약 조건 max_gap 은 이동 시퀀스 내의 두 위치 이동이 식 (2)을 만족하지 못하면 서로 다른 이동 시퀀스로 분리하기 위하여 사용할 수 있다.

$$LC_i[VT] - LC_{i-1}[VT] \leq max_gap \quad (2)$$

4. 차량 이동객체 연산자

차량 이동객체 연산자는 차량 이력 데이터베이스에 저장되지 않은 차량 위치의 추정 및 검색 시에 사용된다.

이동 차량의 위치정보 관리를 위해 다음과 같은 차량 이동객체 연산자를 정의하였다. 차량의 일반 속성에 대한 검색은 기존의 상용 데이터베이스 시스템이 제공하는 기능을 그대로 사용하므로 별도로 정의하지 않는다. 차량 이동객체의 위치정보 관련 연산자로 $m_distance$, $trajectory$, $length$, $position_at$, $m_nearest$, $m_farthest$ 를 Table 4와 같이 정의한다.

Table 4는 기존의 이동객체 연구에서 제시된 연산자들을 토대로 기본 연산자만을 정의한 것이다(P. Sistla 등 1997) (O. Wolfson 등 1999) (한

경복 등, 2006). 입력 및 출력 값에서 G_A 는 차량 소유주, 운전자, 차종 등과 같은 차량 이동객체의 일반 속성을 의미한다. S_A 는 공간 속성을 나타내고, T_A 는 시간 속성인 유효시간을 의미한다. Real은 실수 값을 나타내며, { }로 묶인 속성들은 집합을 의미한다.

Table 4. Vehicle location operator.

종류	입력 값	출력 값	기능
m_distance	$G_A \times G_A \times T_A$	Real	특정 유효시간 동안 두 이동객체간의 거리 계산
trajectory	$G_A \times T_A$	{ S_A }	이동객체의 특정 유효시간 동안의 이동 경로 추출
length	{ S_A }	Real	이동객체의 특정 유효시간 동안의 거리 계산
position_at	$G_A \times T_A$	S_A	임의의 한 시점에 대한 이동객체의 위치 검색
m_nearest	$G_A \times T_A$	S_A	특정 유효시간 동안 가장 가까이 위치하는 객체 검색
m_farthest	$G_A \times T_A$	S_A	특정 유효시간 동안 가장 멀리 존재하는 객체 검색

IV. 차량 유사궤적 및 이동 시퀀스 추출

이 장에서는 GPS에서 연속적으로 수신된 차량 위치정보에서 방향각을 계산하였으며, 계산된 방향각정보와 시간간격을 이용한 하이브리드 기법으로 유사궤적 추적 시스템을 제시하고, 유사궤적을 생성·추적하기 위한 전처리 작업 및 제약 조건 범위를 지정하였다. 추출된 방향각과 차량 위치정보로 유사궤적을 생성하였으며, 이를 하나의 시퀀스를 생성하여 데이터베이스에 저장 관리할 수 있도록 하였다.

1. 차량 유사궤적



차량은 시간 변화에 따라 위치정보가 매 시간마다 갱신되는 시공간 이동객체이므로 방대한 데이터가 발생하고, 이를 처리하기 위해서는 컴퓨터의 많은 자원(resource)을 사용하게 되므로, 매 시간 마다 수신되는 차량 위치정보 중 의미 있는 차량 위치정보를 추출하여, 이를 관리하는 방법이 필요하다. 따라서 본 연구에서는 방향각 정보와 시간간격을 이용한 하이브리드 기법을 제안하였다.

차량은 도로 위를 이동하는 이동객체이므로 도로 조건과 상황에 따라 위치정보가 변경되는 특징을 가지고 있다. 예를 들어, 차량은 직선도로에서 직선으로 이동하고, S자 도로에서는 S자 형태로 이동해야한다. 그리고 교차로에서는 좌회전과, 직진, 우회전 등, 도로의 방향에 의해 차량 방향이 결정된다. 따라서 차량 운행궤적이라고 함은 도로 형태들의 집합이라

고 할 수 있다. 이를 다시 말하면 차량의 진행 방향각을 변경시키는 요인들의 집합이라고 할 수 있다. 즉, 직선도로에서는 거의 방향각 변화가 없지만, 우회전과 좌회전을 하는 경우에는 급격히 차량 진행 방향각이 변경 시점이 있다. 따라서 차량 진행 방향각이 급격히 변경되는 시점의 차량 위치정보를 추출하여, 운행궤적으로 표현 할 수 있다.

제안기법은 GPS에서 수신된 차량 위치정보를 이용하여 방향각을 추출하고, 방향각이 급격히 변하는 시점의 차량 위치정보를 저장·관리하는 것이다.

예를 들면, Fig. 8은 차량에 부착된 GPS를 이용하여 Start 지역에서부터 End 지역까지 제주시 지역을 1시간 동안 차량을 운행한 경로이며, 수신 간격은 1초이고 처음 이동하기 시작하여 5m이내의 위치정보는 무시되었다.

수신된 차량 위치정보는 1900개이고, 이를 각각의 차량 위치정보에서 차량 운행하는 이동 방향각이 변경되는 시점의 위치정보를 추출하여 이를 하나의 차량 이력 시퀀스 정보로 저장 관리하도록 하였다.



Fig. 8 Vehicle moving paths.

Fig. 9는 Fig. 8에서 일부지역을 추출한 것으로 A지점에서 L지점까지 차량이 운행 중이라고 하면, 차량이 운행하는 각각의 지점에서 차량 방향각을 추출하였을 때, 급격히 방향각이 변경되는 시점의 차량 위치정보를 저장·관리하여 차량 유사궤적을 생성하는 기법이다. 예를 들면, A지점에 B지점으로 차량 이동시 추출된 각도는 41.4° 이며, I지점까지는 거의 동일한 각도를 갖는다. 하지만 I지점에서 J지점까지는 45° , J지점에서 K지점까지는 102.2° 을, 그리고 K지점에서 L지점까지는 123° 를 갖으면서 급격히 차량 방향각이 변경되고 있다. 따라서 차량 운행 중 차량 방향각이 급격히 변경되는 시점인 J지점과 K지점의 차량 위치정보 저장하는데 시작지점인 A지점의 차량 위치정보와 J지점, K지점, L지점의 차량 위치정보만을 저장하고 유사한 방향각을 갖는 차량의 위치정보를 제거하였을 경우, 적은 량의 차량 위치정보를 이용하여 차량 운행궤적으로 표현할 수 있다.

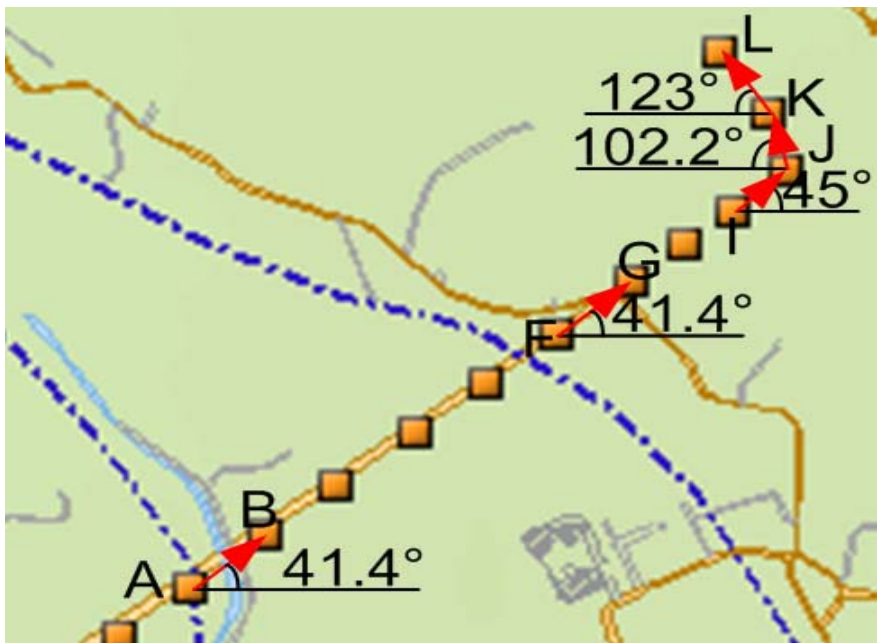


Fig. 9 Vehicle similar trajectories.

2. 방향각 추출

본 절에서는 차량 운행 경로 위치정보에서 각도 추출방법에 대해서 논하였다.

차량 운행 중 방향 정보의 변경은 기본적으로 시공간 이동객체인 차량 이동객체의 이동에 따라 이동 방향을 나타내는 각 θ 가 변경되고, 이 각 θ 가 주어진 오차의 범위를 벗어날 경우 이동 정보의 변경으로 보고 명시적 갱신 연산을 수행하는 것이다. 점 P 는 차량 이동객체의 이동 위치를 나타내며, O 는 기준선을, θ 는 이동 방향을 나타낸다. O 의 X 좌표는 P 의 X 좌표이며 O 의 Y 좌표는 기준점의 Y 좌표이다. 각 θ 는 명시적 갱신이 수행되는 시점에 설정된 기준 각이다. 또한 θ_n 은 기준선 O (또는 O_n)와 차량 이동객체의 위치 P_n 사이의 각을 나타내며 θ_1 에서 θ_4 로 변한다고 가정한다. 두 방법 모두 동쪽을 0° 로 한다. 방향 정보의 변화를 판단하는 방법으로 본 연구에서는 두 가지 방법을 사용하였다.

첫 번째 방법은 n 번째 차량 이동 정보의 입력에서 차량 이동객체의 이동 방향을 나타내는 각 θ_n 은 n 번째 차량 이동 정보의 위치와 $n-1$ 번째 차량 이동 정보의 위치 그리고 기준선 O_{n-1} 이 이루는 각을 구하여 이를 행위 정보에 기록된 방향 정보와 비교하는 것에 의해 이동 방향의 변경 여부를 판단한다. 이 방법에서는 기준선이 동적으로 구해지므로 각각의 차량 이동 정보 입력마다 새로운 기준선 O_{n-1} 의 위치를 계산하는 과정이 필요하다. n 번째 차량 이동 정보의 이동 방향을 나타내는 각 θ_n 은 식(3)으로 표현할 수 있다.

$$\theta_n = \angle P_n P_{n-1} O_{n-1} \text{ (단, } n \geq 1 \text{)} \quad (3)$$

다음의 Fig. 10에서 차량 이동 정보 P_1 이 입력될 때 이동 정보에 기록된 방향 정보인 θ 와 현재 입력하는 차량 이동 정보의 이동 방향을 나타내는 θ_1 의 차이를 계산하여 주어진 오차의 범위를 넘으면 이동 정보를 변경한다. P_2 역시 θ 와 θ_2 의 차이로 행위 정보 변경 여부를 판단하며 이후는 모두 동일하다.

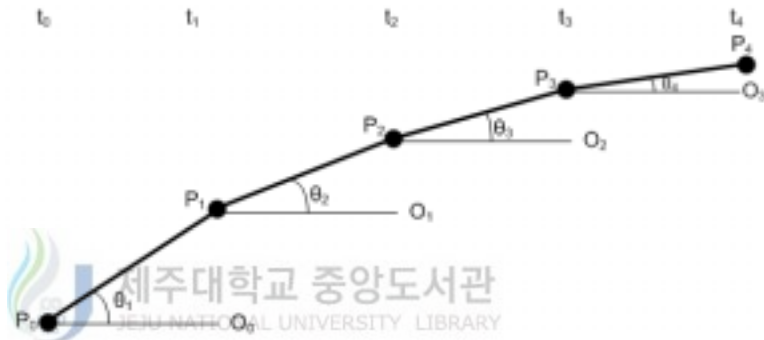


Fig. 10 Dynamic base line.

만일 P_4 의 입력 시점에 이동 정보가 변경되었다고 가정하면, 이동 정보의 위치정보에는 P_4 , 방향 정보에는 θ_4 가 설정된다.

두 번째 방법은 이동 정보의 명시적 수정 연산 수행 시 설정된 시작 위치와 입력된 이동 정보의 위치사이의 각을 기준으로 이동 방향의 변화를 판단한다. 이 방법에서 기준선 O 는 항상 고정적이다. n 번째 이동 정보의 이동 방향을 나타내는 각 θ_n 은 식(4)로 표현할 수 있다.

$$\theta_n = \angle P_n P_o O \quad (\text{단, } n \geq 1) \quad (4)$$

다음의 Fig. 11에서 점선은 16방위 중의 한 부분으로 동일한 방향으로

처리되는 각의 범위, 즉, 오차의 범위이다. θ_n 은 P_n 의 이동 방향을 나타낸다.

Fig. 11에서는 이동 방향을 나타내는 각 θ_4 가 오차의 범위를 벗어나는 시점인 t_4 의 입력에서 이동 정보의 변경이 발생한다. 이때 이동 정보의 위치정보에는 P_4 의 위치가 방향 정보에는 P_4 가 위치하는 영역을 나타내는 방위의 대표 값이 설정된다.

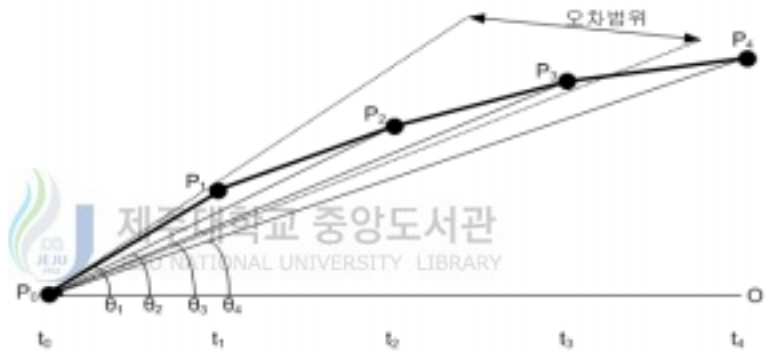


Fig. 11 Fixing base line.

두 가지 방법 모두 각각의 이동 정보 입력에서 시공간 이동객체의 이동 방향을 나타내는 θ_n 을 구해야 한다. 그러나 θ_n 을 이루는 세 점의 특성은 두 방법에 차이가 있다. θ_n 은 $\angle P_1 P_2 P_3$ 라고 하면, 첫 번째 방법은 P_1 , P_2 , P_3 모두가 계속적으로 변경된다. 즉, 이들 세 점을 구하기 위한 추가적인 연산이 필요하다. 반면에 두 번째 방법은 P_2 와 P_3 는 고정적이며 P_1 만을 구하면 되므로 첫 번째 방법보다 추가적인 연산이 적다. 두 방법에서 행위 정보로 기록되는 방향 정보의 값을 살펴보면 첫 번째 방법에서는 특정 크기의 θ 을 가지며 오차의 크기는 응용 프로그램에서 정의되고, 두 번째 방법은 해당 방위별 대표 값이 저장된다. 두 방법은 응용의 특성

에 따라 선택 가능하다.

Fig. 12는 차량 이동 정보 추출 알고리즘을 나타낸 것이다. 데이터베이스에 저장된 차량 위치정보를 검색하여 선분과 각도를 추출하여 차량 이동 정보를 표현하였다.

유사궤적은 차량 이동객체의 궤적을 이루는 각각의 움직임들을 단일 속성인 실제 각도(0~360)로 표현하고, 사용자 궤적 $Q = (q_1, q_2, \dots, q_M)$ 을 추출하였다.

```
public boolean AngleAbstraction(int ID, int x, int y){
    DBmanager dm = new DBmanager();
    /* Base Location */
    int baseX = dm.getBaseX(ID);
    int baseY = dm.getBaseX(ID);
    /* Current Location */
    int currentX = x;
    int currentY = y;

    /* Get Length of current to base */
    int line1 = this.getLength(baseX, baseY, currentX, currentY);

    /*Get Length of base line */
    int line2 = this.getLength(baseX, baseY, currentX, baseY);

    /*Get eaceh angles */
    int currentAlgle = this.GetAngle(line1, line);
    int baseAlgle = dm.getBaseAngle(ID);
    int diff = abs(currentAlgle - baseAlgle)
    /* Compare */
    if(diff > dm.getBaseOrError(I(D))
        return true
    else
        return False
}
```

Fig. 12 Extraction algorithm for vehicle moving action information.

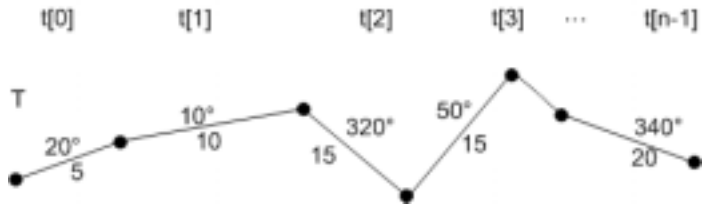


Fig. 13 Trajectory of vehicle moving object T.

차량 이동객체의 궤적 T 을 구성하는 움직임 요소($t[i]$)에 대해서 Fig. 13과 같이 방향(Direction) 정보와 거리(Distance) 정보를 이용하여 모델링 하였다.

주어진 차량 이동객체의 궤적 정보에서 움직임 정보를 위해 방향 정보만을 추출하여 방향 궤적 T_{ang} 을 구성하였다.

제주대학교 중앙도서관
JEJU NATIONAL UNIVERSITY LIBRARY

$$T_{ang} = \{a_i | i=0, \dots, n-1\}, n = |T_{ang}| \quad (5)$$

여기서, a_i 는 방향 궤적 T_{ang} 을 구성하는 i 번째 움직임 요소의 방향 정보를 나타내며 실제 각도 $[0, 360]$ 을 이용하여 표현하였다. $n(=|T_{ang}|)$ 은 궤적을 구성하는 움직임의 수를 의미한다. 궤적을 구성하는 임의의 움직임 요소 a_i 와 a_{i-1} 에 대해서 ADF(Angle Difference Function)함수를 이용하여 두 움직임 요소 사이의 각도의 차(Angle Difference : AD)를 계산하였다.

Fig. 14는 차량 이동객체 방향각 추출함수다.

$$AD_i = ADF(a_i, a_{i+1}), i = 0, \dots, n-2 \quad (6)$$


```

int ADF(int a1, int a2)
{
    int Ang_Diff;
    Ang_Diff = 180° + (a2 - a1);
    if(Ang_Diff > 360°) Ang_Diff = Ang_Diff - 360°;
    else if(Ang_Diff < 0) Ang_Diff = Ang_Diff + 360°;
    return Ang_Diff;
}

```

Fig. 14 Direction angle abstraction function of vehicle moving object

3. 유사궤적 생성을 위한 전처리

본 절은 앞 절에서 제안한 기법을 통하여 차량 이동 방향각을 추출하기 위한 준비단계로 적정한 각도 범위를 지정할 필요가 있다. 이는 본 연구에서 제안한 하이브리드 기법은 차량 이동객체가 운행 중 차량 이동 방향각이 급격히 변경되는 시점의 위치정보를 저장하는 방법이다. 도로 위를 이동하는 차량 이동객체는 잦은 운행 경로 변경으로 많은 방향각 변화가 발생한다. 따라서 이를 계속해서 저장한다면 방대한 데이터가 발생하여 많은 시스템 자원을 사용하게 된다. 따라서 적정범위의 방향각 차가 발생했을 경우에만 저장하도록 하였다.

본 연구에서는 동쪽을 0°로 설정하여 방향각을 계산하도록 하였으며, 방향각이 급격히 변경되는 지점의 차량 위치정보만 저장하도록 하였다.

정확한 분석 및 유사궤적을 추출하기 위해 GPS장비 오차 범위를 측정하여 오차 범위를 설정하였고, 또한 차량 운행 중 신호대기 및 정차 시 수집된 차량 위치정보를 제거하기 위해, 제거 요소들을 정의하였다. 그리고 3장에서 정의한 차량 이동객체 제약 조건에 따른 임계 영역을 정의 하였

다.

1) GPS 수신 NMEA 형식과 오차 범위 측정

(1) NMEA 형식

NMEA(National Marine Electronics Association)는 해양 기자재들 간의 통신을 위한 전기적 인터페이스 및 데이터 프로토콜로 현재 NMEA-0183이 사용되고 있다. Fig. 15는 GPS수신 장비에 수신된 NMEA 형식 내용이다.

```
$GPGGA,141113.999,3730.0308,N,12655.2369,E,1,06,1.7,98.9,M,,0000*3E
$GPGSA,A,3,02,07,01,20,04,13,,,,,3.7,1.7,3.2*31
$GPRMC,141113.999,A,3730.0308,N,12655.2369,E,19.77,195.23,101200,,*3C
$GPGGA,141114.999,3730.0264,N,12655.2351,E,1,07,1.2,98.8,M,,0000*3C
$GPGSA,A,3,02,07,01,20,24,04,13,,,,,2.3,1.2,1.9*3E
$GPRMC,141114.999,A,3730.0264,N,12655.2351,E,15.51,202.12,101200,,*3C
$GPGGA,141115.999,3730.0231,N,12655.2345,E,1,07,1.2,98.7,M,,0000*37
$GPGSV,2,1,07,07,84,025,47,04,51,289,48,20,40,048,47,02,32,203,46*74
$GPGSV,2,2,07,01,23,101,47,13,20,131,32,24,19,268,40*49
```

Fig. 15 GPS receiving format.

Fig. 15처럼 신호가 각 항목마다 1초 내지 수초마다 한 번씩 텍스트 형태로 계속 수신된다.

기본 GPS 수신기들은 GPGGA, GPRMC, GPGSA, GPGSV의 네 항목을 지원하고 있으며, 본 연구에서는 GPRMC 항목을 사용하여 구현하였다.

GPRMC 항목은 GP와 RMC가 합쳐서 GPRMC라고 하는데 RMC는 Recommended Minimum Specific GNSS Data의 약자이며 네비게이션에 필요한 일반적인 항목을 모두 가지고 있다.

Table 5는 GPRMC 항목의 내용이다. Sentence ID 필드는 GPRMC항목

인지 아니면 위에서 언급을 한 GPGGA, GPGSA, GPGSV인지를 인식하는 부분이다. UTC Time는 시각을 나타내는 필드이며, UTC Date 필드와 같이 정확한 날짜와 시간을 얻을 수 있다. UTC Time은 세계 표준시이므로 우리나라에서는 9시간을 더해 주어야 한다. UTC Date는 날짜항목 필드이다. Status 필드는 현재 GPS 수신기 상태를 나타내면 여기에는 A와 V로 표시한다. A는 Valid한 상태이면 V는 Invalid한 상태를 나타내며, V인 경우는 위치정보를 신뢰할 수 없는 상태이다. Latitude 필드는 위도를, N/S Indicator는 북위와 남위를 나타내고 한국에서는 북위와 E/W Indicator에는 동경 밖에 볼 수가 없다. Longitude는 경도를 나타낸다. Speed over ground는 속도를 나타내는데 해양용이기 때문에 Knots단위를 사용한다. 따라서 차량 속도계로 변환시키기 위해서는 1.852를 곱하여 속도를 계산해야 한다. Course over ground는 진행 방향을 나타낸다. 진북을 중심으로 시계방향으로 0 ~ 359까지의 각도가 주어지며, 진북은 WGS-84 타원체를 중심으로 한 것이다. Magnetic variation은 자북과 Course over ground에서 말한 진북과의 차이를 각도로 표시한 것인데 고가의 GPS장비가 아니면 대개는 Null로 채워져 있다. 나머지 필드는 Checksum과 Terminator를 갖는다.

(2) 오차 범위 측정

일반적으로 알려진 GPS의 위치 정확도는 2DRMS(Distance Root Mean Square)로 대략 100m 정도이다. 그러나 미 국방성에서 2000년 5월에 인위적 오차 요인인 SA(Selective Availability)를 끄으로써 위치정확도는 2DRMS 20m 정도가 된다.

GPS 관련 오차는 전리층 지연 오차, 대류권 지연 오차, 다중경로오차, 위성궤도오차, 위성시계오차, 그리고 수신기 오차 등이 있다.

GPS의 오차는 여러 가지 요인으로 발생할 수 있는데, 본 연구에서는

GPS수신기 측정 오차를 말한다. 즉, 동일한 시간, 동일한 장소에서 GPS 수신기에서 수신한 차량 위치 좌표값의 변화를 측정하여 그것을 평균 편차(mean deviation)로 계산하여 GPS 오차 범위를 지정하였다. 또한 차량을 이동시켜 동일 시간이 아니라 시간에 관계없이 동일 장소에서 GPS수신기에 수신된 오차 범위를 측정하여 오차 범위를 지정하였다.

Table 5. GPRMC field composition.

Field	Example	Comments
Sentence ID	\$GPRMC	Recommended Minimum Specific GNSS Data
UTC Time	091342.862	hhmmss.sss
Status	A	A=Valid, V=Invalid
Latitude	3327.4064	ddmm.mmmm
N/S Indicator	N	N = North, S = South
Longitude	12633.8847	dddmm.mmmm
E/W Indicator	E	E = East, W = West
Speed over ground	10.85	Knots
Course over ground	336.50	Degrees
UTC Date	031005	DDMMYY
Magnetic variation		Degrees
Checksum	A*51	
Terminator	CR/LF	

본 연구에서 사용된 GPS장비는 NAVIUS사의 NSA-U3모델이며, 장소는 4층 건물 바로 옆에서 측정하였다. GPS 오차 종류에서도 있듯이 공기 중의 수증기, 기압, 온도에 따라 위치 좌표값이 다르게 수신된다. 따라서 GPS수신 변화가 큰 곳과 작은 곳을 배제하여, 어느 정도 건물이 있고 수신이 양호한 곳에서 GPS수신 오차를 실험하였다. 실험에 사용된 데이터 개수는 각각 100회를 측정하였으나, 본 연구에서는 Table 6처럼 20개의 일부 데이터로 간추렸다.

GPS수신기에서 수신한 차량 위치 좌표값을 측정한 결과, 경도의 평균

편차가 약 0.023초, 위도는 약 0.067초 정도의 평균 편차가 발생하였으며, 이를 거리로 환산하여 계산한 결과 약 2m 오차 범위가 발생하였다.

Table 6. Location coordinate value received by GPS.

	경도	위도
1	126°33 ' 54.18 "	33°27 ' 23.92 "
2	126°33 ' 54.18 "	33°27 ' 23.90 "
3	126°33 ' 54.10 "	33°27 ' 23.95 "
4	126°33 ' 54.19 "	33°27 ' 23.99 "
5	126°33 ' 54.19 "	33°27 ' 23.85 "
6	126°33 ' 54.19 "	33°27 ' 23.70 "
7	126°33 ' 54.22 "	33°27 ' 23.83 "
8	126°33 ' 54.20 "	33°27 ' 23.80 "
9	126°33 ' 54.17 "	33°27 ' 23.75 "
10	126°33 ' 54.25 "	33°27 ' 23.92 "
11	126°33 ' 54.24 "	33°27 ' 23.93 "
12	126°33 ' 54.18 "	33°27 ' 23.87 "
13	126°33 ' 54.22 "	33°27 ' 23.85 "
14	126°33 ' 54.19 "	33°27 ' 23.90 "
15	126°33 ' 54.17 "	33°27 ' 23.92 "
16	126°33 ' 54.22 "	33°27 ' 23.87 "
17	126°33 ' 54.21 "	33°27 ' 23.90 "
18	126°33 ' 54.18 "	33°27 ' 23.92 "
19	126°33 ' 54.20 "	33°27 ' 23.90 "
20	126°33 ' 54.19 "	33°27 ' 23.93 "

2) 필터링 요소 및 제거

필터링은 차량 GPS수신기에서 수신된 차량 위치 좌표값 중 중복되거나 차량 이동 경로에 무의미한 차량 위치 좌표값을 제거하는 것이다. 예를 들어, 차량이 이동 중 신호등에서 신호 대기하는 경우, 또는 차량이 출발하기 전 방향 전환을 위해 후진하는 경우, 중복과 무의미한 차량 위치 정보가 발생한다. 따라서 차량 위치 좌표값이 중복되는 것을 막기 위해, 첫 차량 위치 좌표값만 남기고 나머지 중복되는 차량 위치 좌표값은 제거

해야 한다.

필터링 요소는 차량 이동 중 신호 대기 중 수신된 위치 좌표값, 차량 이동을 위한 차량 후진 시 좌표값, 차량 시간·거리 제약 조건에 속해있는 이동 거리 좌표값, 차량 시간 제약 조건에 만족하는 정차 시 좌표값들은 전부 제거하였다. 이는 GPS 수신기는 매초마다 위치 좌표값이 수신되기 때문에 차량이 정지했을 때도 수신이 된다. 따라서 이런 차량 운행 경로에 불필요한 좌표값과 중복되는 좌표값들을 제거하여 차량 이력 데이터를 최소화시켰다.

Fig. 16은 차량이 신호 대기 상황에서 GPS 수신기를 통하여 차량 위치 좌표값들이 수신되고 있는 것을 보여주고 있다. Fig. 16에서 보여지는 바와 같이 정지한 동안 수신된 차량 위치 좌표값들은 똑같다. 따라서 중복되고 있는 좌표값을 바로 데이터베이스에 저장했을 경우 데이터베이스의 저장 공간을 낭비를 초래하기 때문에 시스템 구축비용의 증가를 초래한다.

좌표 수신 상태	NEWA 모니터링
GPS 좌표	119241 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119242 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119243 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119244 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119245 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119246 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119247 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119248 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119249 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119250 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119251 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119252 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119253 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119254 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119255 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119256 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119257 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119258 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119259 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119260 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119261 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119262 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119263 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119264 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119265 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119266 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119267 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119268 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119269 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119270 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119271 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119272 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119273 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119274 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75
GPS 좌표	119275 A 320T 3636.N 12833.96818 E 0.00 0003985 A=75

Fig. 16 location coordinates received while a vehicle stops.

3) 임계 영역

임계 영역은 차량 거리 제약 조건에 정의된 차량 최소 이동 거리와 GPS오차 범위를 합친 영역을 말한다. 이것은 차량의 유사궤적을 추출하거나 차량 위치 검색 질의 시 반드시 필요한 부분이다.

이동 중인 차량이 GPS 수신기를 통하여 수신된 차량 이동 경로의 이력 데이터를 획득하였을 때 동일한 위치 좌표값을 획득하기란 매우 힘들다. 그 이유는 서술한 GPS 수신 오차가 발생하기 때문이다. 또한 차량은 항상 이동하는 이동객체로서, 이동 시 동일한 속도로 운행하는 것이 아니라 계속적으로 속도 변화가 일어난다. 따라서 속도 변화에 따른 위치 좌표값과 수신 좌표값들이 서로 다르기 때문에 임계 영역을 지정하여 차량 이력 데이터 위치 좌표값이 동일하지 않더라도 수신된 위치 좌표값이 임계 영역 범위 안에 있으면 동일한 위치 좌표값으로 판단한다.

임계 영역 범위 지정을 위한 측정 요소들은 많지만, 본 연구에서는 도로 주행 제한속도로 임계 영역 범위를 지정하였다. 제한속도는 제주도로 특성상 차량 최대 속도는 80km/h로 제한되어 있다. 차량 속도 80km/h는 한 시간에 80km를 이동한다는 것이다. 따라서 이를 초당으로 환산하여 계산하면 약 22m가 나온다. 이를 위치 좌표값으로 변환했을 경우 위도는 0.71초, 경도는 0.85초의 차가 발생한다. 따라서 차량 이동 방향에서 따라 차량 위치 좌표값에 각각 0.71초와 0.85초를 가감(加減)해 주어야 한다.

4. 유사궤적 추출을 위한 차량 정보 표현

차량 이동객체에 대한 향후 이동 위치 검색을 위해서는 첫째, 이동객체가 동일한 행위 정보를 가진 상태로 이동 중일 때의 시작 위치를 나타내는 위치정보가 필요하다. 둘째, 차량 이동객체의 이동 습성을 나타내는 행위 정보가 필요하며, 셋째, 위치 계산을 위한 이동 함수에 관한 정보가 필요하다. 이 절에서는 차량 위치정보 및 행위 정보의 표현 방법에 대해 기술하였다.

1) 차량 이동객체의 위치정보 표현

본 연구에서 제시하는 차량 이동객체 추적의 대상이 되는 차량 이동객체는 이동점 방식으로 정의하였으며, 이동점 객체의 위치정보 표현 방법에 대해 서술하였다.

향후 차량 이동 위치 검색 질의는 설정된 위치정보에 행위 정보를 매개 변수로 사용하는 이동 함수를 적용하여 수행된다. 이동 함수는 현재까지 입력된 이동 정보를 기반으로 객체의 행위 정보와 향후 이동 위치를 계산하기 위한 함수들이다.

먼저, 이동 함수에 대해 살펴보자. 시간 t_1 과 t_2 사이에서 차량 이동객체가 (x_1, y_1) 에서 (x_2, y_2) 로 이동하였다고 하면 차량 이동 거리와 이동 방향 θ 는 Fig. 17과 같이 표현할 수 있다.

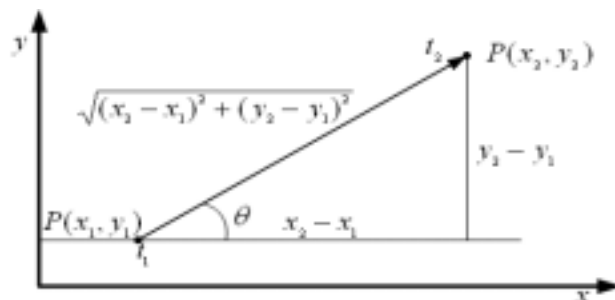


Fig. 17 Location change of moving object between t_1 and t_2

Fig. 17에서 삼각함수와 삼각형의 정의를 이용하여 차량 이동객체의 이동 방향을 구할 수 있다. 다음의 식 (7)에서 각 θ 는 동쪽을 기준으로 하는 이동 방향이 된다.

$$\text{Sin}\theta = \frac{y_2 - y_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}} \quad (7)$$

차량 행위 정보에서 차량 이동객체의 시작 위치를 구할 수 있고, 차량 이동 정보 입력은 차량 이동객체의 현재 위치를 포함하므로 위의 식(7)에 의해 이동 방향을 구할 수 있다. 각 θ 의 범위가 0° 에서 360° 라고 할 때 $\text{Sin}\theta$ 의 값은 Sin 그래프에서 볼 수 있듯이 두 개가 나오므로 (x_1, y_1) 과 (x_2, y_2) 의 좌표를 검사하여 (x_1, y_1) 을 기준으로 (x_2, y_2) 가 몇 사분면에 존재하는지를 파악해야 한다.

차량 이동객체의 이동 속도는 행위 정보에 설정된 이동 속도를 새로 입력된 이동 정보의 이동 속도와 비교한다. 속도 변화가 정의된 오차의 범위 이내이면 명시적 수정 연산 없이 이동 정보를 저장한다. 두 점 사이의 이동 거리 D 는 다음 식 (8)을 이용하여 구할 수 있다.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (8)$$

이동 속도는 t_1 시간부터 t_2 시간까지 이동한 거리 D 를 시간 $|t_2 - t_1|$ 으로 나누면 구할 수 있다. 그러므로 이동 속도 S 는 식 (8)과 식 (9)으로부터 구할 수 있다.

$$S = \frac{D}{|t_2 - t_1|} \quad (9)$$

식 (7)과 식 (9)에 의해 시공간 이동객체의 이동 방향과 속도가 계산된다. 이를 기반으로 차량 이동객체의 이동 위치를 검색하기 위해 원의 방정식을 이용하였다.

Fig. 18에서 $O(x_1, y_1)$ 은 동일한 행위 정보를 가지는 차량 이동객체의 시작 위치이고, $P(x_2, y_2)$ 는 검색하려는 위치가 된다. 주어진 시간 이후에 이동 가능한 객체의 위치는 식 (8)에서 구해진 거리 D 와 시간을 곱한 거리를 반지름으로 하는 원이 된다. 이 중에서 식 (7)에서 구해진 이동 방향 θ 에 해당되는 지점을 구하면 객체의 향후 이동 위치를 구할 수 있다.



Fig. 18 Circle equation.

원의 방정식에서 중점으로부터 주어진 각 θ 와 반지름 D 을 가지는 점 $P(x_2, y_2)$ 의 좌표는 식 (10)과 (11)에 의해 구할 수 있다. 여기서 반지름 D 는 식 (8)의 거리 D 와 동일하다.

$$x_2 = D \cos \theta \quad (10)$$

$$y_2 = D \sin \theta \quad (11)$$

지금까지 설명한 방법으로 차량 이동객체의 향후 위치를 검색할 수 있다. 하지만 이러한 시간 종속적 함수의 정확성을 높이기 위해서는 행위 정보를 유지하는 것이 필요하다.

2) 차량 이동객체의 행위 정보 표현

본 연구에서는 차량 이동객체의 행위 정보로 속도와 방향에 관한 정보를 사용한다. 그리고 시공간 이동객체는 시간 t 가 t_0 에서 t_4 로 흐르는 동안 객체의 위치 P 가 P_0 에서 P_4 로 이동하며, 시간의 단위는 차량 이력 데이터베이스에 저장된 시간으로 한다.

(1) 속도 정보의 표현

속도 정보는 이동 정보의 입력 시에 계산된 속도를 검사하여 행위 정보에 기록된 속도 정보와의 차이가 주어진 오차의 범위를 넘을 경우 명시적 갱신 연산을 수행한다. 본 연구에서는 속도 변화를 판단하는 방법으로 다음의 두 가지 방법을 사용한다.

첫 번째 방법은 속도의 평균을 구하여 속도 변화를 판단하는 방법이다.

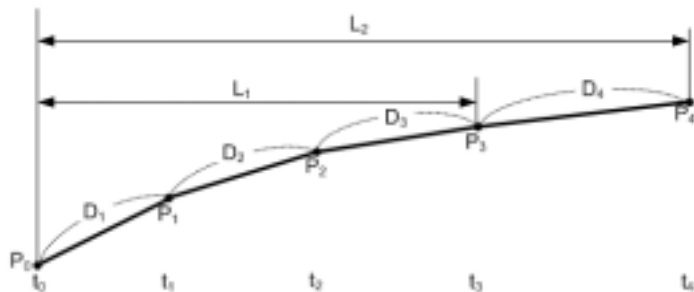


Fig. 19 Average speed.

Fig. 19에서 n 번째 이동 정보인 P_n 의 평균 이동 속도 S_n 은 다음 식 (12)로 구할 수 있다.

$$S_n = \frac{\sum_{i=1}^n D_i}{t_n - t_0} \quad (12)$$

행위 정보의 변경은 이동 정보 입력 시점에서의 평균 속도 S_n 과 행위 정보에 기록된 속도 정보 사이의 차이가 주어진 오차의 범위를 벗어나면 수행되며, 행위 정보가 변경된 경우에는 그 시점의 평균 이동 속도로 행위 정보를 갱신한다.

두 번째 방법은 입력 단계별 속도를 파악하여 속도 변화를 판단하는 방법이다.

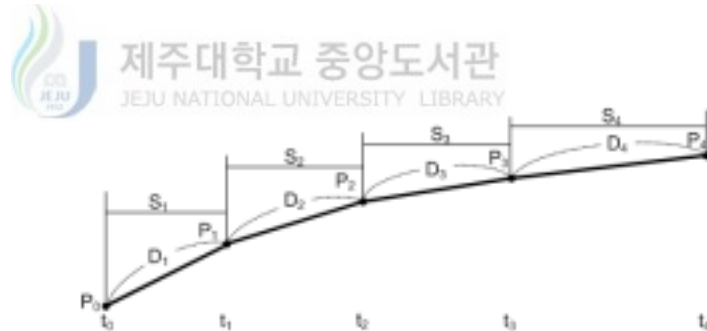


Fig. 20 The speed to input steps.

Fig. 20과 같이 행위 정보의 변경은 행위 정보에 기록된 속도 정보와 현재 속도 사이의 차이가 주어진 오차의 범위를 넘어서는 경우 수행된다. 예를 들어 P_4 의 입력에서 이동 속도의 변경 여부를 파악하는 것은 P_4 의 이동 속도 S_4 와 행위 정보에 기록된 속도 정보 사이의 차이를 비교하는 것이다. Fig. 20의 경우 시간의 단위는 모든 동일하므로 이동 거리 D_4 가 P_4 의 속도가 되며, P_4 의 입력에서 행위 정보가 변경된 경우, 행위 정보

의 속도 정보에는 S_4 가 설정된다.

앞의 두 가지 방법을 비교해 보면, 첫 번째 방법은 이동 정보의 속도 평균을 구하여 속도 정보의 변화를 파악하므로 간헐적인 속도 변화에 쉽게 영향을 받지 않는다. 즉, 속도의 변화가 전체 이동 정보로 분산되어 계속적인 속도 변화가 발생하거나 전체 평균 속도에 영향을 줄만큼 큰 차이의 속도 변화가 발생해야 행위 정보의 변경이 수행된다. 반면에 두 번째 방법은 입력된 이동 정보의 이동 속도만으로 속도 정보의 변화를 파악하므로 속도의 변화에 민감하여 첫 번째 방법에 비해 빈번한 갱신이 발생한다.

5. 차량 이동 시퀀스 추출



차량 이동 시퀀스(Sequence)는 유사계적 추출 방법으로 추출된 차량 이동점의 시작점과 끝나는 점을 하나의 이동 구간으로 만들어 차량 이동 경로를 시퀀스로 생성 저장한다. 이는 차량 이동 패턴 마이닝에 쓰이는 하나의 트랜잭션을 만드는 것이다.

차량 이동 패턴 탐사는 차량의 미래의 위치를 추정할 때 중요한 역할을 한다. 예를 들어, 한 차량 이동객체 MP 이 주기적으로 출발지에서부터 A, B, C, D 구간으로 목적지에 도착하는 차량이다. 이러한 경우 출발지에서 출발하여 A구간을 지나 B구간을 운행 중인 차량의 미래의 위치를 검색했을 경우 MP 의 이동 빈발 패턴을 검색하여 C구간으로 갈 것을 예측 할 수 있을 것이다. 또한 MP 이 A, B, C구간을 지나 D구간으로 이동하지 않고 다른 경로 E를 선택했을 경우 MP 의 이동 경로는 A, B, C, E로 변

경되는 것이다. 이렇듯 차량 이동객체의 이동 시퀀스가 달라진다. 또한 차량 이동객체 MP 이 A, B, C, D 구간을 순서적으로 이동했다고 가정하자. 시간을 고려하지 않을 경우, 이 네 구간은 차량 이동객체 MP 의 이동 경로, 즉 $A \rightarrow B \rightarrow C \rightarrow D$ 가 될 수 있다. 그러나 C구간에서 정차하여 한 시간이 지나서 D구간으로 이동했다면, 차량 이동객체 MP 의 이동 경로가 여전히 $A \rightarrow B \rightarrow C \rightarrow D$ 라고 보기는 어렵다. 다시 말하면, 차량 이동객체 MP 의 이동 경로는 $A \rightarrow B \rightarrow C$ 와 D, 두 개의 시퀀스로 나뉘어야 한다. 이렇게 의미 있는 이동 시퀀스를 생성하기 위해서 시퀀스를 구성하는 지역 간에 최대 시간 제약조건을 두어서 두 위치 사이의 시간이 최대 시간 제약조건 내에 있을 때만 하나의 시퀀스로 생성하도록 해야 한다.

따라서 이 과정에서는 객체의 공간 속성에 대한 유효시간을 검사해서 특정 위치에 머무른 시간이 max_gap 을 초과하면 초과하기 이전의 이동 시퀀스와 이후의 이동 시퀀스로 분리된다.

Table 7은 데이터베이스에서 각 차량 이동객체의 이동 시퀀스를 추출한 것이다. 이 예에서는 max_gap 이 30분이라고 가정한다. 여기서 차량 이동객체 3의 세 번째와 네 번째 위치 사이에 시간 간격이 max_gap 인 30분을 초과하므로 두 개의 이동 시퀀스로 분리된다.

Table 7. Moving sequence.

	Moving Sequences
1	<A B C D>
2	<A B C> <i>oid</i>
3	<B C D>, <F>

Fig. 21은 차량 운행 경로를 각각의 구간으로 나누어서 저장하여 이동

시퀀스를 작성한 화면이다. S_x , S_y 는 시작 위치 좌표값이고, E_x , E_y 는 끝나는 위치 좌표값이다. 그리고 $Lang$ 은 이동 거리를 나타내며, Ang 는 방향각도, $Speed$ 는 속도를 말한다.

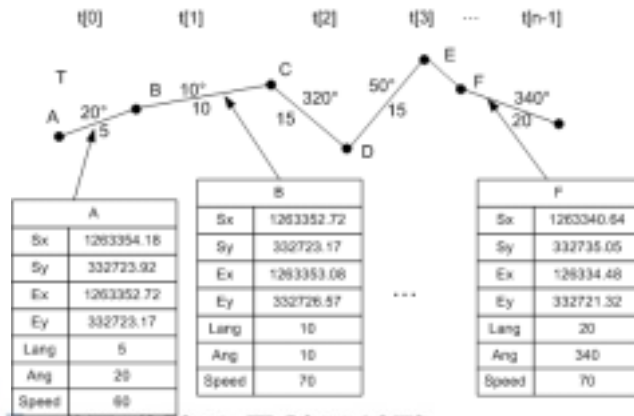


Fig. 21 Attribute informations of vehicle moving object running sections.

Fig. 22에서 제시한 알고리즘은 정렬된 차량 이동객체 데이터 집합에 대하여 구간 공간 개념 계층과 임계치 등을 적용하여 차량 이동객체에 대한 단위 이동 시퀀스를 추출해 내는 알고리즘 `MovingSequenceExtract()`의 일련의 과정을 가상코드로 기술한 것이다. 입력된 차량 이동객체 데이터 집합 S_{MP} 의 각 차량 이동 시퀀스에 대하여 시간과 거리 임계치인 min_dist , max_gap 을 만족하도록 단위 이동 시퀀스를 형성하며 이때 최소 거리 임계치 적용시 경계를 고려한 단위 이동 시퀀스를 형성한다. 마지막 단계에서는 `SCGeneralize()` 연산을 통해 이동 구간 개념 계층인 SCH 에 대하여 요구되는 일반화 임계치인 $gen_threshold$ 수준의 개념 값으로 각 단위 이동 시퀀스의 이동 변화를 일반화한다.

Algorithm : MovingSequenceExtract();

Input :

S_{MP} : 차량 이동객체 데이터 집합

MAP : 배경 지도 데이터 집합

SCH : 이동 구간 개념 계층

$gen_threshold$: 이동 구간 개념 일반화 수준 임계치

min_dist : 최소 거리 임계치

max_gap : 최대 시간 간격 임계치

Output :

MS : 일반화된 이동 시퀀스 집합

Method :

MS empty;

// 각 차량 이동객체의 이동 시퀀스에 대하여

FOREACH(s IN S_{MP})

{

$usid$ 1; // 단위 이동 시퀀스 식별자 초기화

$PreLC$ 1; // 이전의 이동 인덱스 초기화

 // 새로운 단위 시퀀스 생성

sid = getSid(s);

US = new US (sid , $usid$);

US = s [1];

 // 시퀀스 분리 단계

 FOR (i = 2; i < s .length; i ++) {

 // 최대 시간 간격 임계치 적용

 IF(s [i -1], s [i] > max_gap) {

MS = MS + US ;

 // 시퀀스 분리 및 새로운 단위 이동 시퀀스 생성

$usid$ = getNextUsid();

US = new US (sid , $usid$);

US = s [i];

$PreLC$ = i ;

 }

 ELSE {

 // 최소 거리 임계치 적용

 IF($dist$ (s [$PreLC$], s [i] > min_dist) {

US = US + s [i];

$PreLC$ = i ;

 }


```

ELSE {
    //구간 경계를 고려한 시퀀스 요약
    PreArea = GeneralizeIntoRegion
                (S[ PreLC], MAP);
    CurArea = GeneralizeIntoRegion(S[i], MAP);
    IF( PreArea <> CurArea ) {
        US = US + s[i];
        PreLC = i;
    }
}
}
}
MS = MS + US;
}
// 이동 구간 개념 계층을 적용한 일반화 수행
MS SCGeneralize( MS, SCH, gen_threshold ) ;
return MS;

```

Fig. 22 Moving sequence abstraction algorithm.

6. 빈발 이동 패턴 마이닝

빈발 이동 패턴 마이닝은 이동 시퀀스로부터 사용자가 지정한 임계치를 초과하는 빈발한 차량 이동 패턴을 마이닝하는 것이다. 알고리즘은 효율적으로 후보 집합을 줄일 수 있는 대표적인 연관 규칙 알고리즘인 Apriori(R. Agrawal, 1994),(R. Srikant, 1994) 알고리즘을 변형하여 사용하였다.

Apriori 알고리즘과 같이 알고리즘의 기본적인 구조는 데이터에 대해서 여러 번의 스캔 과정을 거친다. 각 패스에서 후보 시퀀스를 생성하고 데

이터베이스에서 후보 집합을 포함하는 차량 이동 시퀀스의 개수를 계산한다. 각 패스에서 생성된 빈발 이동 시퀀스는 다음 패스의 후보시퀀스를 만들기 위해 사용된다. 알고리즘은 패스가 끝난 후 더 이상 빈발 이동 시퀀스가 없거나 생성될 후보 집합이 없으면 끝난다.

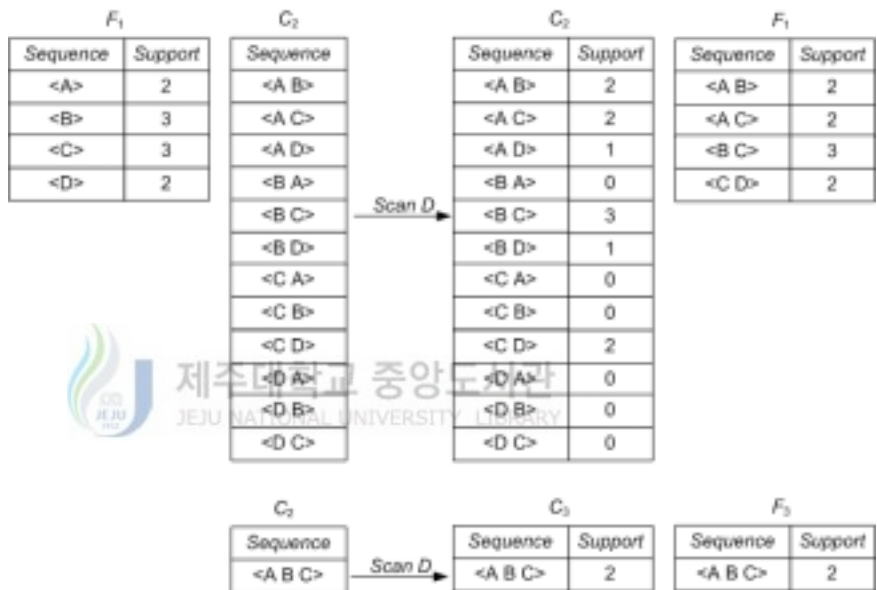


Fig. 23 Candidate sequence creation example.

F_k 는 빈발 k -시퀀스를 표현하고 C_k 는 후보 k -시퀀스를 표현한다. C_k 는 F_{k-1} 의 셀프 조인, 즉 $F_{k-1} * F_{k-1}$ 을 통해 생성된다. 이 때, F_{k-1} 에 존재하는 각각의 이동 시퀀스 s_1, \dots, s_{k-1} 과 s'_1, \dots, s'_{k-1} 가 존재할 때, s'_1, \dots, s'_{k-1} 가 s_1, \dots, s_{k-2} 을 포함하거나, s_1, \dots, s_{k-1} 가 s'_1, \dots, s'_{k-2} 를 포함할 때 조인이 이루어진다. 그런 다음, C_k 내의 시퀀스에 대하여 F_{k-1} 에 포함되지 않은 부분 시퀀스를 포함하는 시퀀스는 삭제된다. 이 과정은 빈발하지 않은 집합들의 상위 집합들은 빈발하지 않다

는 관찰에서 수행된다.

후보 집합을 저장하고 이동 시퀀스에 해당되는 후보 집합이 있는지 효율적으로 검색하기 위해 해쉬 트리를 사용한다. 해쉬 트리는 해쉬 테이블과 항목집합을 포함하는 내부노드(internal node)와 단말노드(leaf node)로 구성된다. 내부노드의 해쉬 테이블의 각 버킷은 자식 노드를 가리키는 데, 깊이 d 의 내부 노드는 깊이 $d+1$ 의 노드를 가리킨다. 후보 항목집합을 해쉬 트리에 추가할 때 루트로부터 단말노드에 이르기까지 깊이를 1씩 증가시키면서 트리에 저장한다. 깊이 d 의 내부 노드에서 후보 항목집합의 d 번째 항목을 해쉬 함수에 적용하여 자식 노드를 결정한다.

min_sup 이 2개의 시퀀스라고 가정했을 때, Table 7의 이동 시퀀스로부터 추출된 빈발 이동 패턴은 $\{ \langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle, \langle A B \rangle, \langle A C \rangle, \langle B C \rangle, \langle C D \rangle, \langle A B C \rangle \}$ 이 되며, Fig. 23은 이 과정을 보여준다.

지금까지 차량 이동 패턴 마이닝을 위한 수행 과정을 살펴보았다. Fig. 24는 이러한 각 단계별 알고리즘 VMP(Vehicle Moving Pattern mining)의 전체 구성을 보여준다.

```
VMP(database D, max_gap, min_sup, Area A)
begin
  foreach moving objects mpoint in D do
    foreach location x, y of mpoint do
      foreach area a in the A do
        if (contains(mpoint, a))
          set location of mpoint as a
          break;
      for (i=1; end of a mpoint; i++) do
        begin
          if  $vt_i - vt_{i-1} > max\_gap$ 
            insert with new sequence id
        end
      end
    end
  end
```

```

F1 = {large 1-sequence};
for(k=2; Fk-1 ≠ ∅; k++)do
  begin
    Ck = New candidates generated from Lk-1
    foreach moving-sequence c in D do
      increment the count of all candidates in Ck that are contained in c
    Fk = Candidates in Ck with min_sup
  end
  Answer = Frequent Sequences in UkFk
end

```

Fig. 24 Moving pattern mining algorithm VMP's whole composition.

7. 유사궤적 데이터베이스 설계



제한한 차량 유사궤적 위치정보를 저장하기 위해서 크게 차량 속성 정보를 저장하는 차량 속성 테이블, 운전자 속성을 반영한 운전자 속성 테이블, 차량 이동 경로를 저장하는 차량 이력 테이블, 차량 이력 데이터를 바탕으로 한 시퀀스 테이블, 빈발 시퀀스 테이블로 구성된다.

1) 차량 속성 테이블

차량 속성 테이블은 차량에 대한 속성 정보를 저장하는 테이블이다. 차량 속성 정보는 운행 목적에 따라 시퀀스 추출에 중요한 요인이 될 수 있다. 예를 들면 출·퇴근 목적으로 차량을 운행하는 차량이라면 시퀀스는 크게 출근 시퀀스와 퇴근 시퀀스만 생각할 수 있다. 하지만 영업용 목적으로 차량을 운행한다면 차량의 시퀀스는 여러 가지 시퀀스가 나올 수 있다. 따라서 차량 속성 정보를 저장 할 수 있는 테이블이 필요하다. 차량

속성 정보는 Table 8과 같이 구성된다.

Table 8. Vehicle attribute table.

필드명	Type	의미
vmo_id	String	차량 ID
vmo_user	String	운전자 이름
vmo_user_no	String	운전자 주민번호
vmo_vol	String	차량의 용량
vmo_op_purpose	String	차량 운행 목적
vmo_kind	String	차종

2) 운전자 속성 테이블

운전자 속성 테이블 차량 이동객체를 운전하는 사람 즉, 운전자에 따라 차량을 운전하는 패턴이 다르다. 예를 들면, 차량을 출발 시킬 때 급가속을 하는 사람이 있는 반면 서서히 출발하는 사람도 있다. 따라서 가족 내에서도 주로 차량을 운전하는 사람 있고, 가끔 차량을 운행하는 사람이 있을 것이다. 이런 운전 패턴이 다른 사람들이 한 차량을 운전을 했을 경우 서로 판이한 운전 패턴을 갖게 된다. 운전자 속성 테이블은 이러한 운전자 속성 정보를 저장함으로써 주 운전자와 보조운전자의 패턴을 따로 저장하여 이동 패턴을 추출할 때 보다 구체적인 데이터를 추출할 수 있도록 하기 위해서이다. 테이블 구성은 Table 9와 같이 구성이 된다.

Table 9. Driver attribute table.

필드명	Type	의미
vmo_id	String	차량ID
user_age	number	주 운전자 나이
user_sex	String	주 운전자 성별
user_Office_addr	String	주 운전자 직장 주소
user_Home_addr	String	주 운전자 집 주소

3) 차량 이력 테이블

차량 속성 테이블은 시간 변화에 따른 차량 이동객체의 ID와 차량 이동 좌표값을 저장하고, 또한 차량 이동 좌표값의 변화량을 저장하는 테이블이다. 테이블 구성은 Table 10과 같다.

Table 10. Vehicle history table.

필드명	Type	의미
vmo_id	String	차량 ID
mo_date	Date	차량 운행 시간
vmo_x	double	차량 x좌표
vmo_y	double	차량 y좌표

4) 차량 이동 시퀀스 테이블

차량 이동 시퀀스 테이블은 차량 이동 시퀀스 추출방법에 의해서 추출한 것으로 Table 11과 같은 구조를 갖는다.

Table 11. Vehicle moving sequence table.

필드명	Type	의미
seq_no	number	시퀀스 번호
s_x	double	시작 경도 좌표
s_y	double	시작 위도 좌표
e_x	double	종료 경도 좌표
e_y	double	종료 위도 좌표
lang	number	거리
ang	double	각도
speed	number	속도

5) 차량 빈발 시퀀스 테이블

차량 빈발 시퀀스 테이블은 차량이 이동하기 시작했을 때부터 끝날 때까지 차량 이동 시퀀스 테이블에 들어있는 시퀀스를 몇 번 운행했는지를


저장한다.

이것은 차량의 다음 구간을 판단할 때 중요한 요소가 된다. 예를 들면 차량 이동객체가 A, B, C 경로로 운행 중이라면 다음 운행 구간의 시퀀스를 예측할 수 있다는 것이다.

빈발 시퀀스 테이블 구조는 Table 12와 같다.

Table 12. Frequent sequence table.

필드명	Type	의미
vmo_id	string	차량ID
seq_no	number	시퀀스 번호
tf	number	빈발개수



제주대학교 중앙도서관
JEJU UNIVERSITY LIBRARY

8. 차량 위치정보 불확실성 처리

차량 유사궤적 표현기법은 방향각이 급격히 변경되는 시점을 기준으로 운행 구간을 저장하는 기법이므로, 시작 위치정보와 종료 위치정보를 갖고 있기 때문에 차량이 운행 중 특정 구간에 있을 시, 또는 데이터베이스에 저장이 되지 않은 위치에 있는 차량을 검색하기 위해 차량의 불확실한 위치를 검색할 필요가 있다. 본 절에서는 차량 위치정보 불확실성처리에 대하여 기술하였다.

1) 불확실성 모형

(1) 생성 유형

차량 위치정보 불확실성(Uncertainty)이란 임의의 차량 이동객체에 대하여 원하는 시간이 주어진 경우 여러 가지 이유로 대응하는 공간 위치데이터를 데이터베이스로부터 검색하는지 못하는 경우를 방지하고, 동시에 이를 위해 요구되는 차량 이동객체 위치데이터의 분량을 최소화하기 위한 기술을 의미한다.

입력 질의에 대한 결과로서 반환되는 차량 이동객체의 위치 정확도를 향상시키기 위해서는 보다 빈번하게 데이터를 저장하여야 한다. 실제 응용에서 차량 대수가 증가할수록 그 규모가 급격히 방대해지는 특성으로 인해 불확실성 처리를 제대로 지원하지 못하고 역으로 데이터베이스에 저장된 임의의 시간을 기준으로 질의하는 형태로 운용되고 있다.

차량의 위치에 관련된 불확실성을 유발하는 원인에는 ① 차량 위치정보, ② 통신 오류 또는 DB오류로 인한 유형, ③ 도로 및 교통정보 오류로 인한 유형, ④ 응용 분야 정보 오류로 인한 유형 등 크게 네 가지로 분류할 수 있다.

차량 위치정보 수집 방법에 따른 유형은 이산적인 모델에 내재된 오류로서 차량 이동객체 관리 시스템(MOMS)의 성능과 비용에 막대한 영향을 미치는 중요한 요소이다. 차량 이동객체의 유형에 따라서 발생하는 위치정보의 오류 범위는 매우 다양하며, 차량 이동객체 위치정보 측위 장비의 정밀도와 통신망 속도 등과 같이 다양한 형태의 고려사항이 존재한다.

차량 위치정보를 유무선 통신망을 통하여 시스템의 데이터베이스에 저장하려면 수많은 통신 절차를 수행한다. 특히 무선 통신망의 경우는 주위 환경의 동적인 변화에 따른 영향으로 속도와 안정성에서 차량 이동객체의 위치정보에 대한 많은 불확실성을 야기 시킨다. 또한 관련 시스템의 표준화가 이루어지지 않는 환경에서는 개별적인 시스템의 데이터베이스에 저장된 차량 위치정보는 상호 호환성 관점에서 문제가 될 수 있다.

도로 및 교통정보는 차량에 관련된 직접적인 정보로서 매우 밀접한 관련성을 가지며, 차량 위치정보를 위한 부가적인 데이터로서 활용될 수 있다. 일례로 일반통행여부, 제한속도, 차선 등의 도로의 상세한 정보와 시간대별 교통량 통계 값을 들 수 있다. 그러나 간혹 이들 정보의 부정확성으로 인해 오히려 차량 위치정보의 불확실성을 증가시킬 수 있다.

(2) 질의 영역

차량 관리 기술에는 시간지원 데이터베이스(Temporal Databases) 개념을 반드시 포함하기 때문에 질의 형태를 크게 현재 시점을 기준으로 ① 과거의 위치정보 불확실성 질의(Retro-active Uncertainty)와 ② 가까운 미래의 위치정보 예측 불확실성 질의(Pro-active Uncertainty Query)로 구분할 수 있다.

과거의 차량 위치정보 불확실성 질의는 데이터베이스 시스템에 저장된 차량 위치정보로부터 과거의 임의의 시간의 위치정보를 추정하는 문제이다. 이 경우, 시공간 데이터베이스에 해당 차량에 대하여 최소한 두 개 이상의 위치정보가 저장되어 있으며, 두 지점 사이의 임의의 위치 값을 추정하는 문제로 귀결한다.

가까운 미래의 차량 위치정보 예측 불확실성 질의는 최대 허용 예측 시간범위 내의 미래에 대하여 최근에 획득된 차량 위치정보를 토대로 추정하는 문제이다.

(3) 불확실성 처리 모형

차량의 위치는 실시간으로 변경되기 때문에 모든 시간에서 검출된 위치를 데이터베이스에 저장하는 것은 불가능하다. 따라서 차량의 위치는 특정 시간 구간을 갖고 이산적으로 데이터베이스에 저장된다. 하지만 이러한 데이터베이스는 저장되지 않은 시점의 차량 위치에 대한 질의에 대한 응답이 불가능하기 때문에 이에 대한 적절한 처리 방법과 데이터베이

스에 저장되지 않은 차량의 불확실한 이동 위치정보에 대한 적절한 처리 방법이 필요하다.

차량의 불확실한 위치 추정은 다음과 같은 두 가지 위치 추정 모델을 사용한다. 첫 번째 모델은 선형 함수를 이용한 위치 추정 모델이다. 선형 함수는 가장 단순한 1차 함수를 사용하므로 최소한의 연산으로 위치 추정 값을 구할 수 있는 장점을 갖는다. 두 번째 모델은 곡선을 표현하는 3차 스플라인 함수(Spline function)를 이용한 위치 추정 모델이다. 3차 스플라인 함수를 이용할 경우 일반적으로 선형 함수에 의한 위치 추정 결과보다 정확한 값을 얻을 수 있으나 3차 함수를 구하기 위한 연산 처리 과정이 복잡하여 처리 속도가 지연되는 단점을 갖는다.

선형 함수는 차량의 이동 이력에 저장된 임의의 시간 간격의 위치 좌표를 이용하여 1차 함수를 구하고 이 함수를 이용하여 임의의 시점에 대한 위치 추정 값을 구한다. 선형 함수를 이용한 위치 추정값의 근사는 차량의 이동 궤적을 직선 형태의 선형 궤적으로 모델링함을 의미한다.

선형 궤적 모델의 과거 위치 추정 연산은 식 (13)을 사용한다. 과거 위치 추정 연산은 특정 구간의 시작 시간, 시작 시간의 X, Y좌표 및 종료 시간의 X, Y좌표를 사용하여 시간 구간의 변화 함수를 구하고 이 변화 함수에 질의 시점을 입력하여 질의 시점의 위치 좌표를 추정한다. 여기서 X, Y좌표는 각각 차량의 검출된 위치 값으로서 시스템에 인식되어지는 좌표체계상의 값을 의미한다.

$$x_{t_p} \leftarrow \frac{x_{t_{i+1}} - x_{t_i}}{t_{i+1} - t_i} (t_p - t_{i+1}) + x_{t_{i+1}}$$

(13)

$$y_{t_p} \leftarrow \frac{y_{t_{i+1}} - y_{t_i}}{t_{i+1} - t_i} (t_p - t_{i+1}) + y_{t_{i+1}}$$

예를 들어, “차량의 ID가 7377인 차량 이동객체의 유효 시간 23에서의 위치 값을 구하십시오” (단, 질의 구간의 시작 시점은 20, 이때의 위치 좌표는 (125, 350)이고 종료 시점은 25, 이때의 위치 좌표는 (145, 360)이다.) 라는 질의를 보면 질의 시점은 23이 되고 질의 시점을 포함하는 최소 시간 구간은 [20, 25]가 된다. 과거 위치 추정 함수에 적용되는 위치정보 값은 (20, 125, 350)과 (25, 145, 360)이 된다. 식 (13)에 의해 추정된 위치 값은 (137, 356)이 된다.

(4) 과거의 위치 추정

과거의 차량 위치 추정은 식 (14)에서 제시된 바와 같이 차량 이력데이터가 차량 이력 데이터베이스에 저장된 시점이 $\{t_0, t_1, t_2, \dots, t_{i-1}, t_i, t_{i+1}, \dots, t_{now-1}, t_{now}\}$ 일 때, $t_i < t_p < t_{i+1}$ 의 조건을 만족하는 시점 t_p 에 대해 (t_i, x_{t_i}) 와 $(t_{i+1}, x_{t_{i+1}})$ 의 위치좌표 쌍을 이용하여 $x(t_p)$ 함수를 구하고, (t_i, y_{t_i}) 와 $(t_{i+1}, y_{t_{i+1}})$ 의 위치좌표 쌍을 이용하여 $y(t_p)$ 함수를 구한다. 이 때 함수 $x(t_p)$ 와 $y(t_p)$ 를 이용하여 과거의 시점 t_p 의 위치 값을 추정한다.

Fig. 25의 알고리즘 *past_location*은 차량 이동객체 *MP*의 *umo_id*와 과거의 임의의 시점 t_p 를 입력 값으로 받는다. 이 후 차량 객체의 속성 정보가 저장된 *v_history_data* 테이블에서 입력된 *umo_id*을 가지는 객체를 검색한다. 검색된 객체가 존재할 경우, 차량 이력 정보가 저장된 *v_history_data* 테이블에서 t_p 와 가장 인접한 이전의 시점 t_i 의 (x, y) 좌표 쌍 $(t_i, x_{t_i}), (t_i, y_{t_i})$ 를 검색한다. 그리고 t_p 와 가장 인접한 이후의

시점 t_{i+1} 의 (x, y) 좌표 쌍 $(t_{i+1}, x_{t_{i+1}}), (t_{i+1}, y_{t_{i+1}})$ 를 검색한다. 검색된 위치 좌표를 이용하여 x_{t_p} 와 y_{t_p} 값을 구하여 결과를 반환한다.

Algorithm past_location (vmo_id, t_p)

입력 => vmo_id 임의의 차량 이동객체 MP 의 식별자, t_p : 과거의 특정 시점

출력 => x_{t_p} : 시점 t_p 시 MP 의 x 좌표값, y_{t_p} : 시점 t_p 시 MP 의 y 좌표값

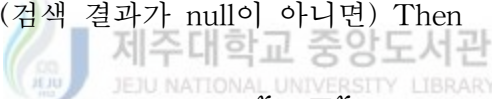
Begin

v_history_data에서 입력된 vmo_id 를 가지는 객체 검색

If(검색 결과가 null이 아니면) Then

v_history_data에서 $t_i < t_p < t_{i+1}$ 의 조건을 만족하는 위치 좌표 쌍 $(t_i, x_{t_i}), (t_i, y_{t_i})$ 와 $(t_{i+1}, x_{t_{i+1}}), (t_{i+1}, y_{t_{i+1}})$ 을 검색

If (검색 결과가 null이 아니면) Then



$$x_{t_p} \leftarrow \frac{x_{t_{i+1}} - x_{t_i}}{t_{i+1} - t_i} (t_p - t_{i+1}) + x_{t_{i+1}}$$

$$y_{t_p} \leftarrow \frac{y_{t_{i+1}} - y_{t_i}}{t_{i+1} - t_i} (t_p - t_{i+1}) + y_{t_{i+1}}$$

Else $x_{t_p} \leftarrow$ 오류 값, $y_{t_p} \leftarrow$ 오류 값

Return x_{t_p} , and y_{t_p} //과거의 t_p 시점의 (x, y) 좌표값 반환

End

Fig. 25 Past location estimation algorithm.

(5) 미래의 위치 추정

차량의 특정 시간의 미래의 위치를 추정하기 위해서 미래의 위치 변화 함수를 이용해야 하며, 미래의 위치 변화 함수를 생성하기 위해 과거 차량 이력 정보의 평균 이동 정보를 활용한 모델을 제시한다. 입력된 미래의 임의의 시점 t_f 가 $t_{now} < t_f$ 의 조건을 만족할 경우, t_{now} 이전의 N 개의

이력 위치 좌표를 검색하여 함수 $x(t_f)$ 와 $y(t_f)$ 을 구하여 미래의 시점 t_f 의 위치값을 추정한다.

```

Algorithm future_location(vmo_id, t_f)
입력 => vmo_id 임의의 이동객체 MP의 식별자, t_f: 미래의 특정 시점
출력 => x_{t_f}: 시점 t_f시 MP의 x좌표값, y_{t_f}: 시점 t_f시 MP의 y좌표값
N : 미래의 위치 좌표 추정에 사용될 최근 이동정보의 위치 좌표 쌍의 개수
Begin
  v_history_data에서 입력된 vmo_id를 가지는 객체 검색
  If(검색 결과가 null이 아니면) Then
    v_history_data에서 t_{now} 이전의 최근 N개의 위치 좌표 쌍을 검색
    If(검색 결과가 null이 아니면) Then
      검색된 위치 좌표를 임시 배열 x_i, y_i, t_i에 저장(단, i = 1 ~ N)
      x_{t_f} ← ( ∑_{i=2}^N (x_{t_i} - x_{t_{i-1}}) ÷ ∑_{i=2}^N (t_i - t_{i-1}) ) × (t_f - t_N) + x_{t_N}
      y_{t_f} ← ( ∑_{i=2}^N (y_{t_i} - y_{t_{i-1}}) ÷ ∑_{i=2}^N (t_i - t_{i-1}) ) × (t_f - t_N) + y_{t_N}
    Else x_{t_f} ← 오류 값, y_{t_f} ← 오류 값
  Return x_{t_f} and y_{t_f} //미래의 t_f시점의 (x, y)좌표값 반환
End
  
```

Fig. 26 Future location estimation algorithm.

Fig. 26의 미래의 위치 추정 함수에 사용되는 과거 이력 위치정보의 개수는 N 이라고 가정한다. 알고리즘 *future_location* 은 이동객체 *MP*의 *vmo_id*와 미래의 임의의 시점 t_f 을 입력 값으로 받은 후, *v_history_data* 테이블에서 입력된 *vmo_id*를 가지는 객체를 검색한다. 검색된 객체가 존재할 경우, 이력 정보가 저장된 *v_history_data* 테이블에서 t_{now} 시점 이전의 N 개의 위치 좌표를 검색하여 각각 x_i , y_i , t_i 의 배열에 저장한다. 이

때 배열의 첨자 i 는 $1 \sim N$ 까지의 값을 가진다. 임시 배열에 저장된 위치 좌표를 이용하여 x_{t_i} 와 y_{t_i} 값을 구하여 결과를 반환한다. 이 때 x_{t_i} 와 y_{t_i} 을 구하는 수식의 설명은 다음과 같다. 먼저, $\sum_{i=2}^N (x_{t_i} - x_{t_{i-1}})$ 와 $\sum_{i=2}^N (y_{t_i} - y_{t_{i-1}})$ 는 N 개의 이력 위치정보의 이동시간의 합이다. $(t_f - t_N)$ 은 이동객체 MP 가 현재 시점 t_N 에서 미래 시점 t_f 까지 이동할 향후 이동시간이다. x_{t_N} 과 y_{t_N} 은 t_N 시점의 x , y 좌표값을 나타낸다.



V. 분석 방법 및 성능 평가

5장에서는 본 연구의 성능을 평가하기 위한 분석 방법과 성능 평가 기준을 작성하고, 그에 따른 구현 결과를 나타내었다. 또한 작성된 분석 방법을 기반으로 제안된 차량 하이브리드 기법에 대한 성능을 평가하고 결과를 분석하였다.

1. 분석 방법



이 절에서는 성능을 평가하기 위한 분석 방법과 전제 조건 그리고 성능 평가 기준을 작성하였다.

분석을 위해서 다음과 같이 네 가지 방법으로 측정하였다.

첫 번째, 차량 운행궤적의 유사성(類似性) 측정이다. 이는 실시간으로 도로를 주행한 차량 운행궤적과 시간 간격, 방향각 유사궤적, 방향각과 시간간격을 혼합한 하이브리드(Hybrid) 궤적을 비교 분석하여 실시간 차량 운행궤적과 어느 정도 정확한 궤적을 표현하는지를 측정하는 것이다. 이를 위해 실제 차량 이동 객체에 GPS장비를 탑재한 후 실시간으로 제주시 일부 지역을 운행한 운행궤적과 시간간격별 차량 운행궤적 그리고 제안한 방향각 궤적, 그리고 하이브리드 궤적으로 표현한 운행궤적의 유사성을 비교·분석하였다.

차량 운행궤적을 비교하기 위해 GPS TrackMaker라는 프로그램과 차량 영상 기법을 이용하여 차량 운행궤적의 유사성을 정량적(定量的) 및 시각

적(視覺的)으로 분석하였다.

두 번째, 차량 운행궤적의 일관성(一貫性)측정이다. 동일한 조건하에서 동일한 구간을 운행한 차량은 동일한 차량 운행궤적을 표현해야 한다. 예를 들어, 처음 1분 시간간격을 갖고 운행한 차량 운행궤적과 두 번째 1분 시간간격을 갖고 운행한 차량은 동일한 운행궤적을 표현해야 한다. 따라서 시간간격별과 방향각 별 유사궤적, 그리고 하이브리드 유사궤적으로 운행 궤적의 일관성을 시각적으로 분석하였다.

세 번째, 성능 비교 분석이다. 실시간으로 운행한 차량 위치정보량(파일 크기와 레코드 수)과 시간간격별 차량 위치정보량, 방향각 유사궤적 기법으로 생성된 정보량 그리고 하이브리드 유사궤적을 비교 분석하여 성능을 평가하였다.

네 번째, 효율성을 평가하였다. 실시간으로 운행한 차량 운행거리와 시간간격별로 측정한 차량 운행거리, 방향각별 차량 운행거리 그리고 하이브리드 궤적으로 추출한 차량 운행거리를 비교하여 효율성을 평가하였다.

2. 분석 환경과 조건

이 절에서는 비교 분석을 위해 분석 환경과 조건을 제시하였다.

1) 분석 환경

분석 환경은 실제 도로 주행 시 제주시 도로 상황을 그대로 반영하기 위해 도심 속 도로 및 주택가 골목길, 도시 외각도로를 운행하였으며, 차량 운행 시 도로 교통 상황을 그대로 반영하여 차량 위치정보를 수집하였

다. 또한 교통량에 따른 분석이 필요하기 때문에 아침, 저녁 러시아워 (Rush hours) 시간대의 차량 위치정보도 수집하였다.

2) 분석 조건

분석 조건은 분석 비교할 정보 수집에 따른 조건으로 첫 번째, 차량 운행 경로 선택은 차량 운전자가 주관적인 성향이 포함이 되어있지만 제주 시 도로 상황에 맞게 직선 도로와 곡선 도로 그리고 정체 구간, 골목길 등 다양한 상황을 바탕으로 경로를 선택하였다.

두 번째, 실시간 차량 위치정보는 동일한 경로를 반복 운행하였으며, 차량 운행 중 신호 대기 및 주·정차 시 차량 위치정보는 제거하였다. 이는 동일한 차량 위치정보가 계속해서 입력으로 인한 시스템 가용성을 떨어뜨리기 때문이다.

세 번째, 비교군인 시간간격별 차량 위치정보는 실시간 차량 위치정보에서 시간간격 별로 추출하였다. 시간간격은 1분, 2분, 3분, 4분, 5분으로 설정하였다.

네 번째, 방향각에서 사용된 방향각 차는 10° , 20° , 30° 로 설정하였다.

다섯 번째, 시간간격과 방향각을 이용한 하이브리드 차량 위치정보 추출은 1분 시간간격과 방향각 10° 와 20° 를 이용하였다.

3. 분석 시스템과 도구

이 절에서는 차량 위치정보를 수집·추출을 위한 시스템 구성과 비교 분석을 위해 사용된 분석 도구들을 제시하였다.

1) 분석 시스템 구성과 도구

차량 위치정보를 추출하기 위한 시스템 구성은 GPS신호를 수신할 수 있는 GPS안테나와 수신된 GPS신호를 기록, 분석할 수 있는 노트북 PC가 사용되었다.

GPS안테나는 (주)네비우스사의 NSA-U3, 차량 운행궤적을 표현하기 위하여 이스트소프트(EST soft)사의 PC용 위성항법 프로그램인 알맵 2.0 Deluxe을 사용하였다.

Fig. 27은 Fig. 8의 사각형 점들을 선분으로 표시하여 차량 운행궤적을 표현한 것이다.

차량 위치정보(경·위도 좌표값)와 차량 운행궤적 분석을 위하여 GPS TrackMaker라는 프로그램을 사용하였다.

GPS TrackMaker는 Odilon Ferreira Junior가 만든 GPS 수신 프로그램으로 다양한 GPS위성에서 위치 좌표를 수신할 수 있으면 NMEA-0183 Protocol를 지원해 준다. Fig. 28은 GPS TrackMaker 실행 화면이며, 실제 차량 운행 중 캡처(Capture)하였다.

이 밖에 추출된 차량 궤적을 비교하기 위해 차영상 기법과 유사궤적 방향각 추출을 위해 Java j2sdk를 사용하였으며, 추출한 차량 유사궤적 위치정보를 저장할 데이터베이스 Oracle 9i가 사용되었다.

Table 13은 차량 유사궤적을 추출하기 위해 사용된 컴퓨터 시스템과 프로그래밍 언어 및 분석도구들이다.



Fig. 27 Vehicle running trajectory.

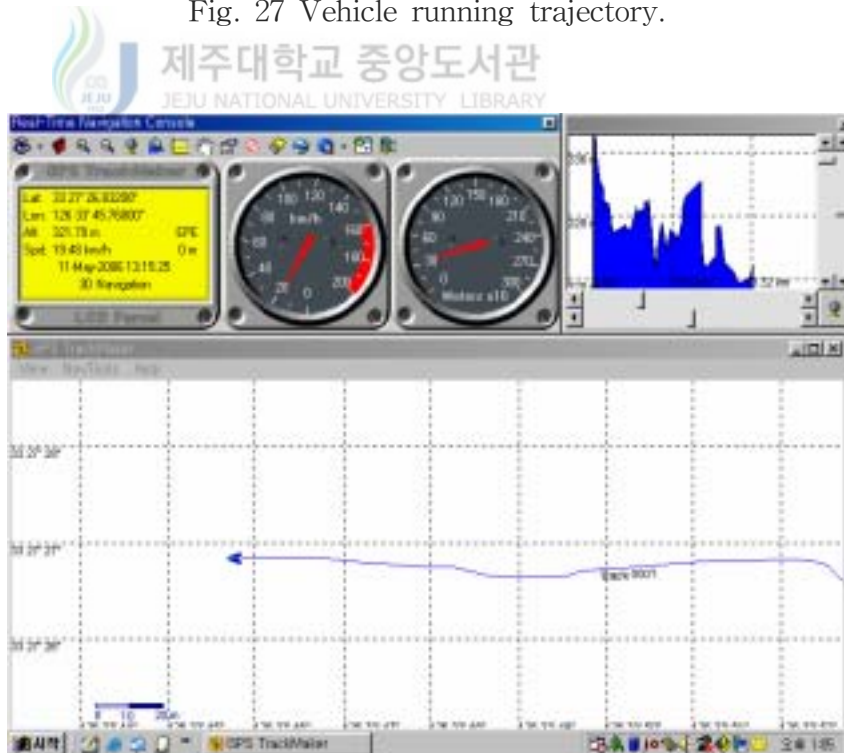


Fig. 28 GPS TrackMaker execution.

Table 13. System and analysis tool for vehicle location information and similar trajectory abstraction.

노트북	CPU : PentiumII Processor RAM : 288MB Hard Disk : 30GB OS : Microsoft Windows 98 Second Edition
데스크 탑	CPU : PentiumIV 1.70GHz RAM : 512MB Hard Disk : 120GB OS : Microsoft Windows XP Professional
GPS 안테나	제조사 : (주) 네비우스 모델명 : NSA-U3
프로그래밍 언어	Java2 SDK(ver 1.4.2)
분석도구	이스트소프트(EST soft)알맵 2.0 Deluxe GPS TrackMaker(ver 13.0)
데이터베이스	Oracle 9i

4. 구현과 분석

1) 차량 운행궤적 추출

차량 운행궤적의 위치정보는 GPS TrackMaker를 사용하여 실시간으로 GPS수신기를 통해 수집하였으며, Table 14와 같이 운행하였다.

Table 14를 보면 운행 거리에 따라 파일 크기와 레코드 수가 증가 하는 것을 볼 수 있다.

Table 14에서 2차 자료를 보면 다른 자료들 보다 높은 값들이 발생했는데 이것은 실질적인 도로 상황을 적용했기 때문이다. Fig. 29를 보면 1차 차량 운행궤적과 2차 운행 궤적을 겹친 화면으로 거의 동일한 궤적을 갖

고 있지만 궤적 중에 서로 다른 궤적을 보이는 곳이 있다. 이것은 차량 운행 중 후진하여 다시 출발 시 생성된 차량 위치정보로 GPS수신 오류에 의한 결과이다.

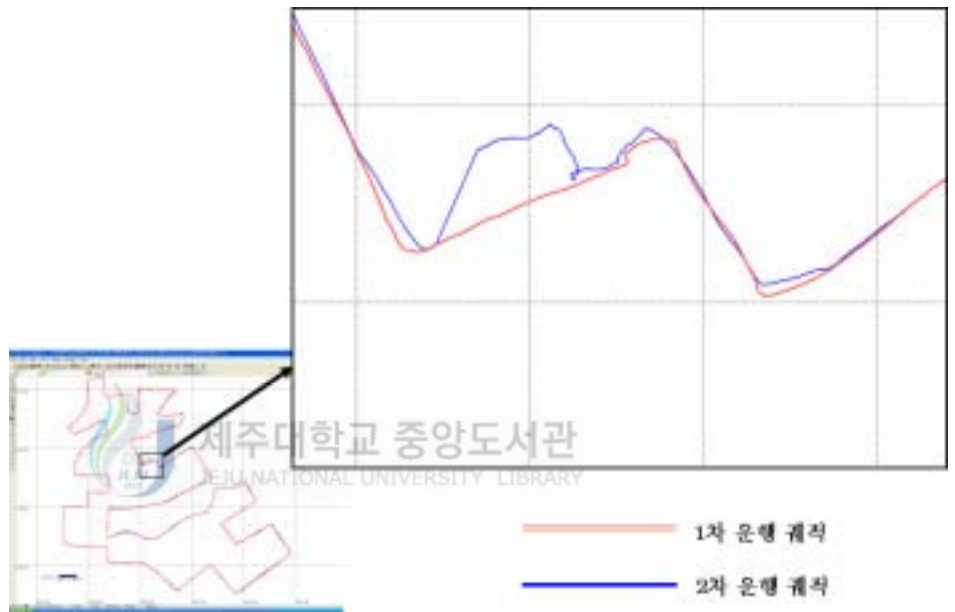


Fig. 29 GPS receiving error by vehicle's backward moving.

Table 14. Vehicle running data for vehicles location information abstraction.

횟수	1차	2차	3차	4차	5차
시작시간	14:48:05	13:12:40	19:08:13	07:19:06	08:08:43
종료시간	15:41:26	14:05:04	20:14:13	08:07:10	09:03:19
운행 시간(초)	3,201	3,164	3,960	2,884	3,276
운행 거리(km)	22,664	22,986	22,791	22,567	22,761
평균 속도(km/h)	25.5	26.3	20.7	28.2	25.0
파일 용량(kb)	131	167	148	124	136
레코드 수	1,885	2,399	2,142	1,754	1,912

Table 14에서 3차 차량 운행 시간을 보면 아침 러시아워 시간대인 5차 차량 운행 시간보다 684초가 더 지연되는 것을 볼 수 있다. 이는 약 11분 정도 차량 지체현상이 일어나는 것으로 저녁 러시아워 시간이 더 지체되는 경향을 보였다.

Table 14에 따른 5차례의 차량 운행궤적은 Fig. 30과 보는바와 같이 모두 동일하며, 운행 궤적은 1:250m 축적으로 표현하였다.



Fig. 30 Vehicle running trajectories.

2) 시간간격별 차량 운행궤적 추출

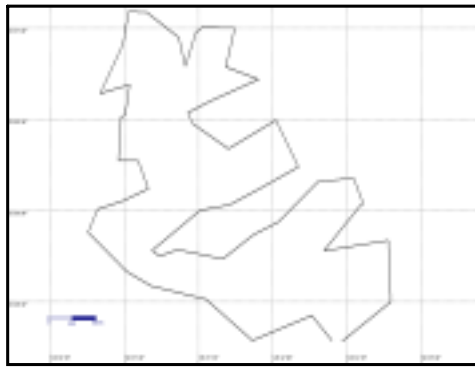
차량 운행궤적은 차량에 GPS장비를 탑재하여 동일한 경로를 5차 반복 하였으며, 다양한 차량 위치정보를 수집하기 위해 아침 출근, 저녁 퇴근 러시아워 그리고 오후 2시 시간대에 차량을 운행하여 차량 위치정보를 수집 하였다.

차량 위치정보는 차량 운행 중 신호대기와 주·정차 시 위치정보는 무시하여 저장하였다.

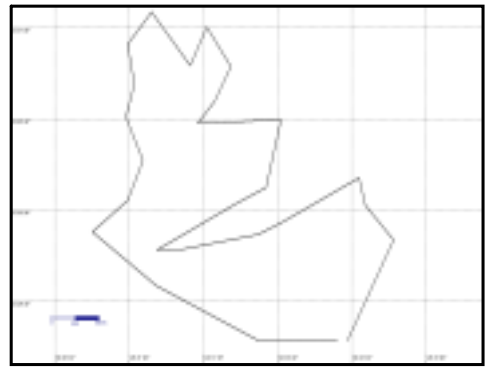
시간 간격은 실시간으로 획득한 위치정보에서 1분, 2분, 3분, 4분, 5분 간격으로 위치정보를 추출하였으며, 되도록 시간 간격을 유지할 수 있도록 하였다. 비록 실시간으로 GPS에서 차량 위치정보를 수집하였지만 GPS 수신 불량 지역과 차량 신호대기 및 주·정차 시 위치정보는 저장되지 않았기 때문에 정확한 시간 간격을 지킬 수 없다.

Fig. 31부터 Fig. 35는 수집한 차량 운행궤적에 따른 각 차수의 시간 간격별 차량 운행궤적을 나타낸 것이다.





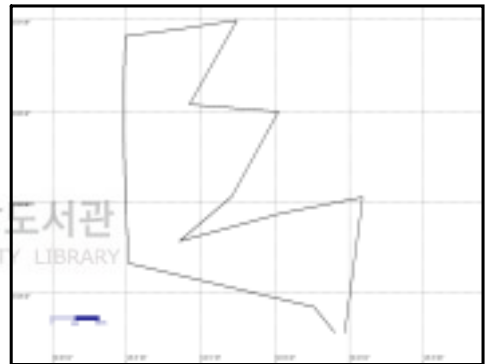
1min



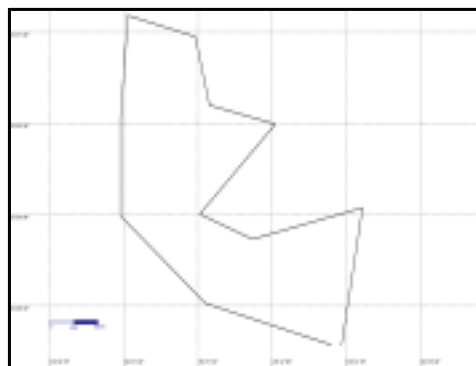
2min



3min



4min



5min

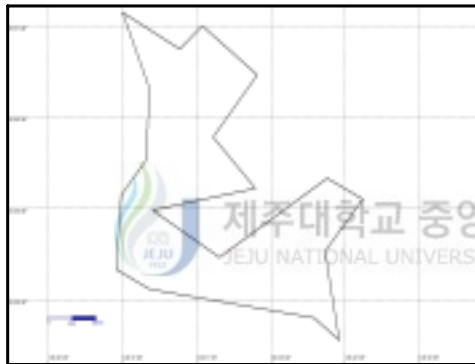
Fig. 31 The first time interval running trajectory.



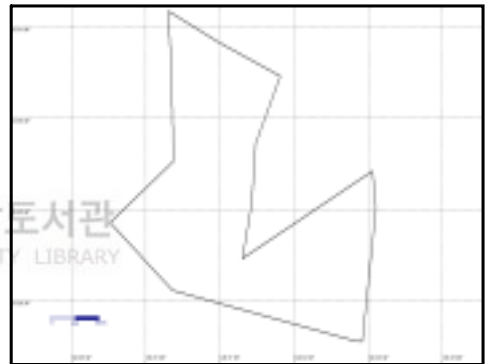
1min



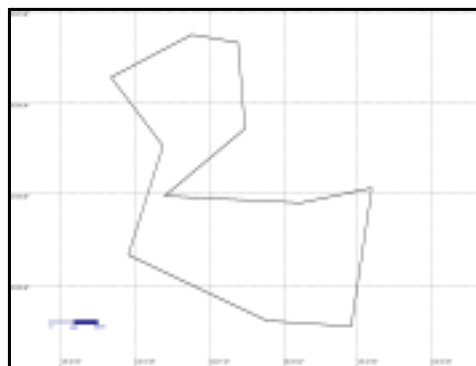
2min



3min

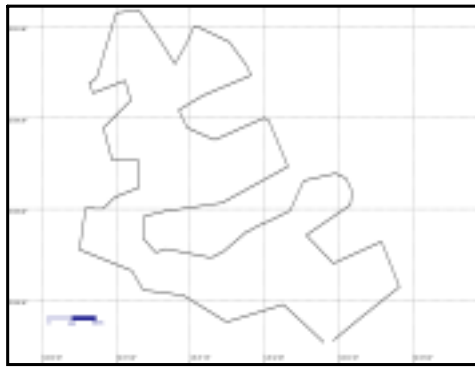


4min

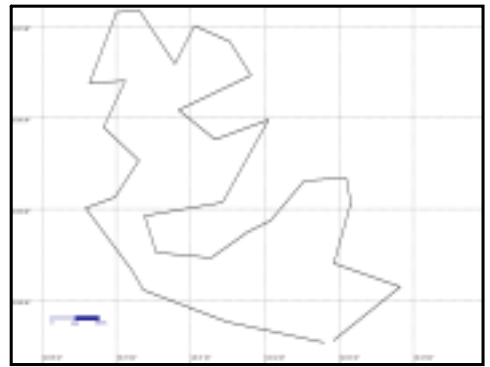


5min

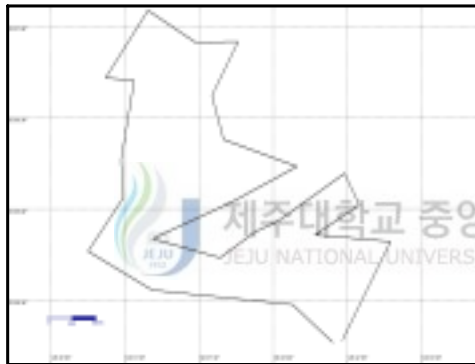
Fig. 32 The second time interval running trajectory.



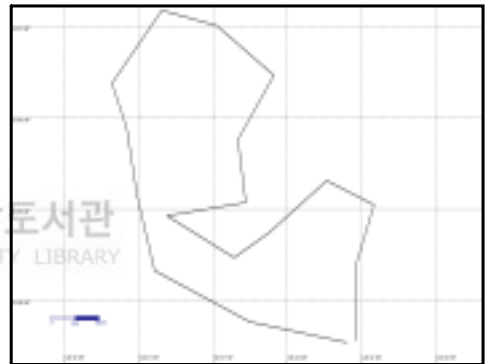
1min



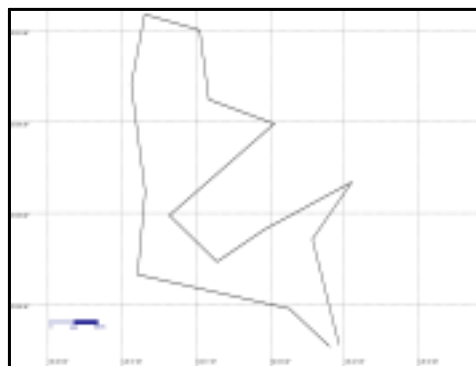
2min



3min

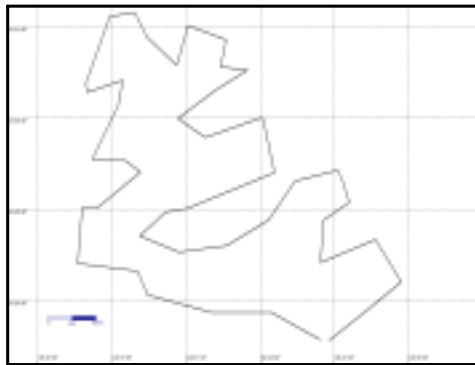


4min

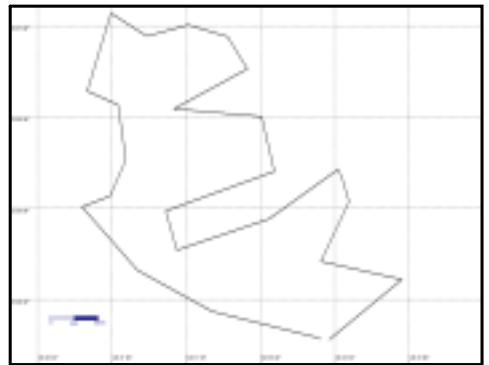


5min

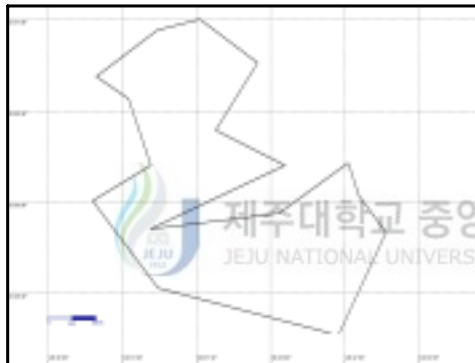
Fig. 33 The third time interval running trajectory.



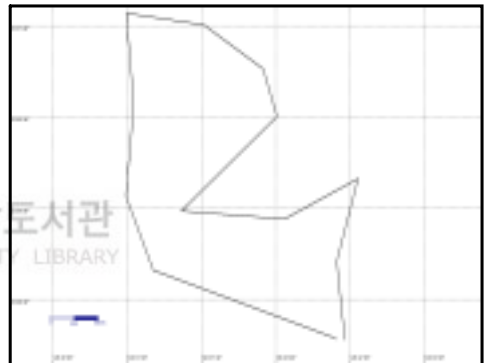
1min



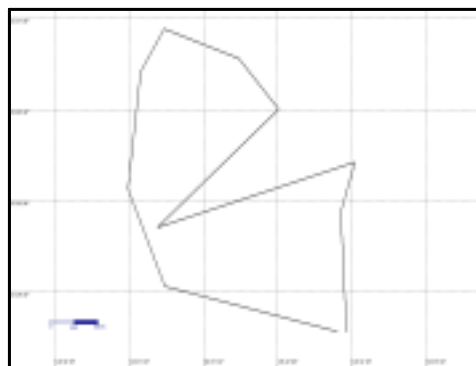
2min



3min



4min



5min

Fig. 34 The 4th time interval running trajectory.

3) 방향각 정보에 의한 차량 운행궤적 추출

방향각 정보에 의한 차량 운행 궤적 추출은 차량 운행 중 차량 진행 방향이 변경되는 시점의 차량 위치정보를 추출하여 저장하는 기법으로 GPS에서 수신된 차량 위치 좌표값 P_0 , P_1 , P_2 가 있다면 각각의 좌표값에서 P_0 에서 P_1 로 진행하는 진행 방향각 값 θ_1 그리고 P_1 에서 P_2 로 진행하는 진행 방향각 값 θ_2 을 계산한다. 계산된 진행 방향각 값 θ_2 에서 θ_1 을 빼서 진행 방향각 차를 계산한다.

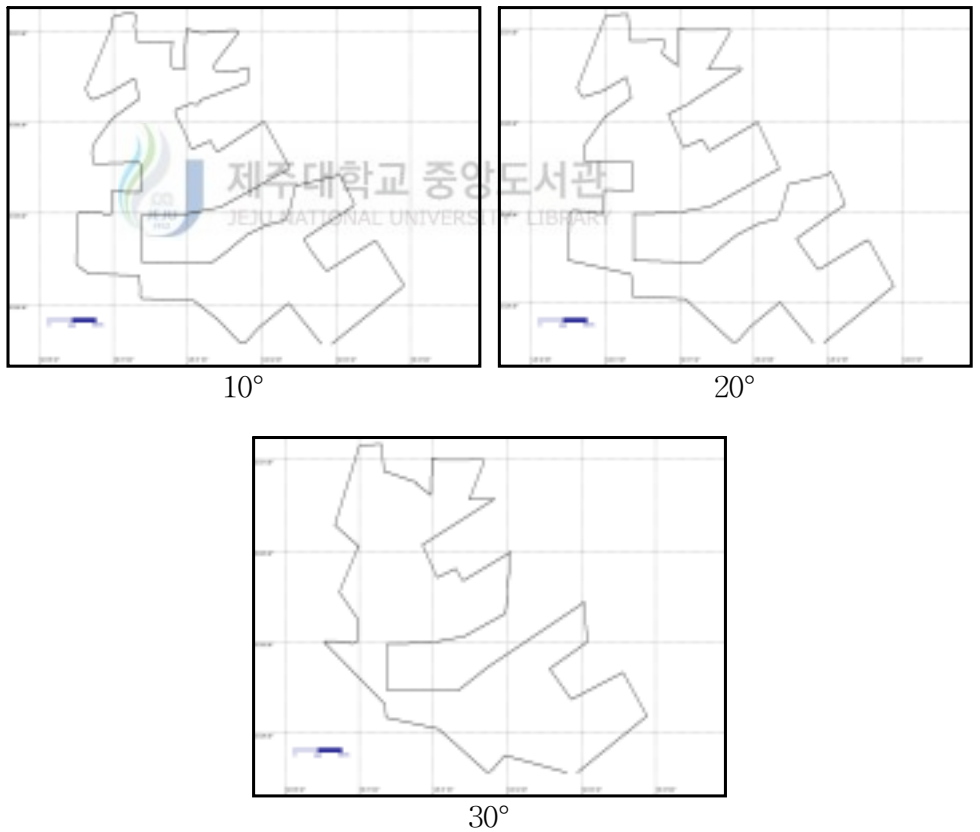


Fig. 36 The first similar trajectory.

계산된 방향각 차 범위가 벗어나는 차량 위치정보를 저장하고 이를 차량 운행 궤적으로 추출하였으며, 방향각 차 범위는 각각 10°, 20°, 30°로 설정하여 각각의 차량 운행궤적을 표현 하였다.

Fig. 36부터 Fig. 40까지는 1차부터 5차까지 실시간으로 측정한 차량 운행궤적에서 방향각 정보로 추출한 차량 운행궤적을 나타낸 것이다.

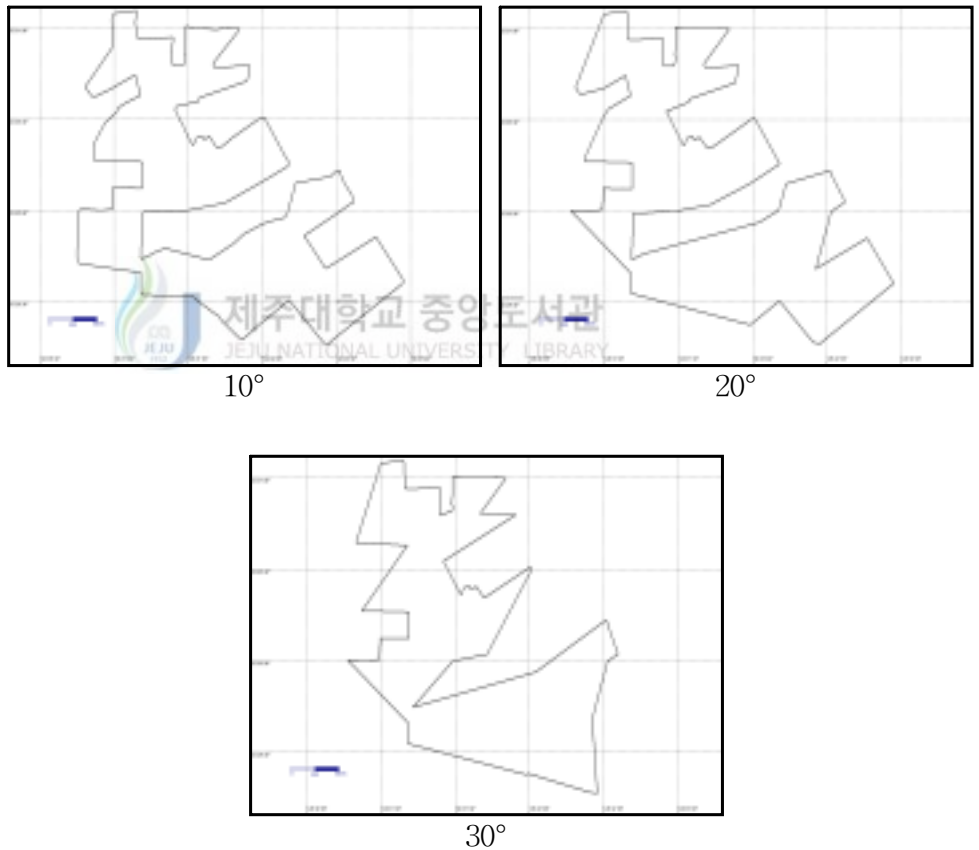


Fig. 37 The second similar trajectory.

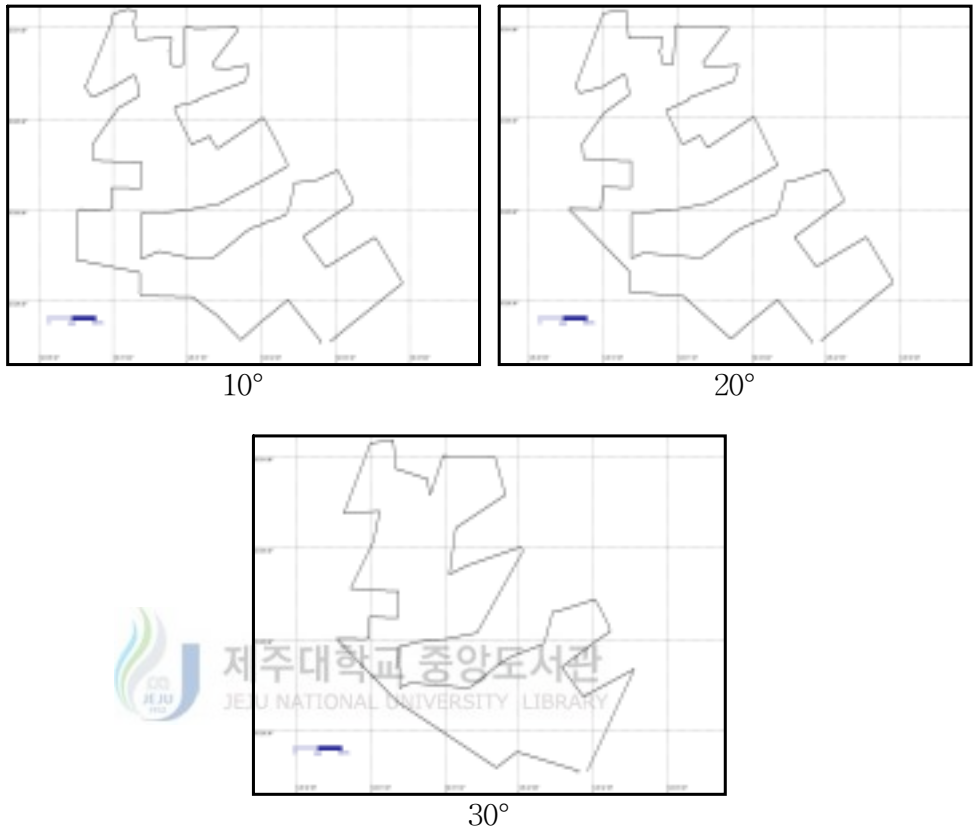
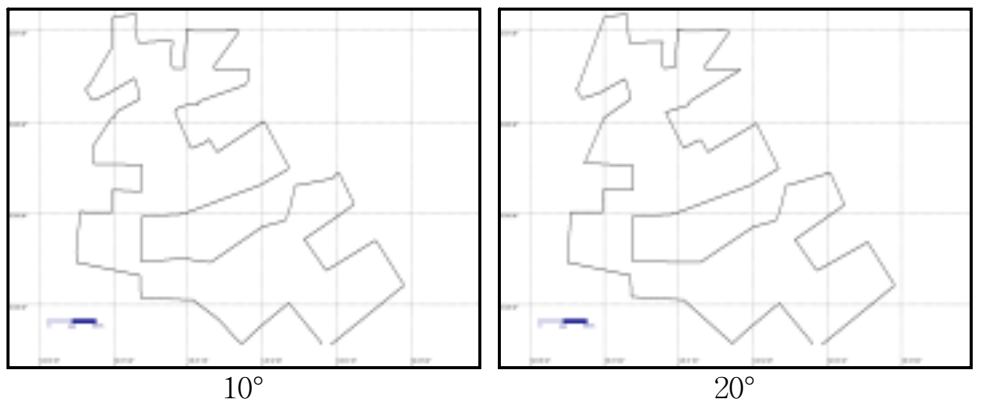
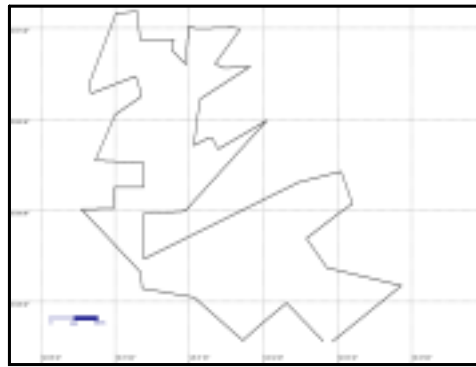


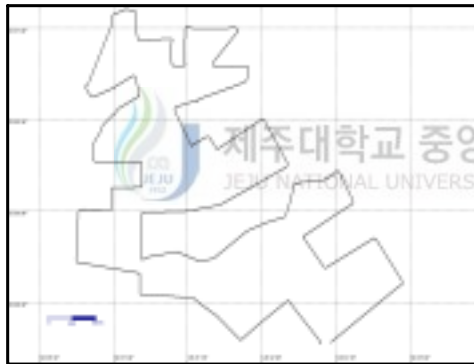
Fig. 38 The third similar trajectory.



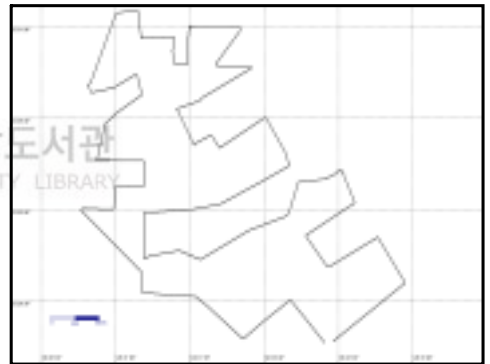


30°

Fig. 39 The 4th similar trajectory.



10°



20°



30°

Fig. 40 The 5th similar trajectory.

4) 하이브리드에 의한 차량 운행궤적 추출

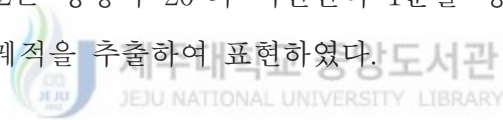
하이브리드 방법은 시간간격과 방향각 운행궤적보다 더 정확 차량 운행궤적을 표현하고자, 시간간격 차량 운행궤적 추출방법과 방향각 유사궤적기법을 혼합하여 차량 운행궤적을 추출하였다.

본 연구에서는 방향각 정보와 시간간격 차량 운행궤적을 병합하여 추출된 차량 궤적을 하이브리드 궤적이라고 하였다.

추출할 하이브리드 궤적은 방향각 10° 와 시간간격 1분 그리고 방향각 20° 와 시간간격 1분으로 정하였다.

Fig. 41은 방향각 10° 와 시간간격 1분을 병합하여 1차부터 5차까지 차량 운행궤적을 추출하여 표현하였다.

Fig. 42는 방향각 20° 와 시간간격 1분을 병합하여 1차부터 5차까지 차량 운행궤적을 추출하여 표현하였다.





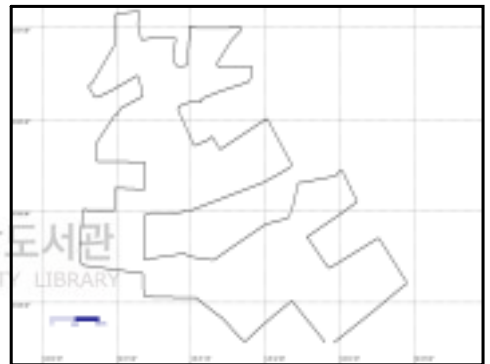
first



second



third



4th

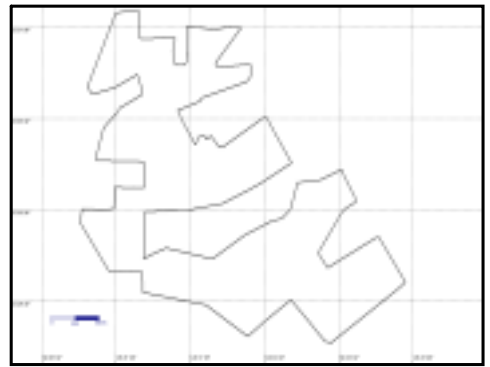


5th

Fig. 41 Hybrid trajectory.(Direction angle 10° + Time interval 1min)



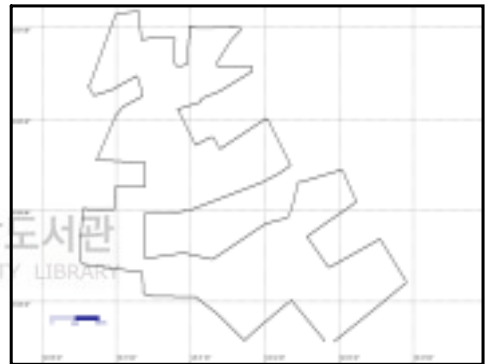
first



second



third



4th



5th

Fig. 42 Hybrid trajectory.(Direction angle 20° + Time interval 1min)

5) 시간 간격 차량 운행 궤적과 유사궤적 비교 분석

이번 항(項)은 전항에서 추출한 차량 운행궤적과 방향각 유사궤적, 그리고 하이브리드 유사궤적을 GPS TrackMaker와 차영상 기법을 통하여 차량 운행궤적의 유사성을 비교 분석하였다.

비교 방법은 실시간으로 운행한 차량 궤적과 시간간격별 차량 운행궤적 그리고 방향각 유사궤적, 하이브리드 유사궤적을 서로 비교 분석하였다.

우선 시각적 비교로 운행궤적의 유사성을 비교하기 위해서 GPS TrackMaker를 통하여 실시간 차량운행궤적과 시간간격별 차량궤적 그리고 방향각 유사궤적, 하이브리드 유사궤적을 비교하였다.

(1) 실시간 차량운행궤적과 시간간격별 차량운행궤적

Fig. 43은 1차 실시간 차량운행궤적과 시간간격별 차량운행궤적을 비교한 것으로 실시간 운행궤적과 시간간격(1min)운행궤적은 어느 정도 궤적의 일치성을 보이지만 시간간격이 클수록 실시간 운행궤적과 다른 운행궤적을 보이고 있다.

Fig. 43부터 Fig. 47까지는 각 차수별로 실시간 차량 운행 궤적과 시간간격별 운행 궤적을 병합시켜 나타내었다. 이를 통해 실시간 운행궤적과 시간 간격별 운행궤적의 유사성을 비교하였지만, 1분 시간간격에서 어느 정도의 유사성을 보였지만 시간간격이 커질수록 유사성이 점점 없어지는 것을 볼 수 있다.

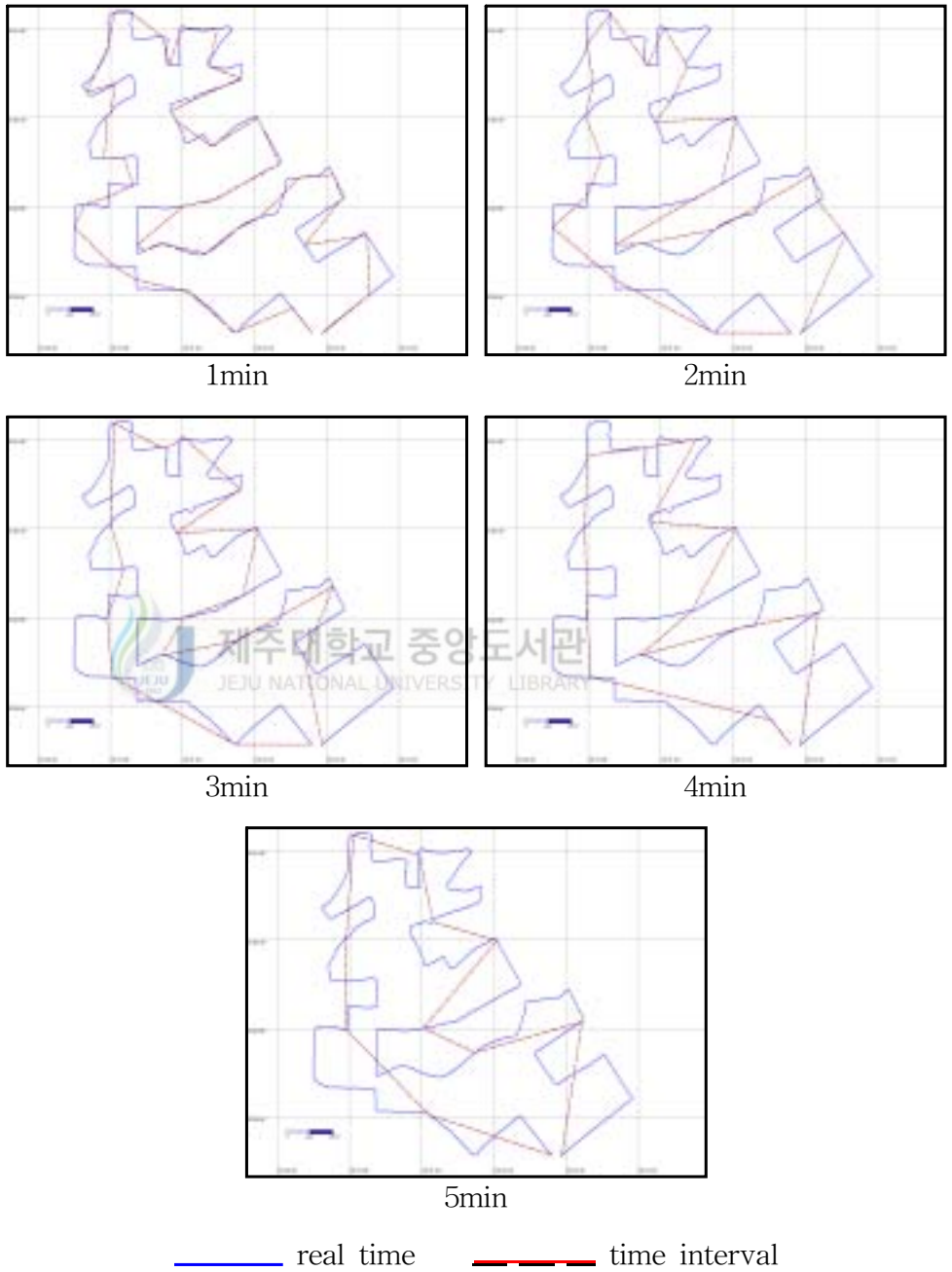


Fig. 43 The first real time running trajectory and time interval running trajectory.

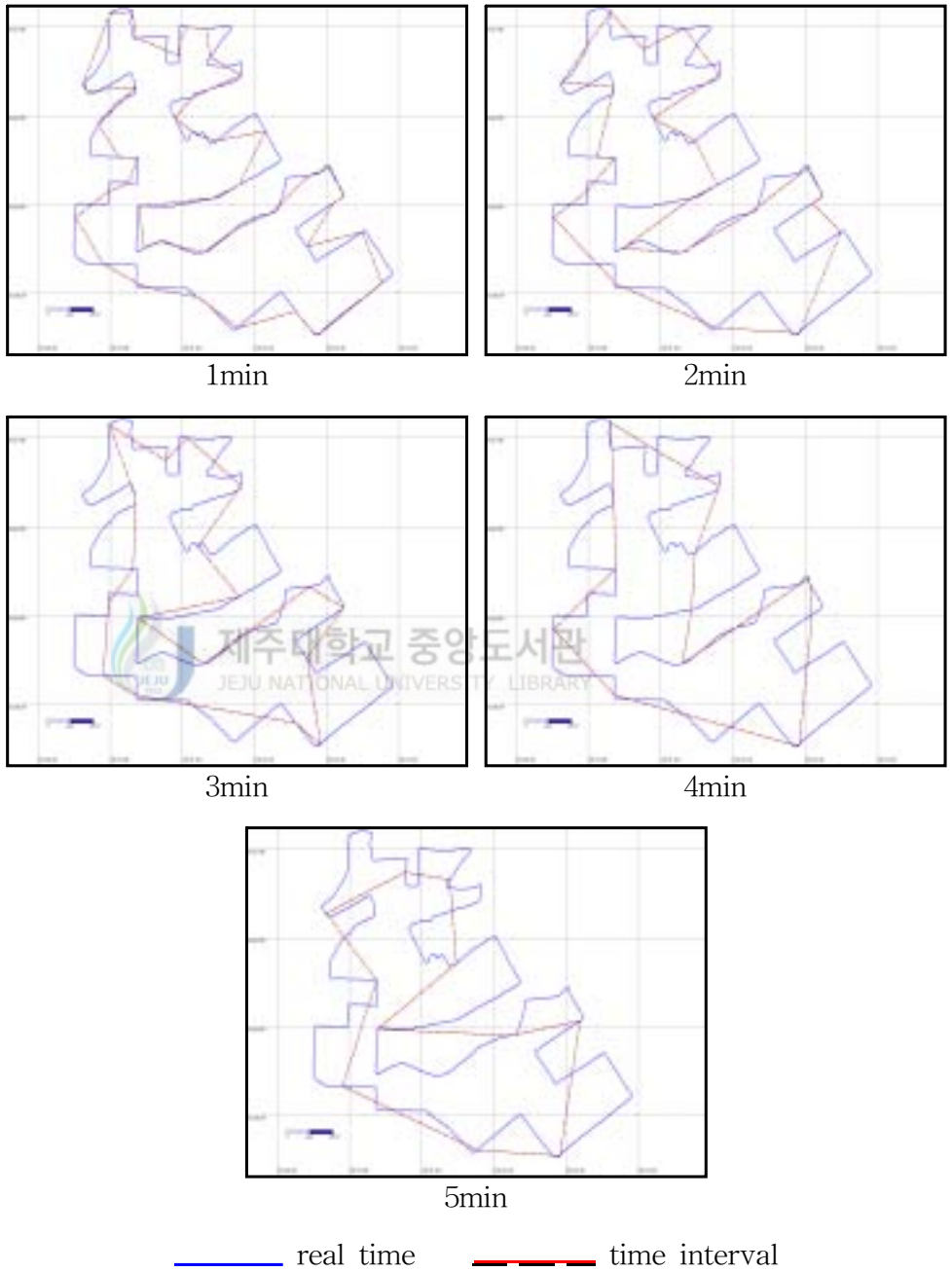


Fig. 44 The second real time running trajectory and time interval running trajectory.

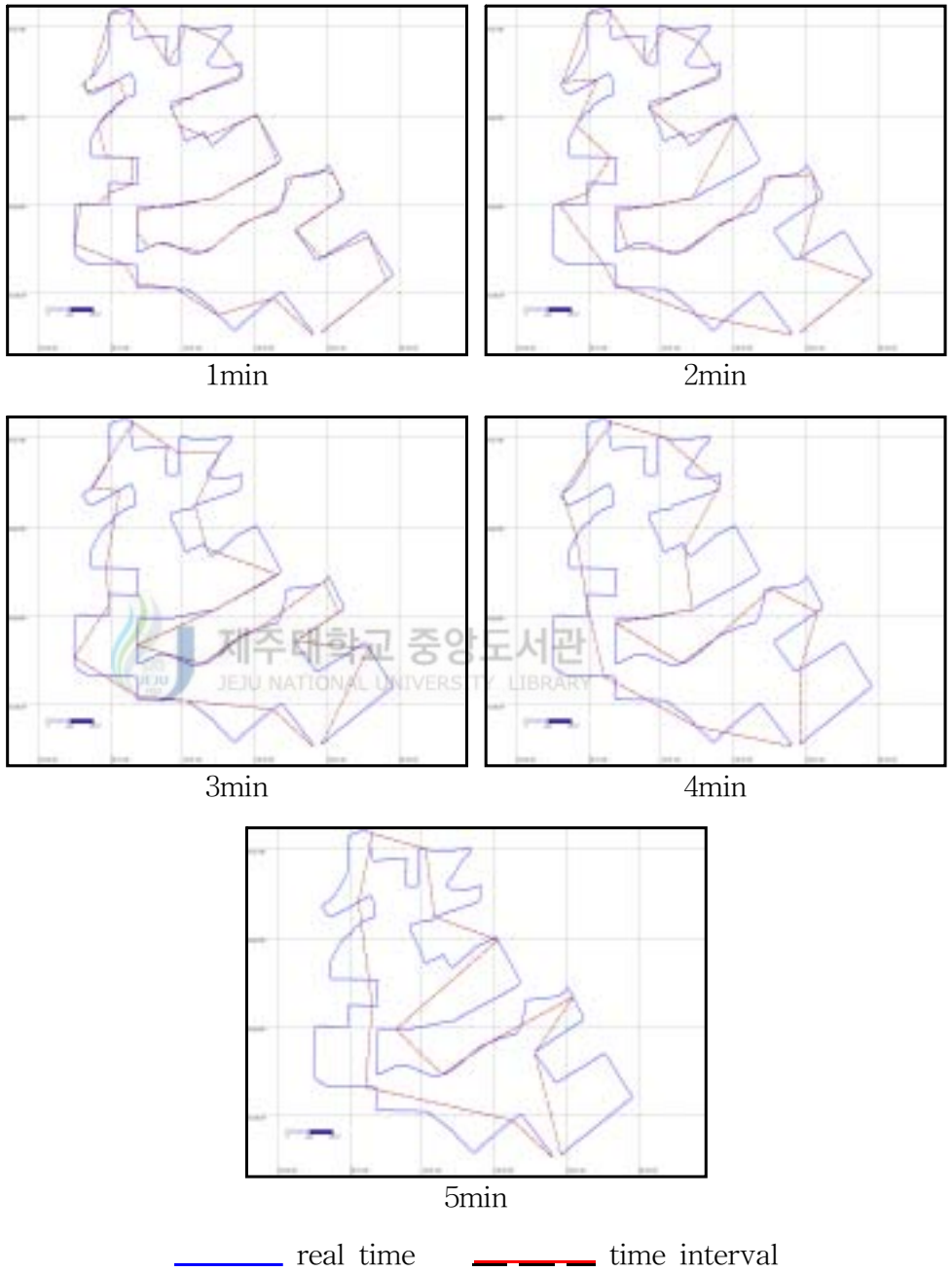


Fig. 45 The third real time running trajectory and time interval running trajectory.

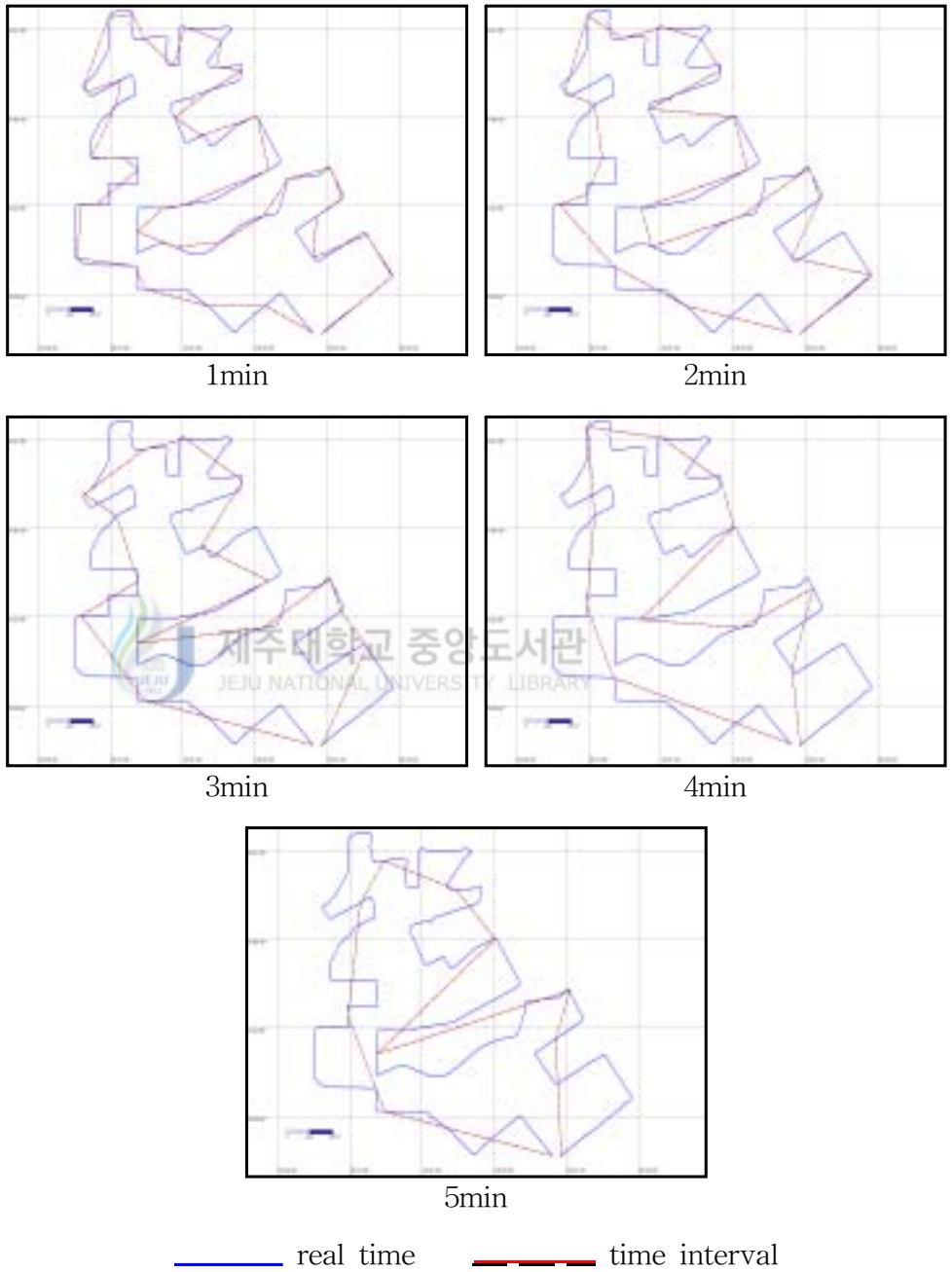


Fig. 46 The 4th real time running trajectory and time interval running trajectory.

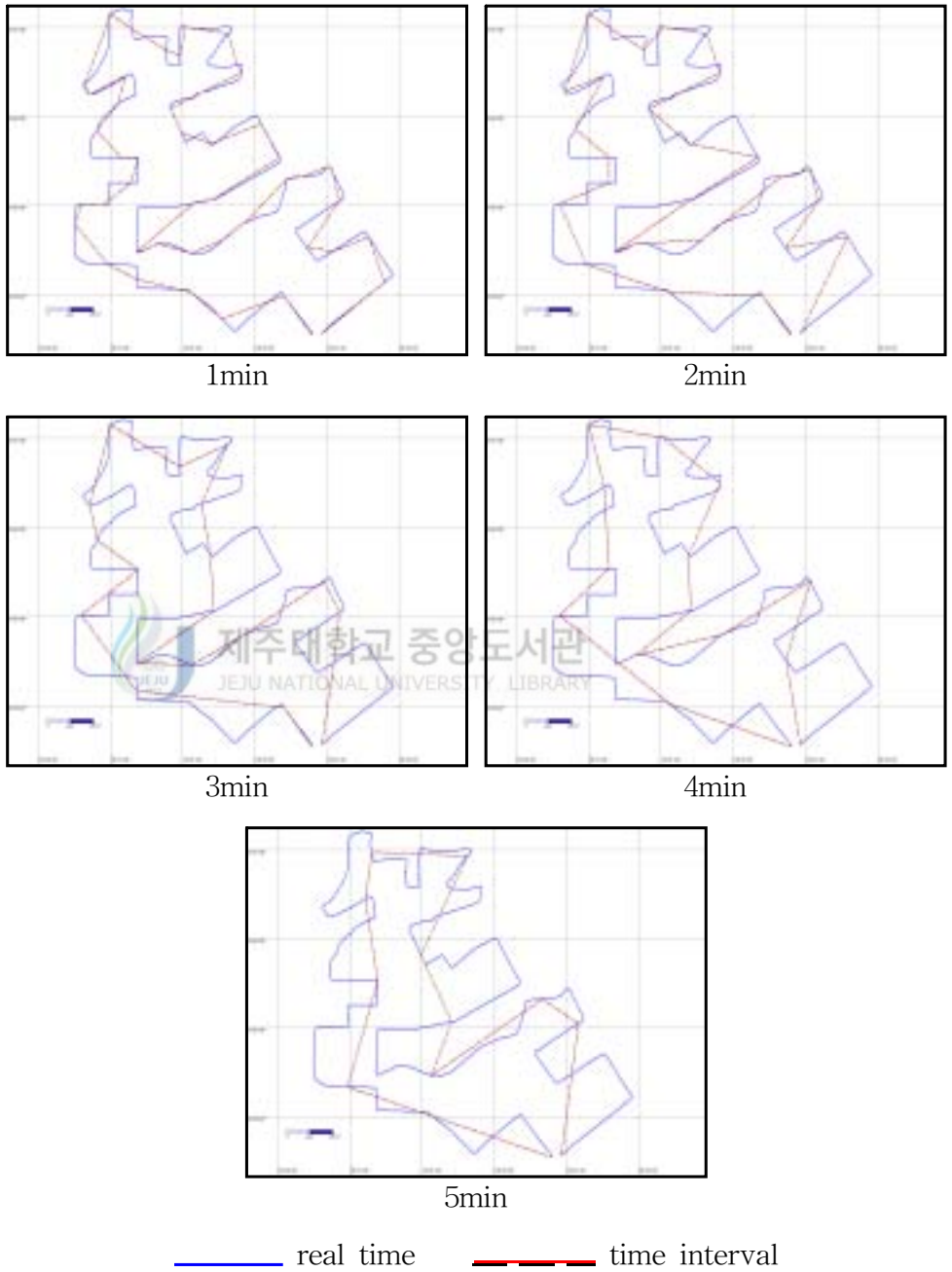


Fig. 47 The 5th real time running trajectory and time interval running trajectory.

(2) 실시간 차량 운행 궤적과 방향각 유사궤적

실시간 차량 운행궤적과 방향각 유사궤적을 GPS TrackMaker를 통해 시각적으로 비교하였다. 실시간 차량 운행 궤적과 방향각 10°, 20°, 30° 유사궤적을 각 운행 차수 별로 비교하여 유사성 정도를 보였다.

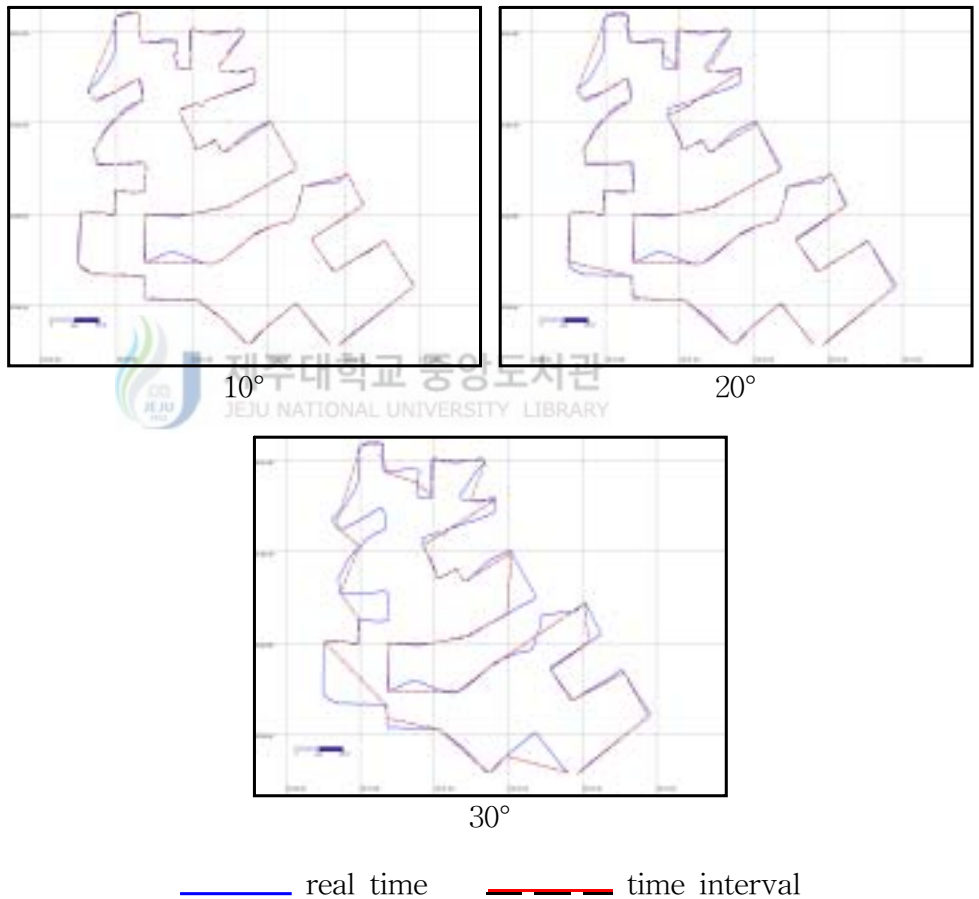


Fig. 48 The first real time running trajectory and similar trajectory.

Fig. 48부터 Fig. 52까지는 각 차수별로 실시간 운행궤적과 유사궤적 10°에서 30°까지 시각적으로 비교한 화면으로 방향각 10°범위에서 실시간 운행 궤적과 거의 일치하는 모습을 보였고, 방향각 범위가 커질수록 유사

성이 많이 떨어지는 것을 보이고 있다.

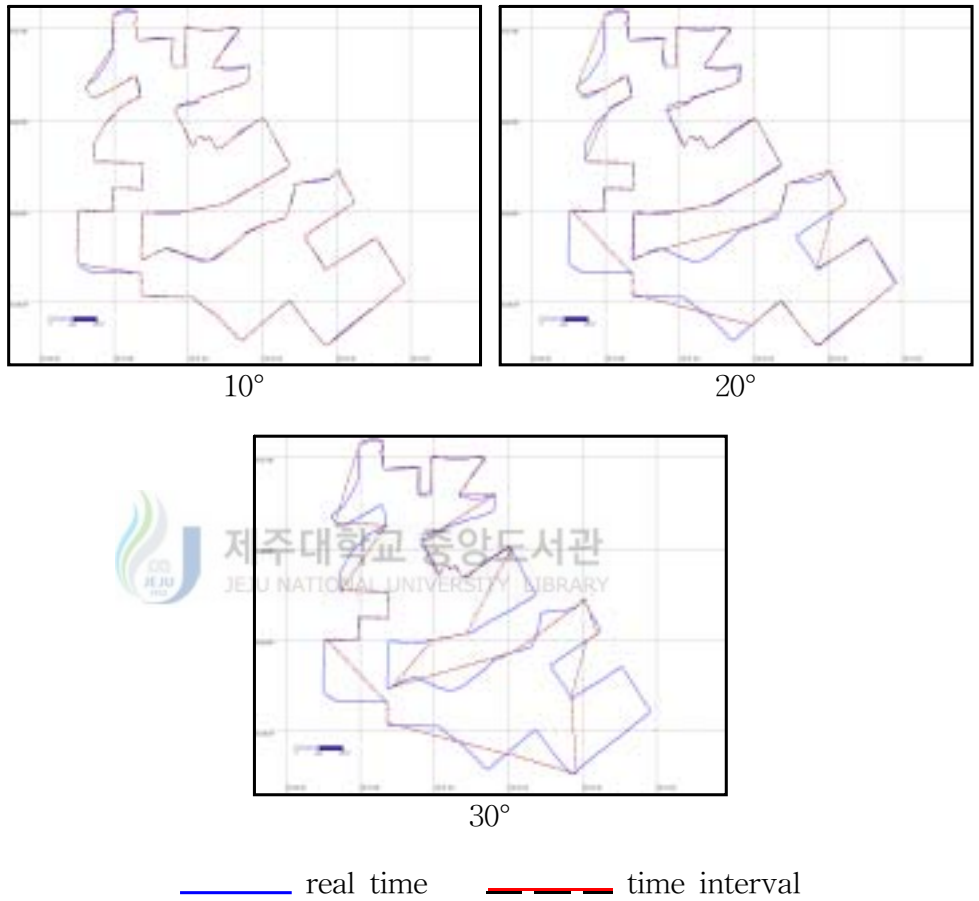
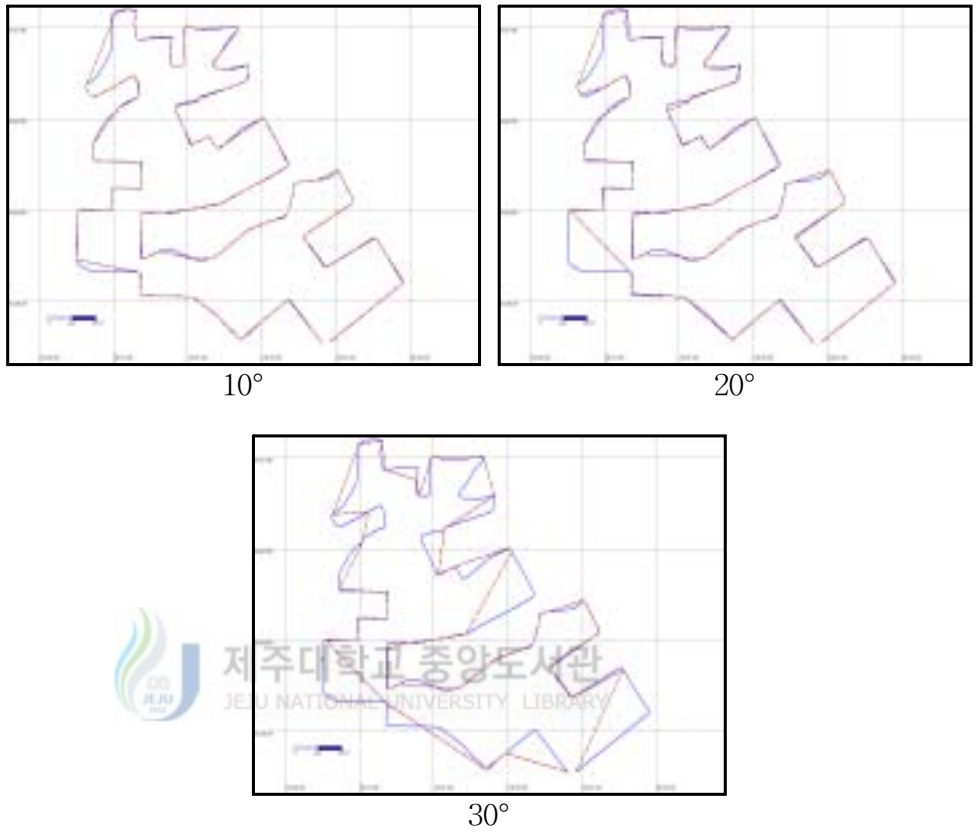
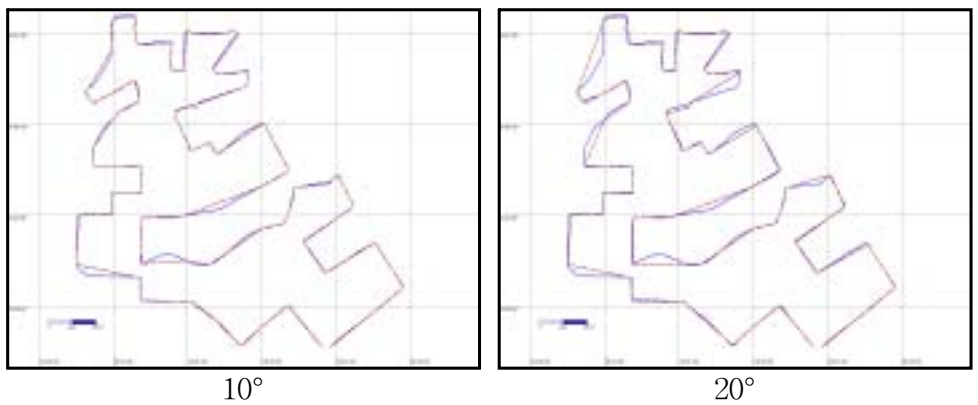


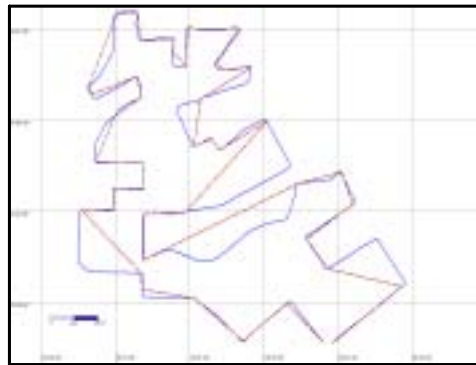
Fig. 49 The second real time running trajectory and similar trajectory.



— real time — similar trajectory

Fig. 50 The third real time running trajectory and similar trajectory.





30°

— real time - - - similar trajectory

Fig. 51 The 4th real time running trajectory and similar trajectory.



10°



20°



30°

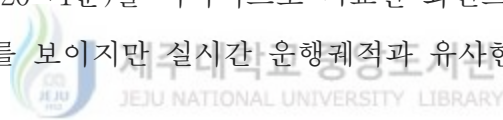
Fig. 52 The 5th real time running trajectory and similar trajectory.

(3) 실시간 차량 운행궤적과 하이브리드 유사궤적 비교

실시간 차량 운행궤적과 하이브리드 유사궤적을 시각적으로 비교 분석하였다. 하이브리드 유사궤적은 방향각 10° 와 시간간격 1분 하이브리드 유사궤적과 방향각 20° 와 시간간격 1분인 하이브리드 유사궤적으로 추출하였다.

추출된 하이브리드 유사궤적과 실시간 차량 운행궤적을 GPS TrackMaker를 통하여 시각적으로 비교 분석하였다.

Fig. 53은 실시간 차량 운행궤적과 하이브리드 유사궤적(방향각 $10^\circ+1$ 분)을 시각적으로 비교한 화면이다. 실시간 운행궤적과 거의 동일한 운행궤적을 보였다. Fig. 54는 실시간 차량 운행궤적과 하이브리드 유사궤적(방향각 $20^\circ+1$ 분)을 시각적으로 비교한 화면으로, 실시간 운행궤적과 약간의 차이를 보이지만 실시간 운행궤적과 유사한 궤적을 보였다.



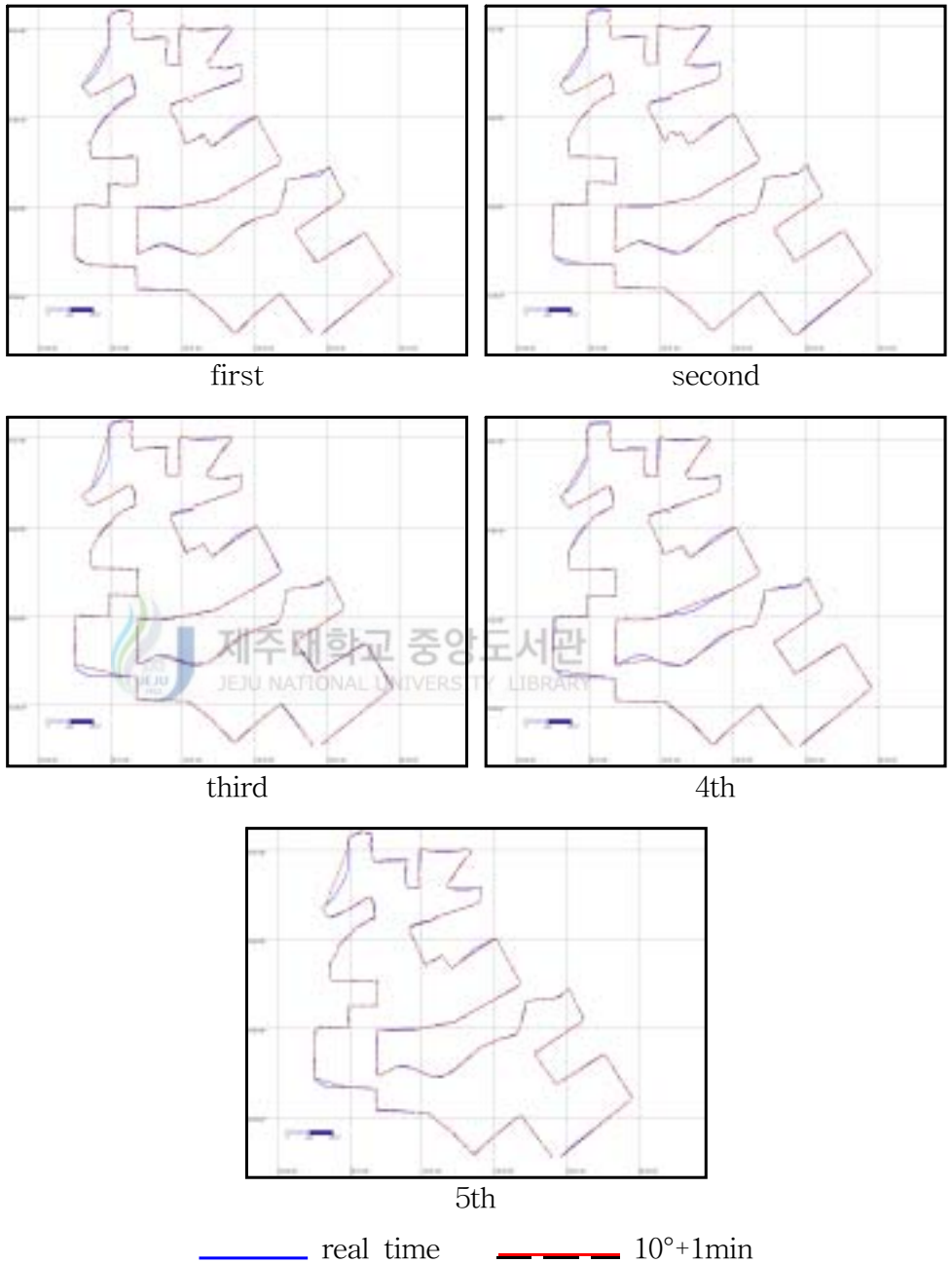
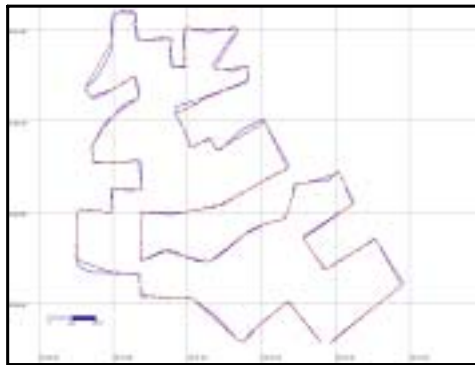
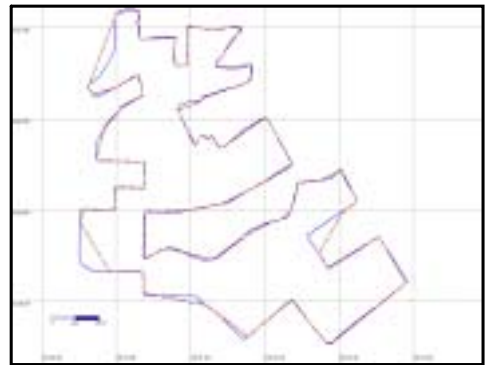


Fig. 53 Real time vehicle running trajectory and hybrid trajectory.



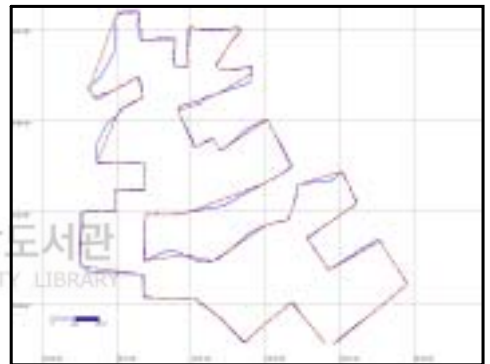
first



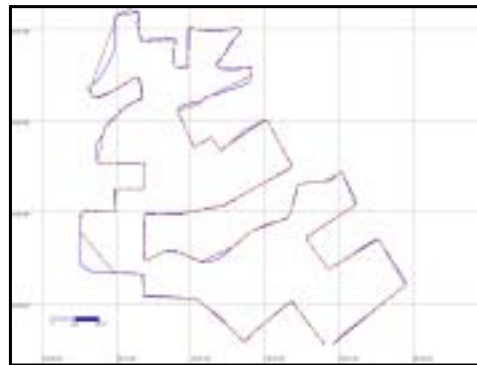
second



third



4th



5th

— real time — 20°+1min

Fig. 54 Real time vehicle running trajectory and hybrid trajectory.

(4) 실시간, 방향각, 하이브리드 유사궤적 비교

이번 항에서는 실시간 차량 운행궤적과 방향각 그리고 하이브리드 유사궤적을 시각적으로 분석하였다. 이는 Fig. 53에서 하이브리드 유사궤적이 실시간 차량 운행궤적과 거의 동일한 차량 운행궤적을 보였다.

세 기법의 비교는 소량의 위치정보로 실시간에 가까운 운행궤적을 표현할 수 있는 기법을 찾기 위해서이다.

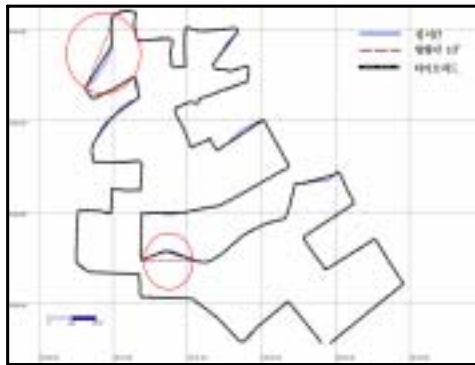
비교 방법은 각 차별로 실시간과 방향각 10° 그리고 하이브리드 유사궤적(방향각 $10^\circ+1$ 분)을 GPS TrackMaker로 비교하였다. 또한 실시간과 방향각 10° 그리고 하이브리드 유사궤적(방향각 $20^\circ+1$ 분), 실시간과 방향각 20° 그리고 하이브리드 유사궤적(방향각 $20^\circ+1$ 분)과 각각 비교하였다.

Fig. 55는 실시간 운행궤적과 방향각 10° 그리고 하이브리드 유사궤적(방향각 $10^\circ+1$ 분)을 표현한 화면이다.

Fig. 55에서 원(○)으로 되어있는 부분에서 방향각 유사궤적과 하이브리드 유사궤적이 약간의 차이를 보이고 나머지 궤적에서는 동일한 궤적을 표현하고 있다.

Fig. 56은 실시간 운행궤적과 방향각 10° 그리고 하이브리드 유사궤적(방향각 $20^\circ+1$ 분)을 표현한 화면으로 원(○)으로 되어있는 부분에서 약간의 차이를 보이고 있지만 방향각 10° 의 궤적과 많이 일치하는 것을 볼 수 있었다.

Fig. 57은 실시간 운행궤적과 방향각 20° 그리고 하이브리드 유사궤적(방향각 $20^\circ+1$ 분)을 표현한 화면으로 원(○)으로 되어있는 부분에서 약간의 차이를 보이면서 방향각 20° 보다는 약간 실시간 운행궤적과 비슷한 궤적을 표현함을 보이고 있다.



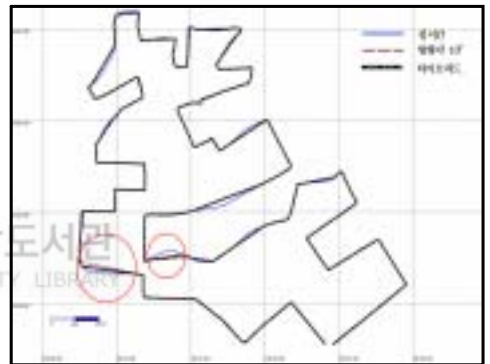
first



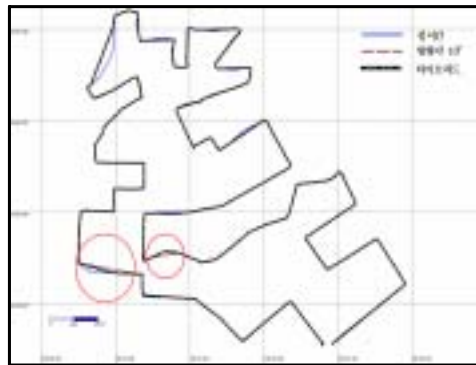
second



third

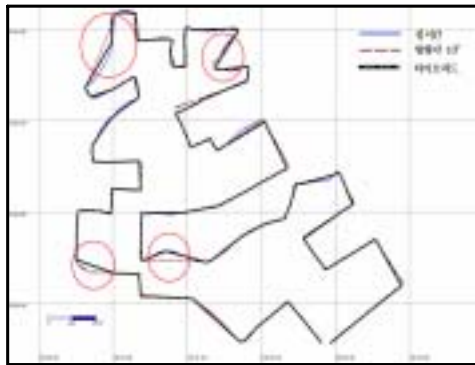


4th



5th

Fig. 55 Trajectory comparison
(Hybrid trajectory : direction angle $10^\circ + 1\text{min}$).



first



second



third



4th



5th

Fig. 56 Trajectory comparison
(Hybrid trajectory : direction angle $20^\circ + 1\text{min}$).

(5) 차영상 비교

차영상(Difference image) 기법은 두 개의 이미지를 이진화(Binarization)시켜 두 이미지의 차를 비교 분석하여 동일한 이미지인지를 판단하는 기법으로 본 연구에서는 차량 운행궤적의 유사성을 이미지 화소수를 비교 분석하여 측정하였다. 즉 두 이미지의 궤적 선분이 겹쳐지는 부분은 블랙화소(Black pixel)로 표현되고, 겹쳐지지 않는 부분은 화이트화소(White pixel)로 표현되도록 하였다. 따라서 두 이미지가 동일한 궤적을 그리면 블랙화소수가 많아지고 동일하지 않으면 화이트화소수가 많아진다.

이미지 비교 방법은 실시간 차량 운행궤적 이미지와 시간간결별 운행궤적 이미지, 방향각 별 유사궤적 이미지 그리고 하이브리드 유사궤적 이미지를 차영상 기법을 통하여 비교하였다.

기준치를 만들기 위해 처음 실시간 차량 운행궤적과 동일한 크기의 화이트 이미지를 만들어 차영상 기법을 통하여 이미지 차를 비교하여 블랙화소와 화이트화소 수를 기준으로 삼았다.

기준치 이미지 화소의 블랙화소와 화이트화소를 설정하여 각각의 비교군들과 비교하였다.

이미지 크기는 400x376으로 화소수는 150,400개이다.

Fig. 58은 1차 기준치 차영상을 구하기 위한 차영상 실행화면이다.

Fig. 59는 실시간 차량 운행궤적과 시간 간격 1분인 차량 운행궤적을 차영상으로 비교한 화면으로 화이트화소가 많아진 것을 시각적으로 확인할 수 있었다.

Fig. 59부터 Fig. 63까지는 1차 실시간 운행궤적 각 시간간결별 차영상 비교화면이다.



Fig. 58 The first norm difference image abstraction.



Fig. 59 The first time interval difference image abstraction(1min).

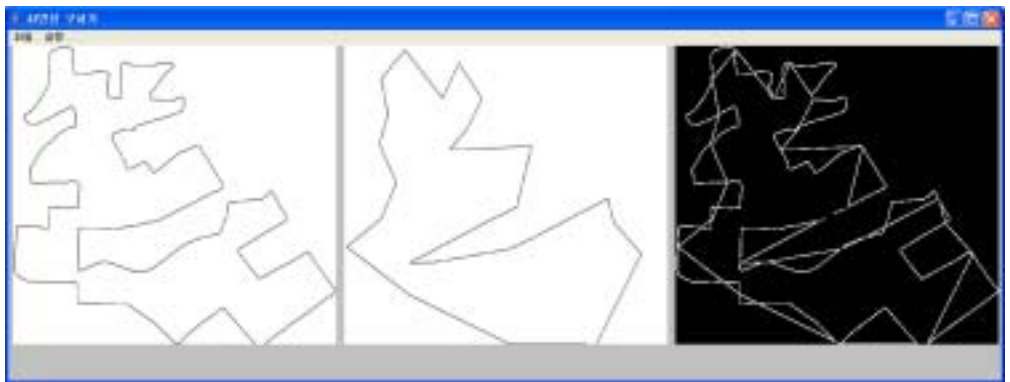


Fig. 60 The first time interval difference image abstraction(2min).

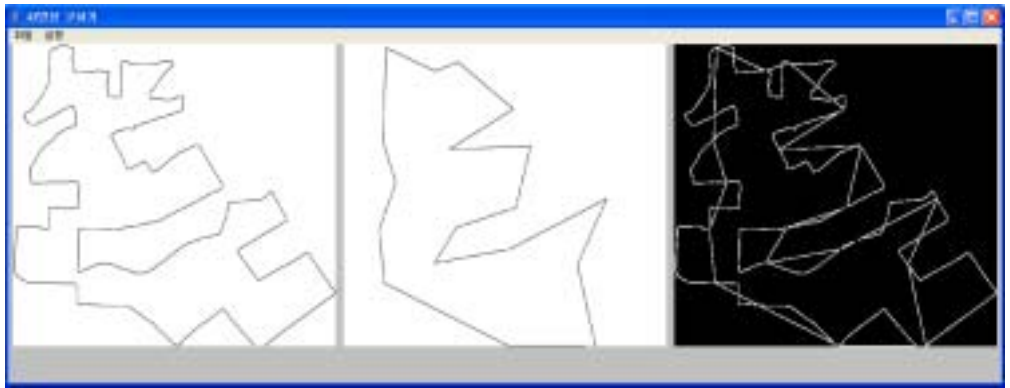


Fig. 61 The first time interval difference image abstraction(3min).



Fig. 62 The first time interval difference image abstraction(4min).

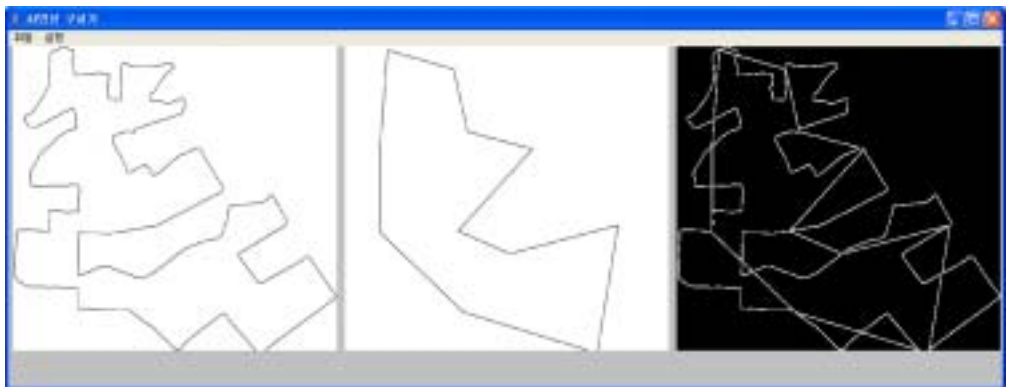


Fig. 63 The first time interval difference image abstraction(5min).

Fig. 64는 1차 실시간 차량 운행궤적과 방향각 10° 의 유사궤적 차영상 비교 화면이며 화이트화소가 확실히 감소하는 것을 볼 수 있다. 즉, 화이트화소가 감소했다는 것은 운행궤적이 일치하는 부분이 많다는 것이며, 유사성이 높다고 볼 수 있다.

Fig. 64부터 66까지는 실시간 차량 운행궤적과 방향각별 유사궤적 차영상을 비교한 화면이다.

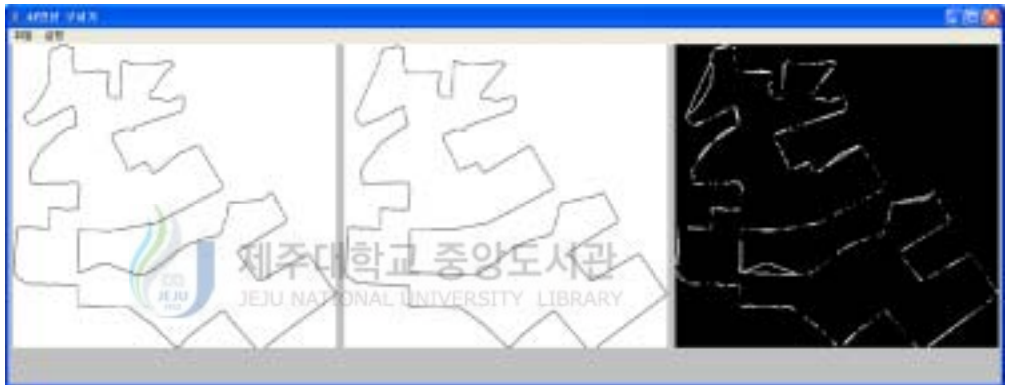


Fig. 64 The first similar trajectory difference image abstraction(10°).



Fig. 65 The first similar trajectory difference image abstraction(20°).

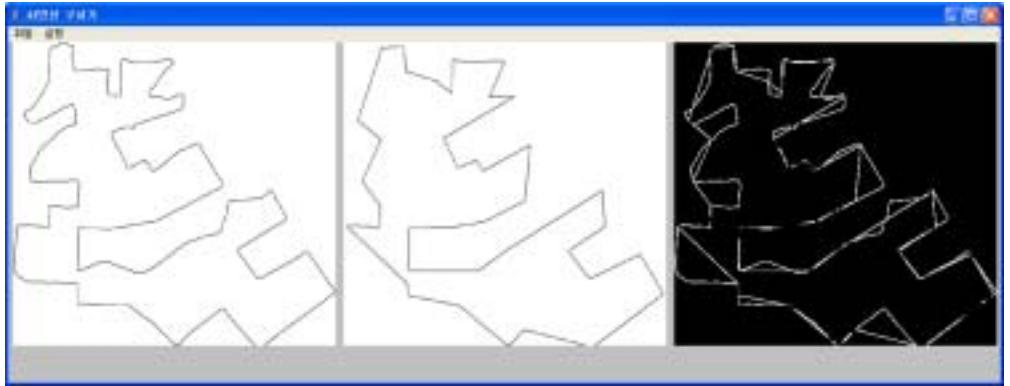


Fig. 66 The first similar trajectory difference image abstraction(30°).

Fig. 67과 Fig. 68은 1차 실시간과 하이브리드 유사궤적을 차영상 기법으로 화소수를 비교한 화면으로 방향각 $10^\circ+1\text{min}$ 인 하이브리드 유사궤적에서 많은 궤적에서 겹쳐지면서 블랙화소수가 늘어나는 것을 볼 수 있다.



Fig. 67 The first hybrid trajectory difference image abstraction($10^\circ+1\text{min}$)

Table 15는 실시간 차량 운행궤적, 시간간격별 운행궤적, 방향각 유사궤적 그리고 하이브리드 유사궤적을 차영상 비교를 통한 화이트화소 및 블랙화소를 비교한 테이블로써 테이블에서도 잘 나타나 있듯이 방향각 10°

에서 가장 작은 화이트화소수를 나타내면 또한 방향각 20°, 30°에서는 시간간격별 화이트화소수보다 감소하는 것을 볼 수 있다. 또한 하이브리드 유사궤적에서도 방향각 유사궤적보다 더 많은 블랙화소수를 볼 수 있다. 여기서 B는 Black, W는 White 색상을 나타내는 것이다.

Fig. 69부터 Fig. 76까지는 2차 실시간과 시간간격, 방향각, 하이브리드 유사궤적을 차영상 기법으로 비교 분석한 화면들이다. 그리고 Fig. 77, 78은 하이브리드 유사궤적을 차영상 기법으로 표현한 화면이다.

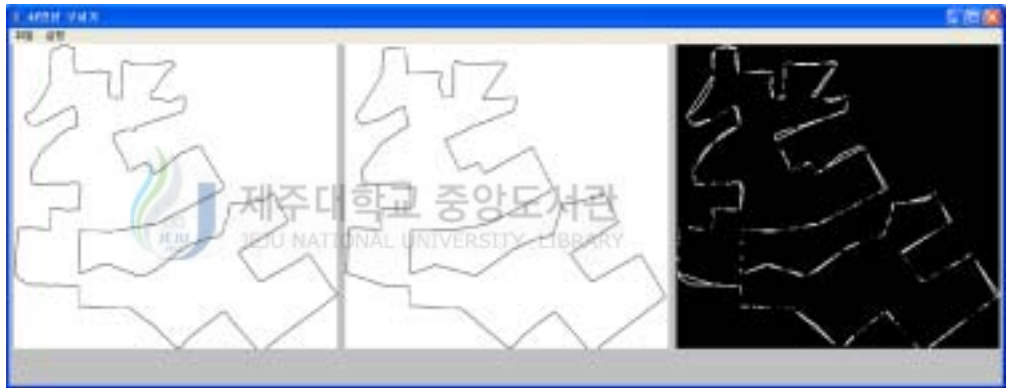


Fig. 68 The first hybrid trajectory difference image abstraction(20°+1min)

Table 15. Comparison of the number of pixels in the first vehicles running trajectories.

시간간격											
1min		2min		3min		4min		5min			
B	W	B	W	B	W	B	W	B	W	B	W
144,255	6,145	144,063	6,337	144,148	6,252	144,166	6,234	144,383	6,017		
유사궤적						하이브리드 궤적				실시간	
10°		20°		30°		10°+min		20°+1min			
B	W	B	W	B	W	B	W	B	W	B	W
146,961	3,439	145,646	4,754	144,653	5,747	147,222	3,178	146,186	4,214	146,306	4,094



Fig. 69 The second time interval difference image abstraction(1min).



Fig. 70 The second time interval difference image abstraction(2min).

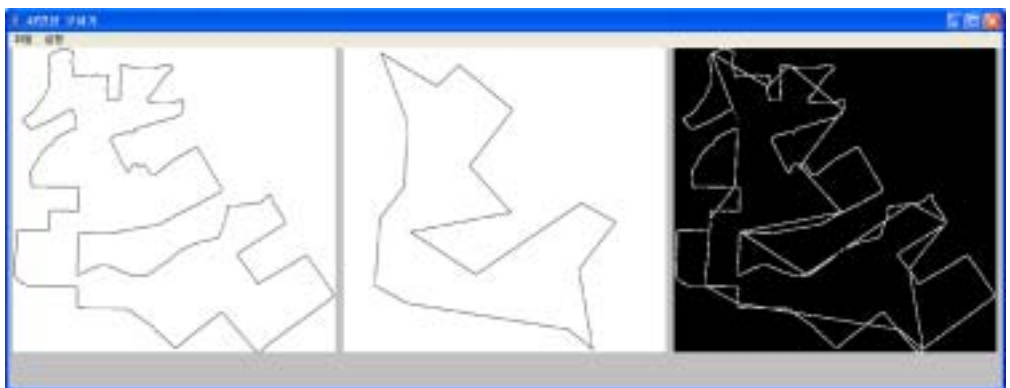


Fig. 71 The second time interval difference image abstraction(3min).

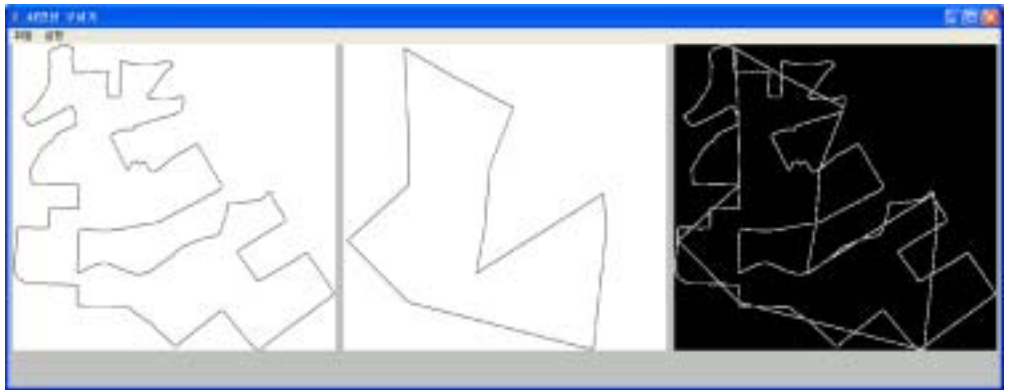


Fig. 72 The second time interval difference image abstraction(4min).



Fig. 73 The second time interval difference image abstraction(5min).

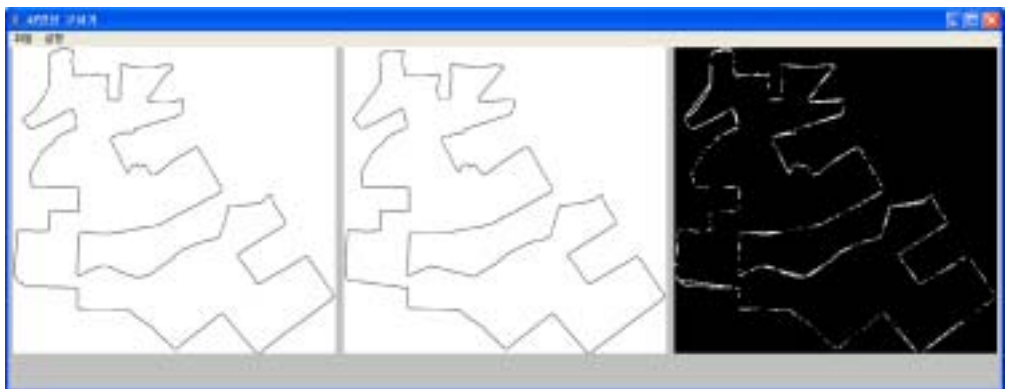


Fig. 74 The second similar trajectory difference image abstraction(10°).

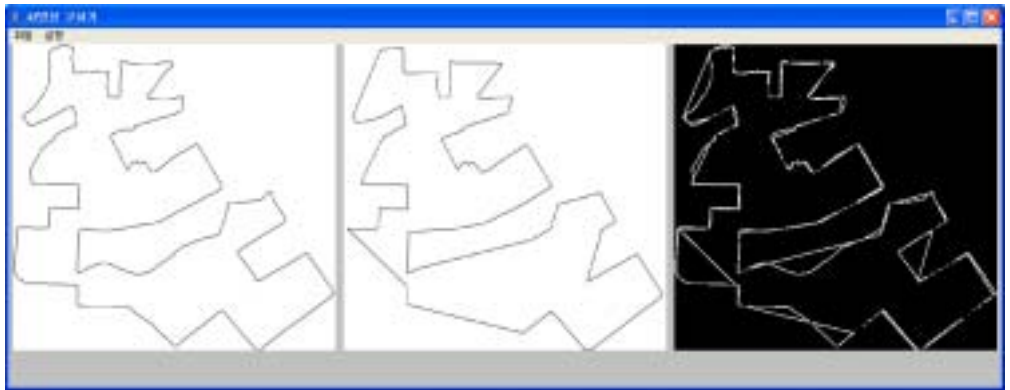


Fig. 75 The second similar trajectory difference image abstraction(20°).



Fig. 76 The second similar trajectory difference image abstraction(30°).

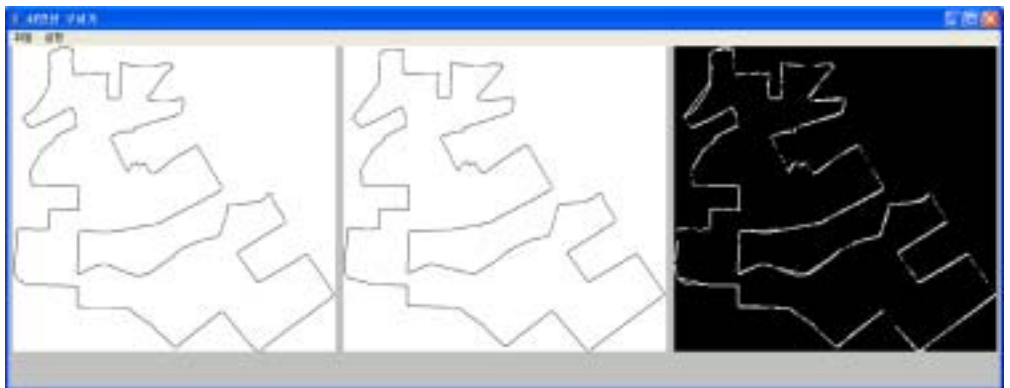


Fig. 77 The second hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).

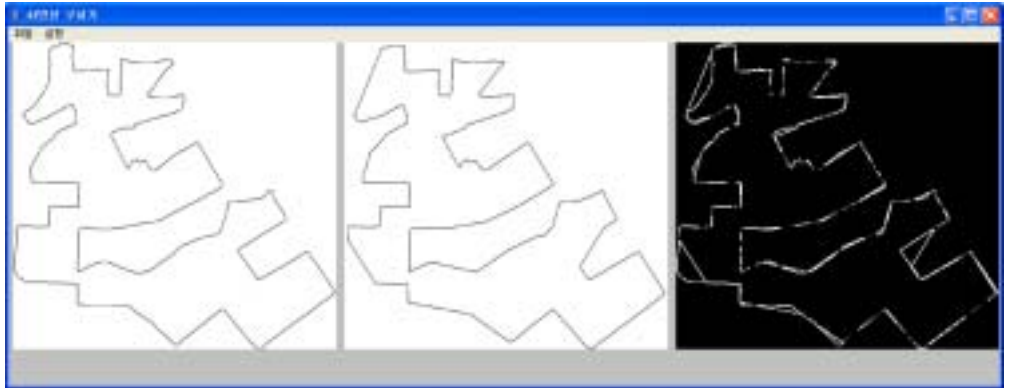


Fig. 78 The second hybrid trajectory difference image abstraction($20^{\circ}+1\text{min}$).

Table 16은 2차 실시간, 시간간격별, 방향각별, 하이브리드 유사궤적 화소수를 비교한 테이블이며 이미지 크기는 400×379 로써, 총 화소수는 151,600화소이다.

Fig. 79부터 Fig. 88까지는 3차 차영상 비교 화면으로 이미지 크기는 480×380 이며 화소수는 152,000이다.

Table 16. Comparison of the number of pixels in the second vehicles running trajectories.

시간간격											
1min		2min		3min		4min		5min			
B	W	B	W	B	W	B	W	B	W	B	W
145,104	6,496	145,156	6,444	145,307	6,293	145,570	6,030	145,568	6,032		
유사궤적						하이브리드 궤적				실시간	
10°		20°		30°		10°+min		20°+1min			
B	W	B	W	B	W	B	W	B	W	B	W
148,172	3,428	145,705	5,895	144,917	6,683	146,653	4,947	146,701	4,899	147,501	4,099

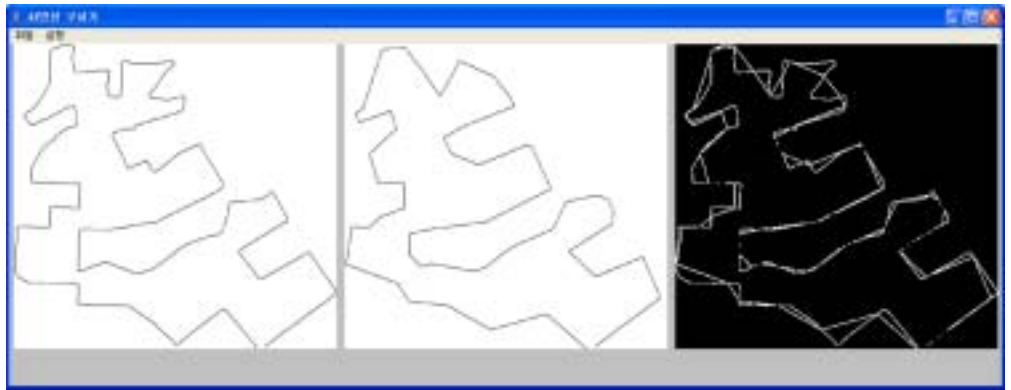


Fig. 79 The third time interval difference image abstraction(1min).

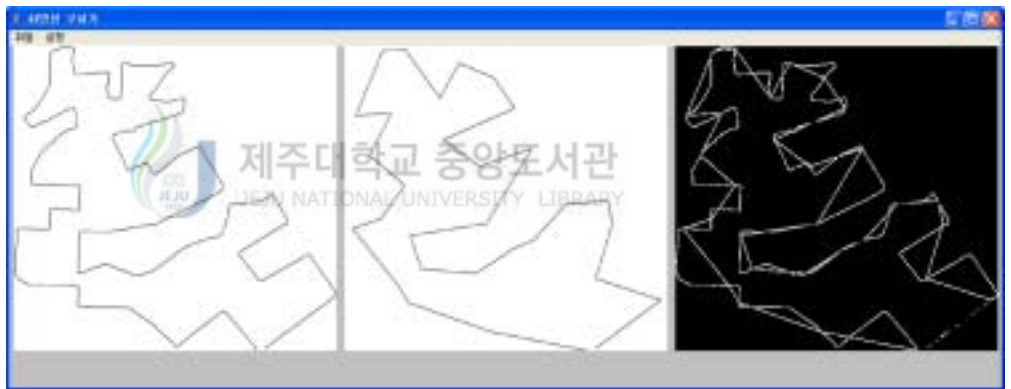


Fig. 80 The third time interval difference image abstraction(2min).

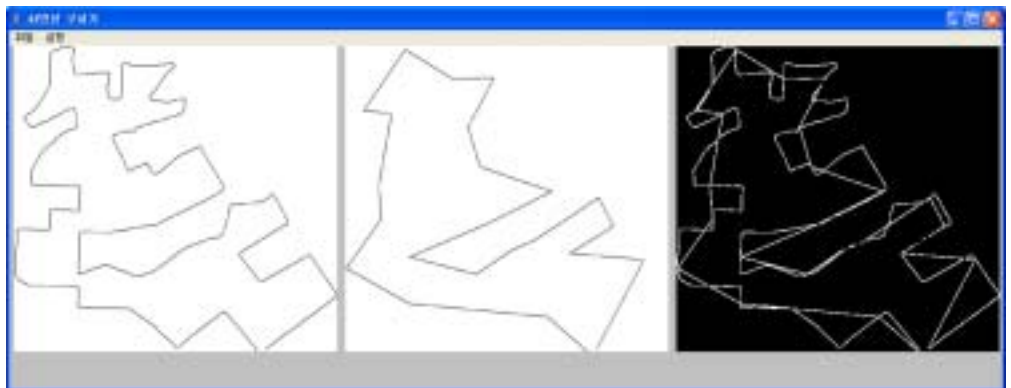


Fig. 81 The third time interval difference image abstraction(3min).

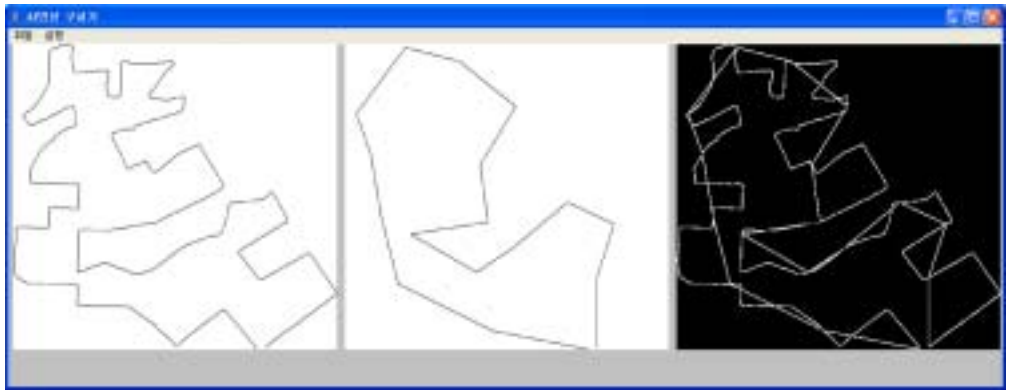


Fig. 82 The third time interval difference image abstraction(4min).



Fig. 83 The third time interval difference image abstraction(5min).

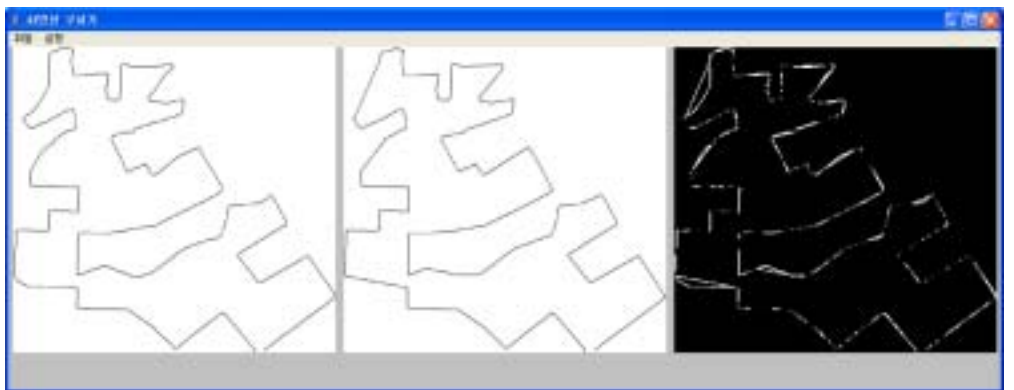


Fig. 84 The third similar trajectory difference image abstraction(10°).

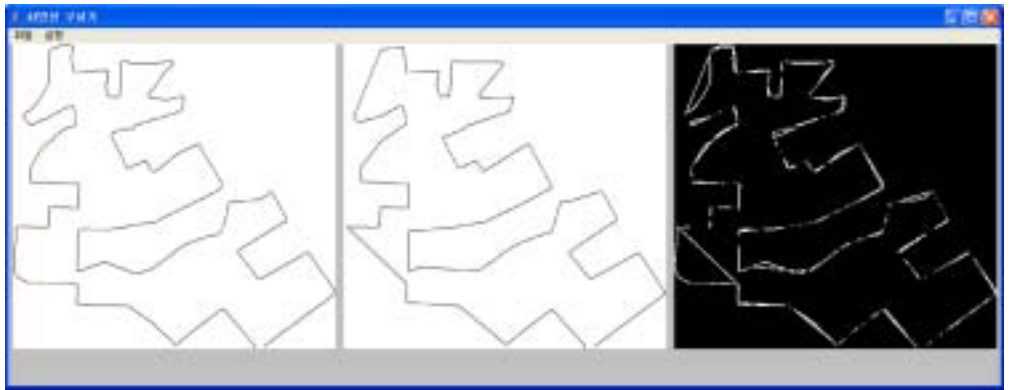


Fig. 85 The third similar trajectory difference image abstraction(20°).



Fig. 86 The third similar trajectory difference image abstraction(30°).

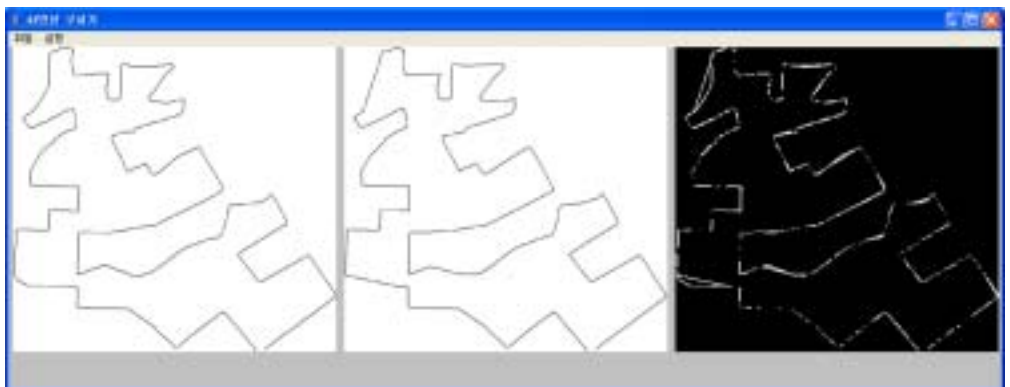


Fig. 87 The third hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).

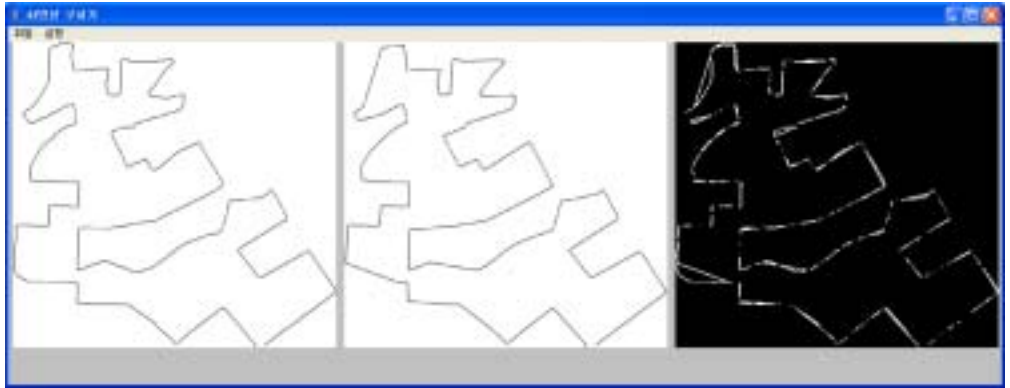


Fig. 88 The third hybrid trajectory difference image abstraction($20^{\circ}+1\text{min}$).

Table 17은 3차 실시간, 시간간격별, 방향각별, 하이브리드 유사궤적 화소수를 비교하여 정리한 테이블로서 이미지 크기는 400x380으로, 총 화소수는 152,000화소이다.

Fig. 89부터 Fig. 83까지는 4차 차영상 비교 화면으로 이미지 크기는 480x377이며 총 화소수는 150,800이다.

Table 17. Comparison of the number of pixels in the third vehicles running trajectories.

시간간격											
1min		2min		3min		4min		5min			
B	W	B	W	B	W	B	W	B	W	B	W
145,802	6,198	145,615	6,385	145,607	6,393	145,981	6,019	145,858	6,142		
유사궤적						하이브리드 궤적				실시간	
10°		20°		30°		10°+min		20°+1min			
B	W	B	W	B	W	B	W	B	W	B	W
148,322	3,678	147,270	4,730	145,918	6,082	148,254	3,746	147,781	4,219	147,907	4,093

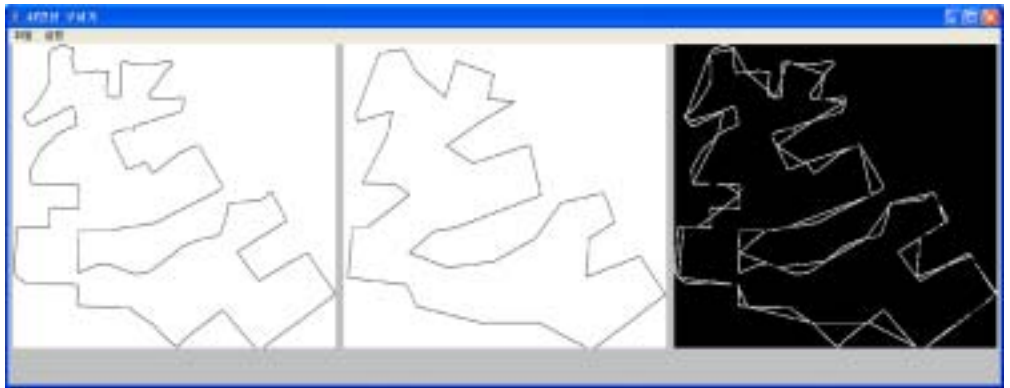


Fig. 89 The 4th time interval difference image abstraction(1min).

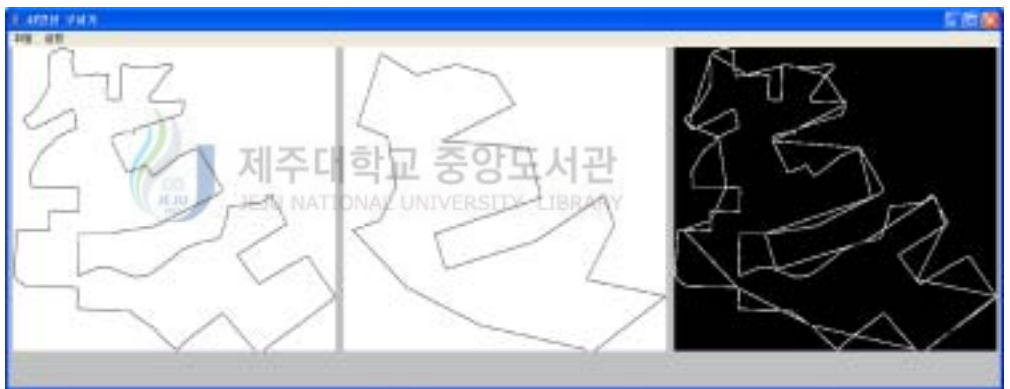


Fig. 90 The 4th time interval difference image abstraction(2min).

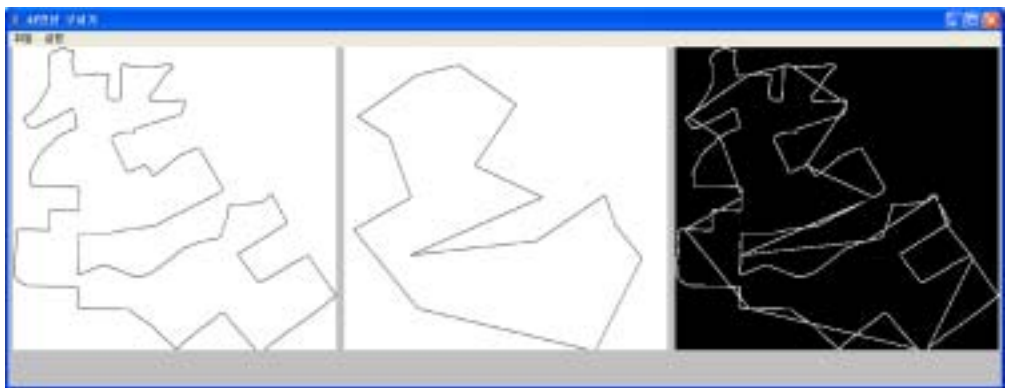


Fig. 91 The 4th time interval difference image abstraction(3min).

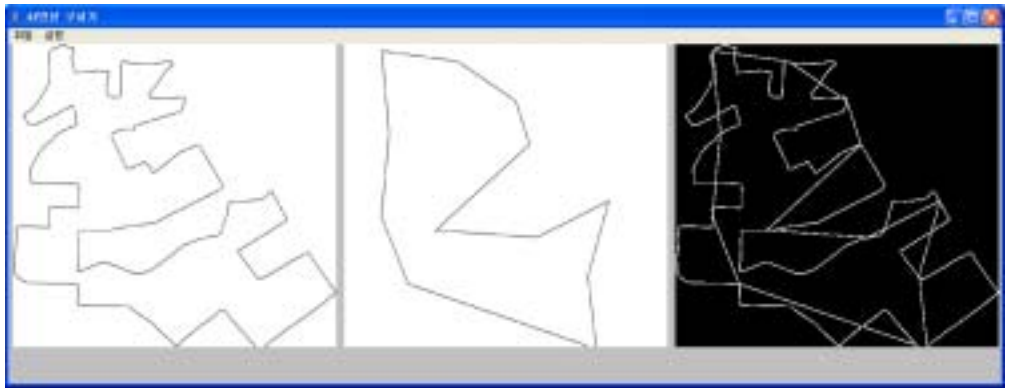


Fig. 92 The 4th time interval difference image abstraction(4min).



Fig. 93 The 4th time interval difference image abstraction(5min).

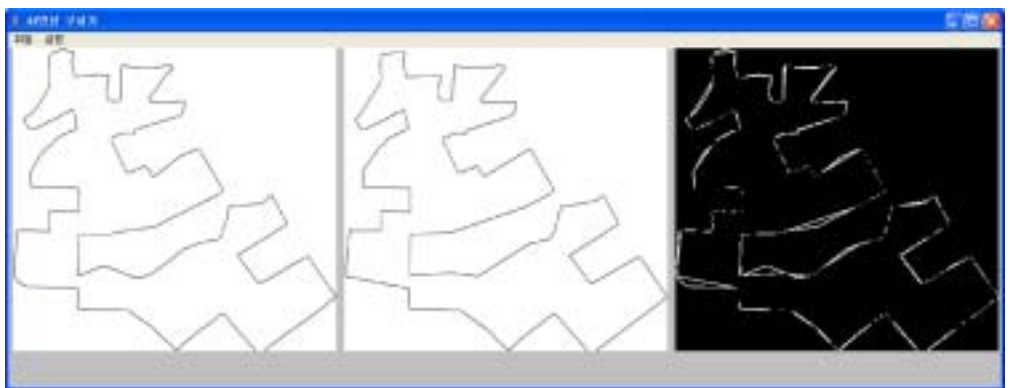


Fig. 94 The 4th similar trajectory difference image abstraction(10°).

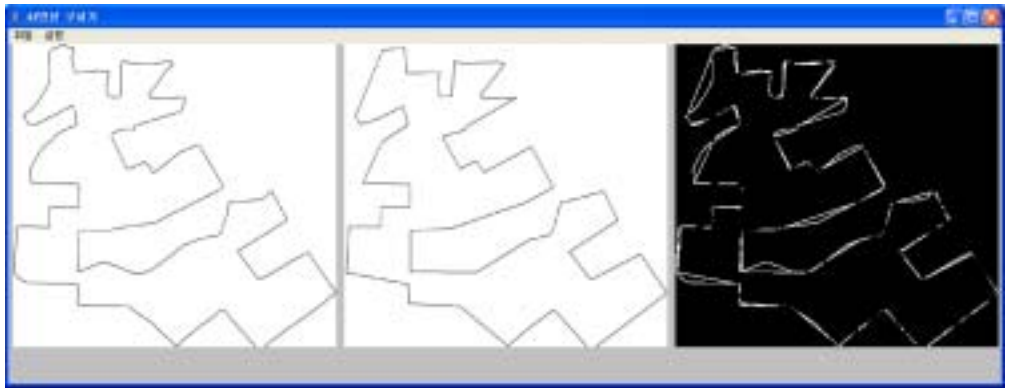


Fig. 95 The 4th similar trajectory difference image abstraction(20°).



Fig. 96 The 4th similar trajectory difference image abstraction(30°).



Fig. 97 The 4th hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).

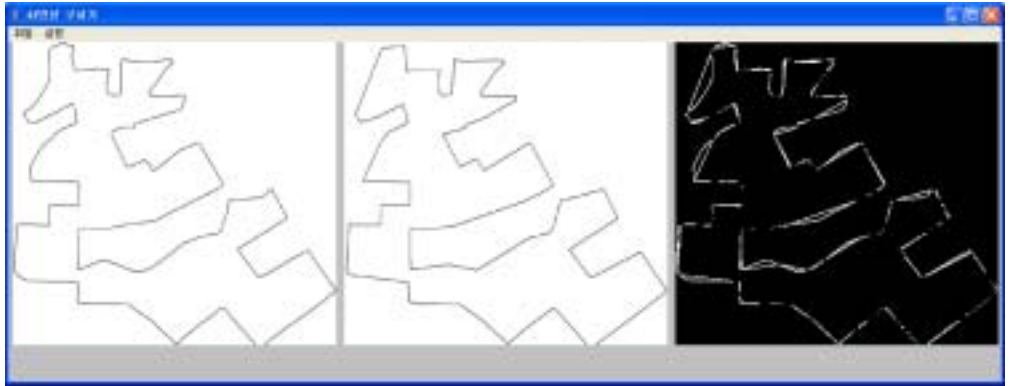


Fig. 98 The 4th hybrid trajectory difference image abstraction($20^{\circ}+1\text{min}$).

Table 18은 4차 실시간, 시간간격별, 방향각별, 하이브리드 유사궤적 화소수를 비교하여 정리한 테이블이다.

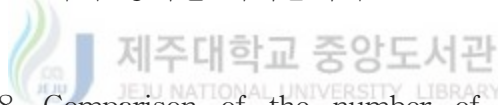


Table 18. Comparison of the number of pixels in the 4th vehicles running trajectories.

시간간격											
1min		2min		3min		4min		5min			
B	W	B	W	B	W	B	W	B	W	B	W
144,321	6,479	144,305	6,495	144,442	6,358	144,738	6,062	144,710	6,090		
유사궤적						하이브리드 궤적				실시간	
10°		20°		30°		10°+min		20°+1min			
B	W	B	W	B	W	B	W	B	W	B	W
146,528	4,272	145,692	5,108	144,773	6,027	146,781	4,019	146,119	4,681	146,730	4,070

Fig. 84부터 Fig. 91까지는 5차 차영상 비교 화면으로 이미지 크기는 480x379이며 총 화소수는 151,600이다.



Fig. 99 The 5th time interval difference image abstraction(1min).



Fig. 100 The 5th time interval difference image abstraction(2min).

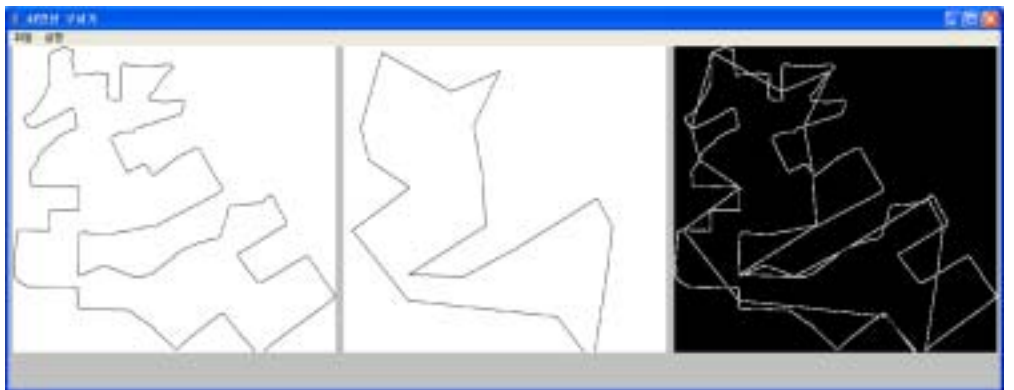


Fig. 101 The 5th time interval difference image abstraction(3min).

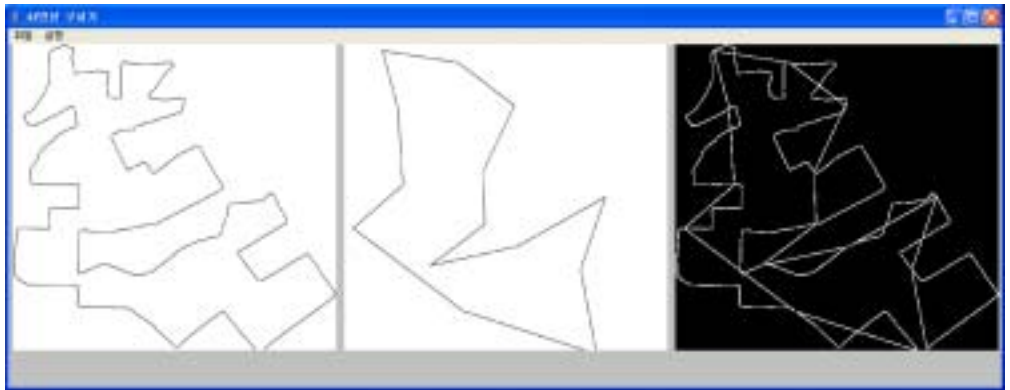


Fig. 102 The 5th time interval difference image abstraction(4min).



Fig. 103 The 5th time interval difference image abstraction(5min).

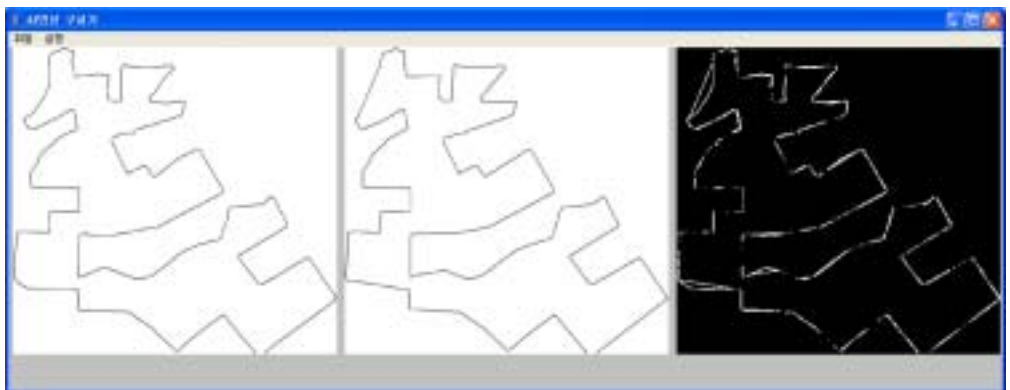


Fig. 104 The 5th similar trajectory difference image abstraction(10°).

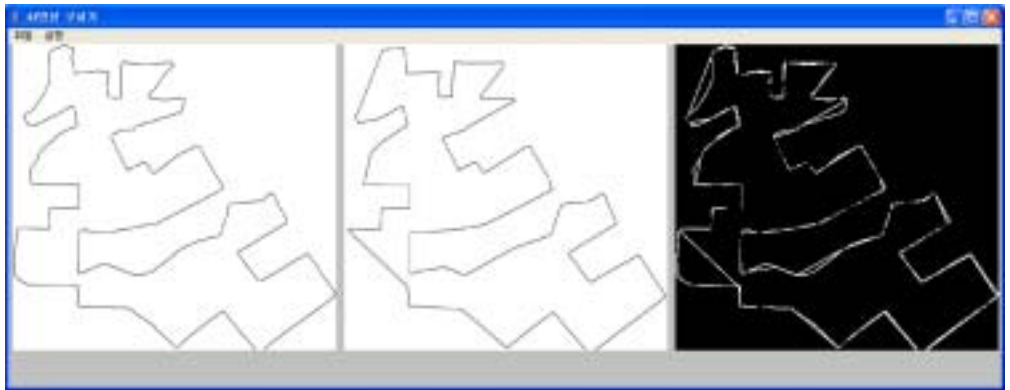


Fig. 105 The 5th similar trajectory difference image abstraction(20°).



Fig. 106 The 5th similar trajectory difference image abstraction(30°).



Fig. 107 The 5th hybrid trajectory difference image abstraction($10^\circ+1\text{min}$).

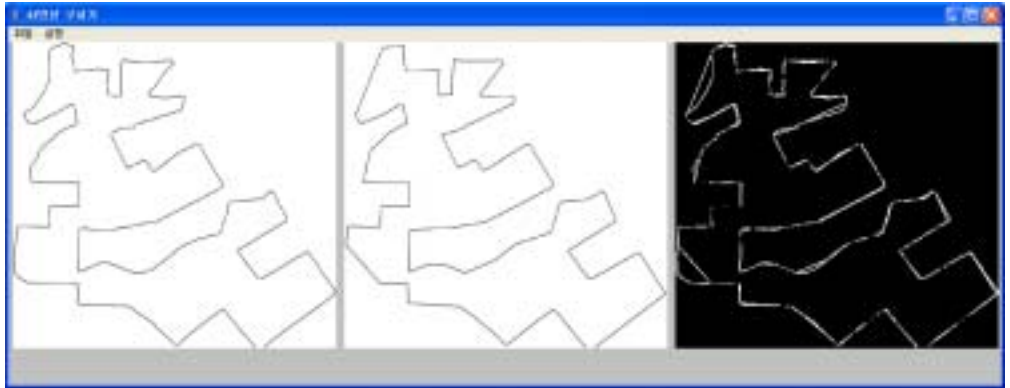


Fig. 108 The 5th hybrid trajectory difference image abstraction($20^{\circ}+1\text{min}$).

Table 19는 5차 실시간, 시간간격별, 방향각별, 하이브리드 유사궤적 화소수를 비교하여 정리한 테이블이다.

Fig. 109와 Fig. 110은 각 차수의 블랙, 화이트 화소수를 비교한 그래프이다.

Table 19. Comparison of the number of pixels in the 5th vehicles running trajectories.

시간간격											
1min		2min		3min		4min		5min			
B	W	B	W	B	W	B	W	B	W	B	W
145,096	6,504	144,856	6,744	145,285	6,315	145,382	6,218	145,566	6,034		
유사궤적						하이브리드 궤적				실시간	
10°		20°		30°		10°+min		20°+1min			
B	W	B	W	B	W	B	W	B	W	B	W
147,435	4,165	146,236	5,364	145,835	5,765	147,643	3,957	146,634	4,966	147,490	4,110

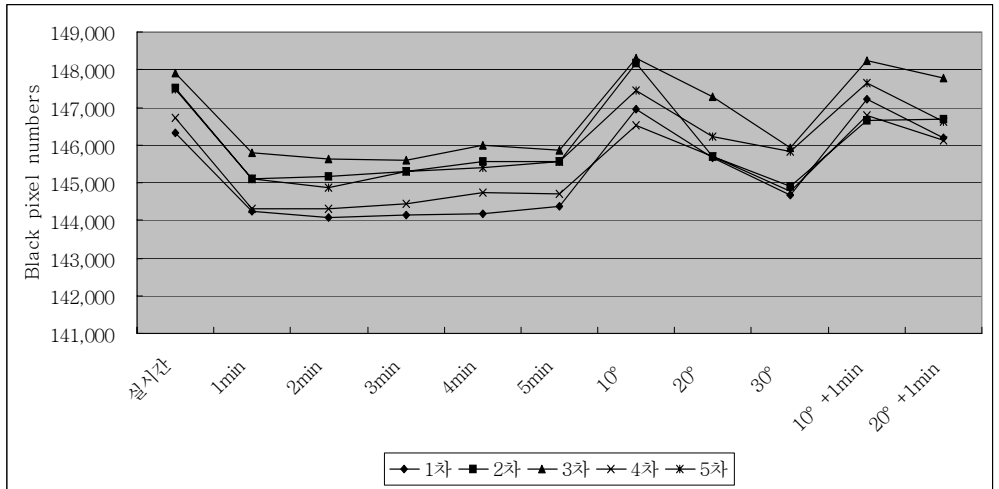


Fig. 109 Comparison graph of the number of black pixels.

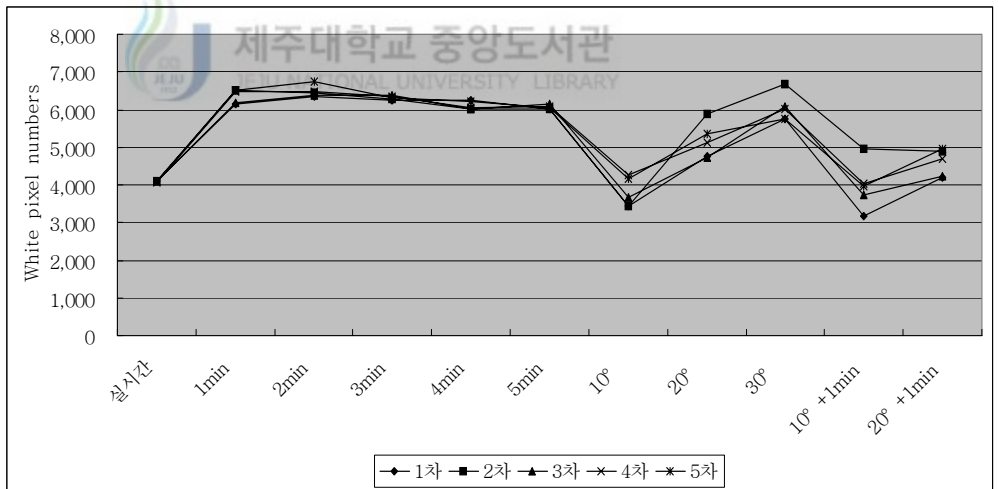


Fig. 110 Comparative graph of the number of white pixels.

(4) 차량 운행 궤적의 정량적 비교

차량 운행 궤적 정량적 비교는 실시간 차량 운행 시 수신된 차량 위치 정보량과 시간간격별 생성된 차량 위치 정보량 그리고 방향각과 하이브리드 기법을 통해 생성된 차량 위치정보량을 정량적 비교를 통하여 제안한

하이브리드기법의 효율성을 제시하였다.

Table 20부터 24까지는 1차부터 5차까지의 차량 운행에 따라 생성된 정보량을 정리하였다.

Fig. 111, 112는 위에서 서술한 Table 20부터 24까지의 각 차수의 파일 크기와 레코드 수를 비교한 내용을 그래프로 표현하였다.

Table 20. Quantitative comparison between time interval and similar trajectory(first).

구분	Time Interval						유사궤적			하이브리드	
	1sec	1min	2min	3min	4min	5min	10°	20°	30°	10°+1	20°+1
파일크기 (kb)	133	3.76	1.98	1.49	1.20	0.99	25.05	12.49	6.59	27.47	14.62
레코드수	1,870	51	26	19	15	12	351	174	91	385	206
운행거리	22.664km		운행시간			3,201초		평균속도		25.5km/h	

Table 21. Quantitative comparison between time interval and similar trajectory(second).

구분	Time Interval						유사궤적			하이브리드	
	1sec	1min	2min	3min	4min	5min	10°	20°	30°	10°+1	20°+1
파일크기 (kb)	170	3.83	2.12	1.48	1.20	0.98	31.74	14.31	8.35	33.94	17.15
레코드수	2,399	52	28	19	15	12	445	200	116	476	240
운행거리	22.986km		운행시간			3,164초		평균속도		26.3km/h	

Table 22. Quantitative comparison between time interval and similar trajectory(third).

구분	Time Interval						유사궤적			하이브리드	
	1sec	1min	2min	3min	4min	5min	10°	20°	30°	10°+1	20°+1
파일크기 (kb)	152	4.83	2.55	1.84	1.41	1.20	31.50	15.64	8.53	33.55	19.33
레코드수	2,143	66	34	24	18	15	441	218	118	484	270
운행거리	22.791km		운행시간			3,960초		평균속도		20.7km/h	

Table 23. Quantitative comparison between time interval and similar trajectory(4th).

구분	Time Interval						유사궤적			하이브리드	
	1sec	1min	2min	3min	4min	5min	10°	20°	30°	10°+1	20°+1
파일크기 (kb)	124	3.70	2.00	1.30	1.10	0.91	23.00	11.00	6.40	25.52	14.15
레코드수	1,754	50	26	17	14	11	314	158	188	356	197
운행거리	22.567km		운행시간			2,884초		평균속도		28.2km/h	

Table 24. Quantitative comparison between time interval and similar trajectory(5th).

구분	Time Interval						유사궤적			하이브리드	
	1sec	1min	2min	3min	4min	5min	10°	20°	30°	10°+1	20°+1
파일크기 (kb)	133.9	4.10	2.20	1.60	1.20	0.99	25.80	13.60	7.50	28.39	16.56
레코드수	1,878	56	29	20	15	12	361	189	104	397	231
운행거리	22.761km		운행시간			3,276초		평균속도		25.0km/h	

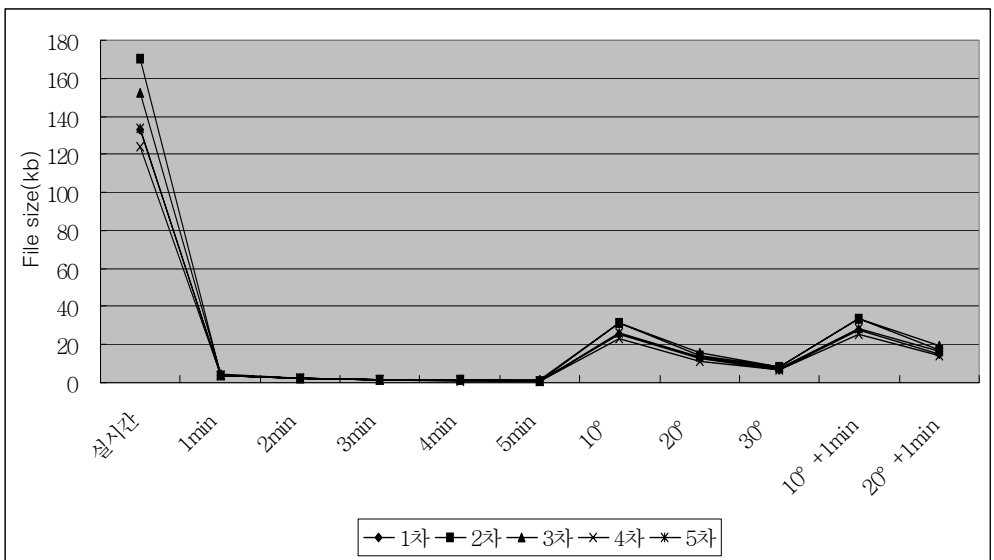


Fig. 111 Comparative graph of the file sizes between time interval and similar trajectory.

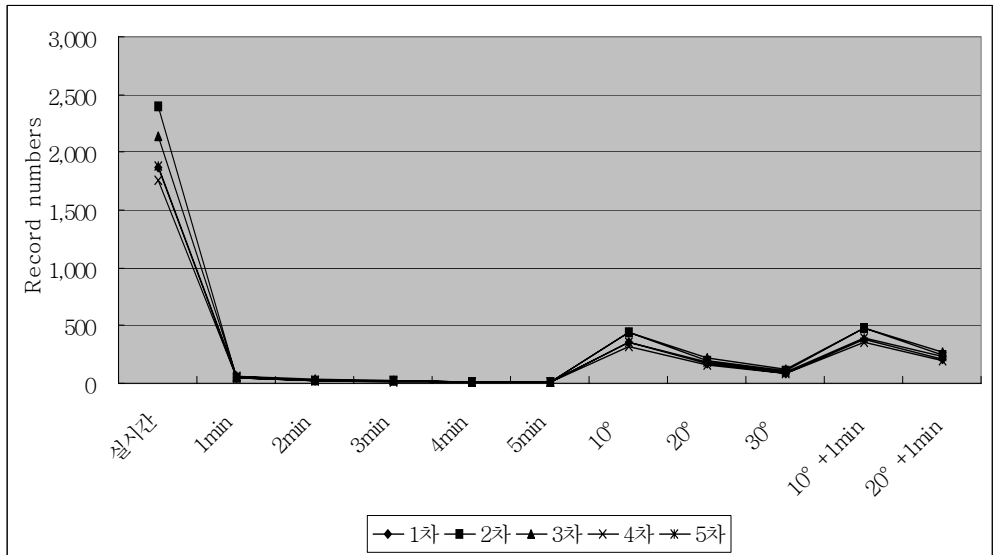


Fig. 112 Comparative graph of the number of records between time Interval and similar trajectory.



(5) 일관성 비교

차량 운행궤적은 동일한 경로를 운행했을 경우에는 동일한 운행 궤적을 그려야 한다. 즉 일관성 비교는 매 측정할 때 마다 동일한 운행궤적을 그릴 수 있는지를 알아보는 것이다. 따라서 시간간격별 운행궤적과 방향각별, 하이브리드 유사궤적의 일관성을 비교·분석하였다.

시간간격 차량 운행궤적은 시시각각으로 변하는 도로상황과 교통상황에 따라 전혀 다른 운행 궤적을 그린다. 하지만 본 연구에서 제안한 방향각과 시간간격을 이용한 하이브리드 기법과 방향각은 동일한 경로를 운행했을 경우에는 동일한 궤적을 그리는 것을 볼 수 있었다.

일관성을 비교하기 위해 1차부터 5차까지의 각 시간간격 운행궤적을 시각적으로 표현하여 방향각과 하이브리드 기법의 일관성이 우수함을 보였다.

Fig. 113은 GPS TrackMaker를 통하여 1차부터 5차까지의 각 시간간격

별 차량 운행궤적을 표현한 화면으로 시각적으로 일관성을 비교·분석하였다.

비교화면을 보면 각 시간간격별 일관성을 보면 매번 다른 운행궤적을 그리는 것을 볼 수 있었다.

Fig. 114는 방향각 10° , 20° , 30° 의 유사궤적 일관성을 비교한 화면으로, Fig. 114의 방향각 10° 를 보면 거의 동일한 운행 궤적을 그리고 있어 일관성이 매우 높은 것으로 나타나고 있다. Fig. 114의 방향각 20° , 30° 은 방향각 10° 보다는 다소 일관성이 떨어지지만 거의 동일한 운행궤적을 보이고 있다.

Fig. 115는 하이브리드 유사궤적의 일관성을 보인 것으로 방향각 10° 와 마찬가지로 하이브리드 유사궤적(방향각 $10^\circ+1\text{min}$)은 높은 일관성을 시각적으로 보였으며, 하이브리드 유사궤적(방향각 $20^\circ+1\text{min}$)은 다소 일관성이 떨어지지만 방향각 20° , 30° 보다는 일관성이 다소 높게 보이고 있다.

Fig. 116은 1차부터 5차까지 차량 운행궤적 중 일관성이 많이 떨어지는 부분을 확대한(축척 1:5m)부분으로, 이것은 GPS 수신 위치정보에 의한 오차라고 볼 수 있다. 즉, 차량 속도 변화에 따른 차량 수신 좌표값 변화, 도로 교통에 따른 차량 수신 좌표값의 변화, 그리고 GPS수신 불량지역에서의 GPS수신 오차에 따른 제반적인 상황에 따라 방향각 계산 시 방향각의 변화에 의해서 일관성 차이를 보이고 있다.

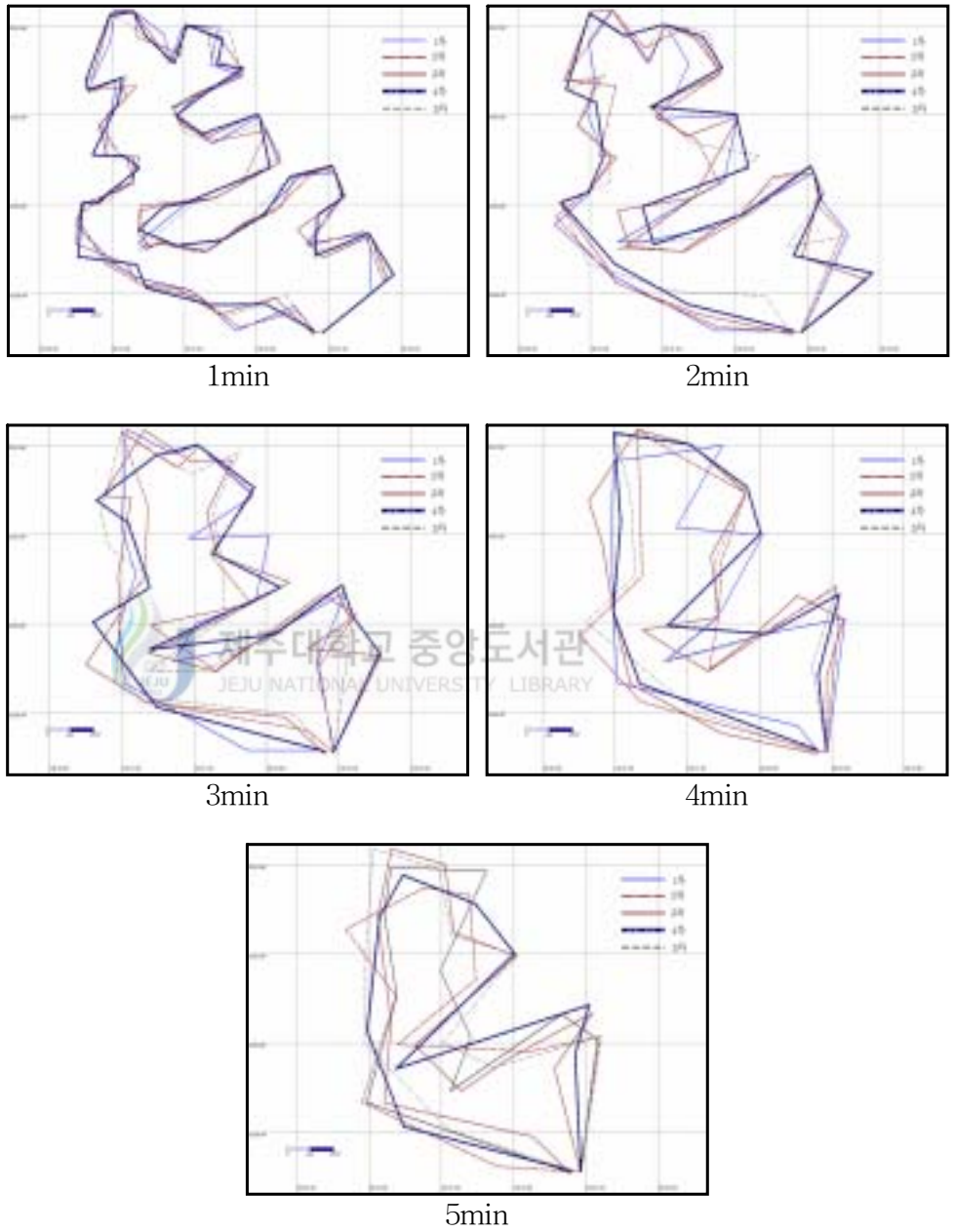


Fig. 113 Time Interval consistency comparison.

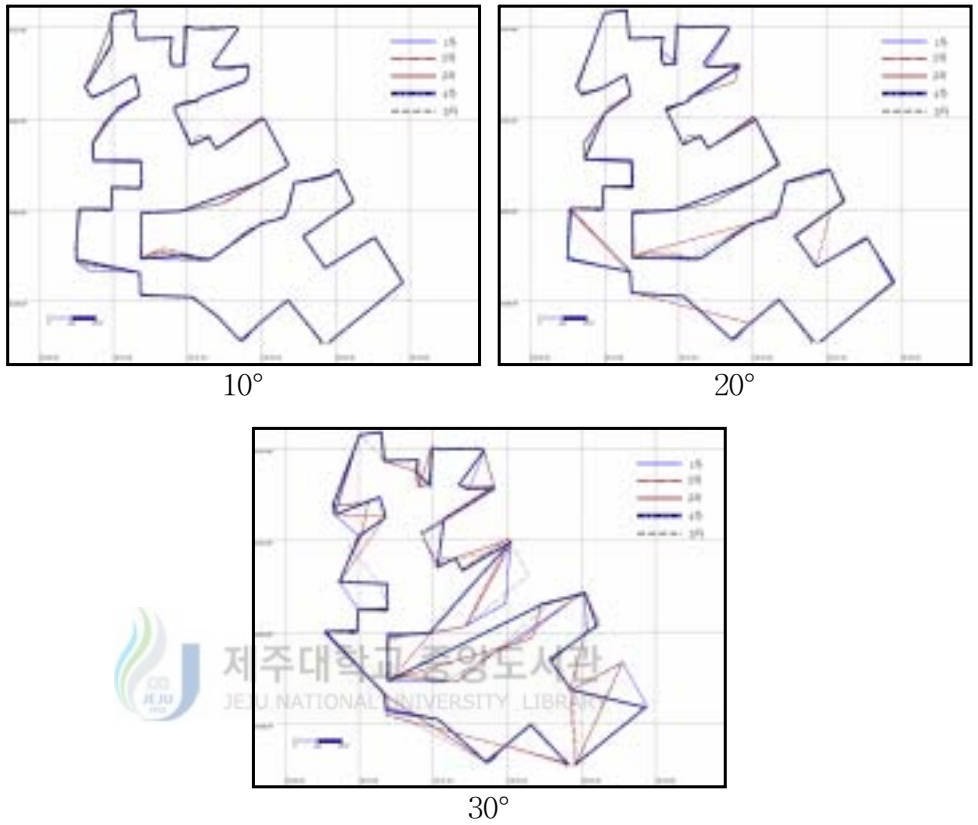


Fig. 114 Similar trajectory consistency comparison.

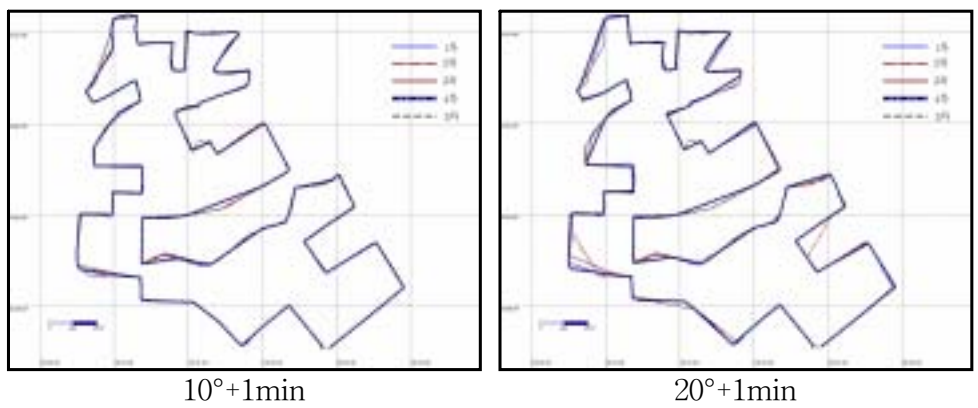


Fig. 115 Hybrid trajectory consistency comparison.

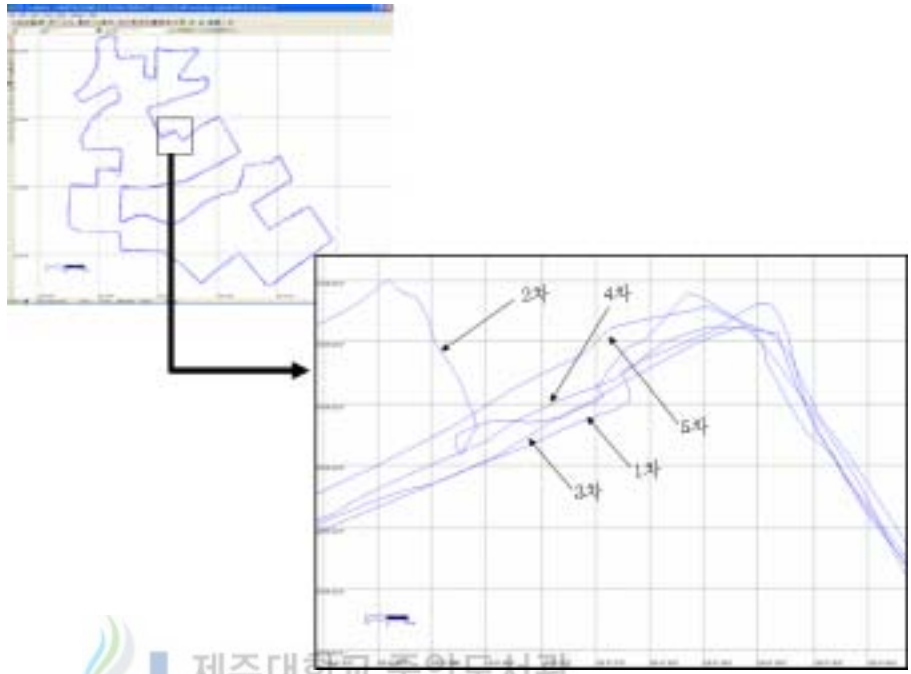


Fig. 116 GPS location information change by vehicles running.

(6) 운행 거리 비교

동일한 경로를 운행한 차량은 동일한 운행 거리를 갖는다. 즉 실시간으로 운행한 차량의 운행 거리와 시간간격별, 방향각별 유사궤적의 운행 거리 그리고 하이브리드 유사궤적의 차량 운행거리를 비교하여 방향각과 하이브리드 유사궤적의 성능을 평가하였다.

Table 25는 실시간, 시간간격, 방향각, 하이브리드의 운행거리를 정리한 테이블이다.

Table 25의 실시간 운행거리와 방향각 유사궤적의 운행거리를 보면 400m에서 500m 정도의 적은 차이를 보이고 있다. 하지만 1분 시간간격의 차량 운행 거리의 차이를 비교하였을 경우 약 4~5km정도의 차이를 보이고 있다.

하이브리드의 운행거리도 거의 실시간에 가까운 차량 운행거리를 보이고 방향각 차량 운행거리보다 더 실시간에 가깝다.

Table 26은 Table 25의 실시간 운행거리와 시간간격별, 방향각별, 하이브리드 운행 거리를 백분율(%)로 표현 한 것이다.

Table 25. Vehicle running distance. (units : km)

구분	1차	2차	3차	4차	5차
실시간	22.664	22.986	22.791	22.567	22.582
1min	18.462	18.389	18.984	18.665	18.573
2min	14.716	15.344	16.98	15.782	17.321
3min	13.918	13.833	15.505	14.158	13.823
4min	12.76	12.017	12.626	11.775	12.623
5min	11.252	11.512	12.45	11.779	11.628
방향각10°	22.19	22.565	22.276	22.151	22.194
방향각20°	21.716	21.383	21.671	21.766	21.587
방향각30°	18.785	18.431	18.942	19.905	19.905
하이브리드(방향각10°+1min)	22.279	22.582	22.355	22.190	22.257
하이브리드(방향각20°+1min)	21.934	21.98	22.015	21.871	21.849

Table 26. Percentage of vehicles running distance. (units : %)

구분	1차	2차	3차	4차	5차
1min	81.46	80.00	83.30	82.71	82.25
2min	64.93	66.75	74.50	69.93	76.70
3min	61.41	60.18	68.03	62.74	61.21
4min	56.30	52.28	55.36	52.18	55.85
5min	49.65	50.08	54.63	52.20	51.49
방향각10°	97.91	98.17	97.74	98.16	98.28
방향각20°	95.82	93.03	95.09	96.45	95.59
방향각30°	82.88	80.18	83.11	88.20	82.01
하이브리드(방향각10°+1min)	98.30	98.24	98.09	98.33	98.56
하이브리드(방향각20°+1min)	96.78	95.62	96.60	96.92	96.75

Fig. 117과 Fig. 118은 실시간 차량 운행거리와 시간간격, 방향각, 하이브리드의 운행 거리를 그래프로 표현 한 것으로, 차량 운행거리 그래프와 차량 운행거리 백분율 그래프이다.

차량 운행거리 그래프와 백분율 그래프에서도 방향각과 하이브리드 유사 궤적의 차량 운행거리는 실시간 차량 운행거리와 거의 동일한 위치에 그래프가 존재하였으며, 백분율에서는 실시간 차량 운행거리의 97 ~ 98%를 표현 할 수 있었다.

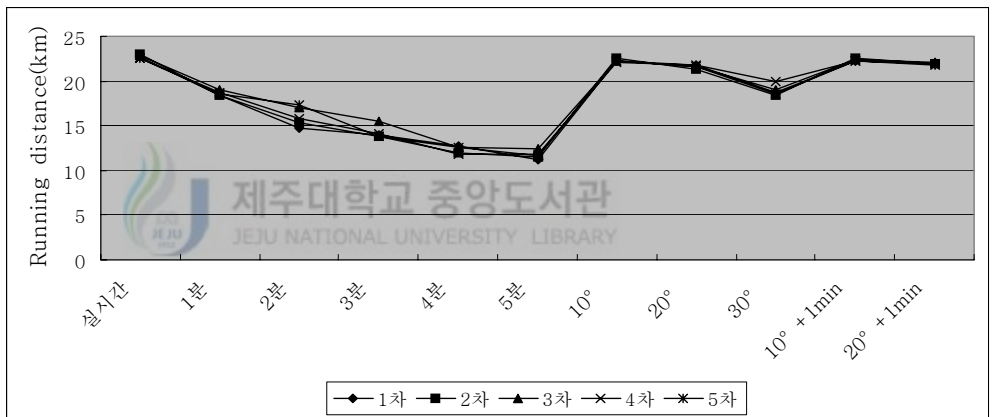


Fig.117 Vehicle running distance.

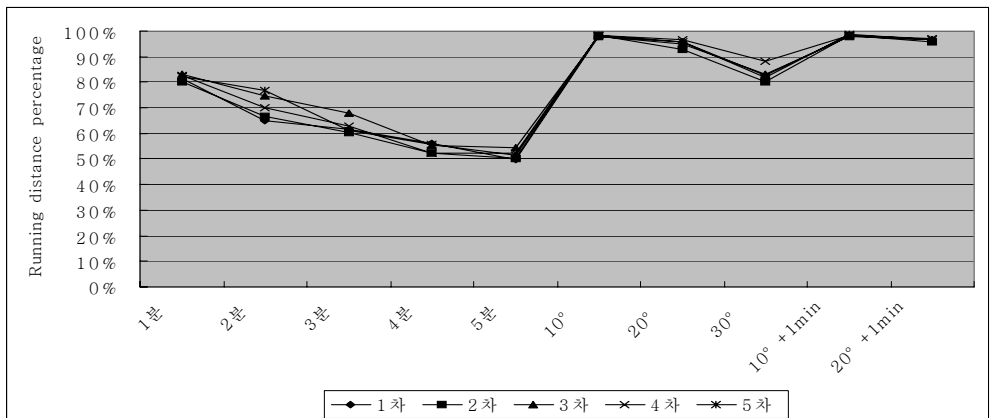


Fig. 118 Percent of vehicle running distance.

(7) 직선 도로에서의 유사궤적

본 목(目)에서는 고속도로와 같은 직선 도로와 커브가 완만한 도로에서도 방향각과 하이브리드 추출에 의한 차량 위치정보 관리에 효율성을 제시하였다. 이를 위해 차량 위치정보와 맵과 매핑을 하기 위해 알맵이라는 프로그램을 이용하여 차량 운행 경로를 표시하였고, 이를 근거로 직접 차량을 운행하여 차량 위치정보를 획득하였다.

총 운행 거리는 20.782km이고, 평균 속도는 76.7km/h, 총 운행 시간은 16분 15초(975초) 동안 운행을 하였다.

Fig. 119는 차량 운행 경로이며, Start지점에서 출발하여 End지점까지 운행한 경로 궤적을 보여주고 있다.

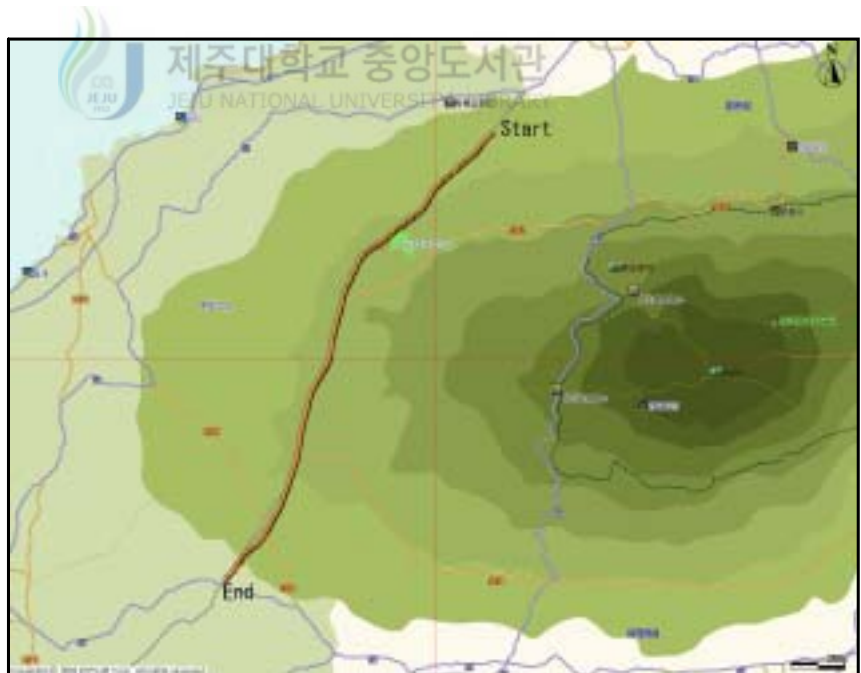


Fig. 119 Vehicle running path.

Fig. 120은 Fig. 119의 운행 경로 궤적 중 Start지점의 운행 경로와 운

행 경로 포인트들을 보여주고 있다.



Fig. 120 Starting area and running path point.



Fig. 121 역시 Fig. 120의 End 지점의 차량 운행 경로와 운행 경로 포인트들을 보여주고 있다.



Fig. 121 Finishing area and running path point.

Fig. 122는 Fig. 119의 차량 운행 경로 중 차량 운행궤적만 추출한 화면으로 축척은 1:1.5km이며 1초당 GPS에서 획득한 차량 위치정보로 표현한 것이다.

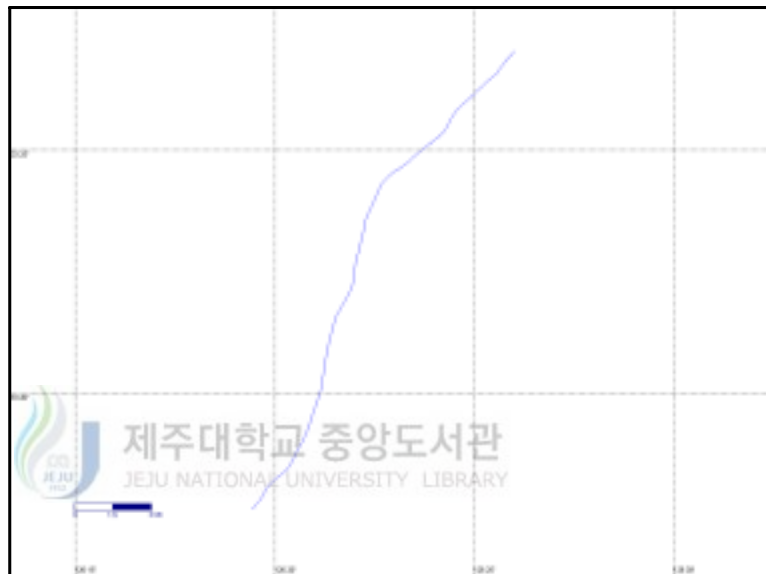


Fig. 122 Vehicle running path abstraction.

직선 도로에서의 차량 운행궤적을 차영상 및 GPS TrackMaker로 비교한 결과 실시간과 시간간격 1분에서 3분까지는 거의 동일한 운행궤적을 그렸지만 4분과 5분에서는 많은 오차가 발생하는 것을 볼 수 있었다. 그러나 방향각이 30°인 경우에도 거의 동일한 운행궤적을 그리는 것을 볼 수 있었다. 따라서 장시간 운행하는 경우 특히 방향각 및 하이브리드 기법이 파일 용량뿐만 아니라 레코드수를 비교 하더라도 효율적인 것을 볼 수 있었다.

Fig. 123은 실시간과 시간간격 4분과 5분 운행궤적을 비교한 화면이고 Fig. 124는 실시간과 방향각 30°를 비교한 화면이다.

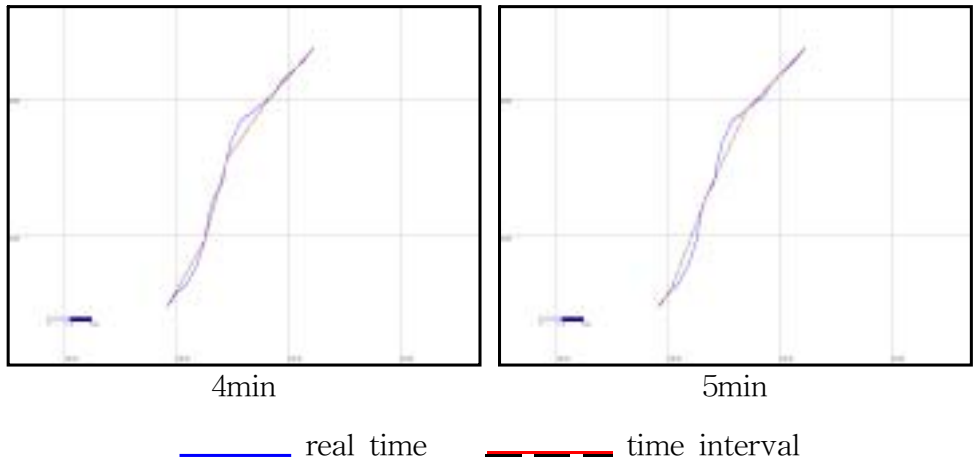


Fig. 123 Real time running trajectory and time interval running trajectory in straightaway.

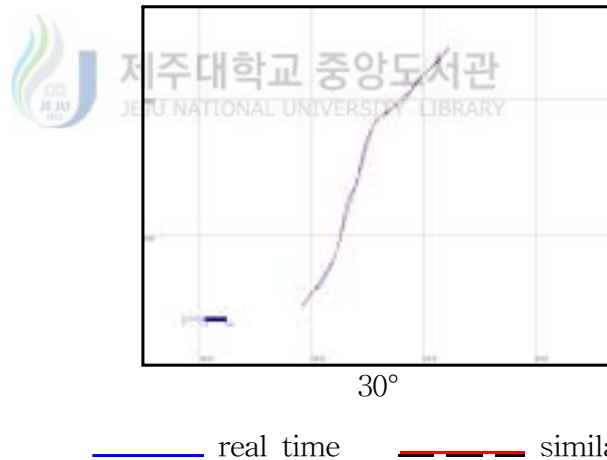


Fig. 124 Real time running trajectory and similar trajectory in straightaway.

Table 27은 시간간격과 방향각 그리고 하이브리드 유사궤적을 정량적으로 비교 분석한 테이블이며, 시간간격 1분과 방향각 30°의 파일 용량과 레코드수를 비교하면, 시간간격 1분인 경우 파일용량이 1.43kb이고 방향각 30°은 1.93kb이다. 그리고 레코드 수는 1분인 경우 18개이고 방향각

30°인 경우에는 25개이다. 따라서 Fig. 124에서 보았듯이 직선도로에서도 방향각 기법이 거의 동일한 운행궤적을 그리며 파일 크기와 레코드 수도 감소하는 것을 볼 수 있었다.

Table 27. Quantitative comparison between time interval and similar trajectories.

구분	Time Interval						유사궤적			하이브리드	
	1sec	1min	2min	3min	4min	5min	10°	20°	30°	10°+1	20°+1
파일용량 (kb)	62.82	1.43	0.85	0.64	0.565	0.495	3.37	2.29	1.93	4.45	3.37
레코드수	873	18	10	7	6	5	45	30	25	60	45
운행거리	20.782km		운행시간			975초		평균속도		76.7km/h	

Fig. 125와 Fig. 126은 파일 용량과 레코드 수를 비교한 화면이다.

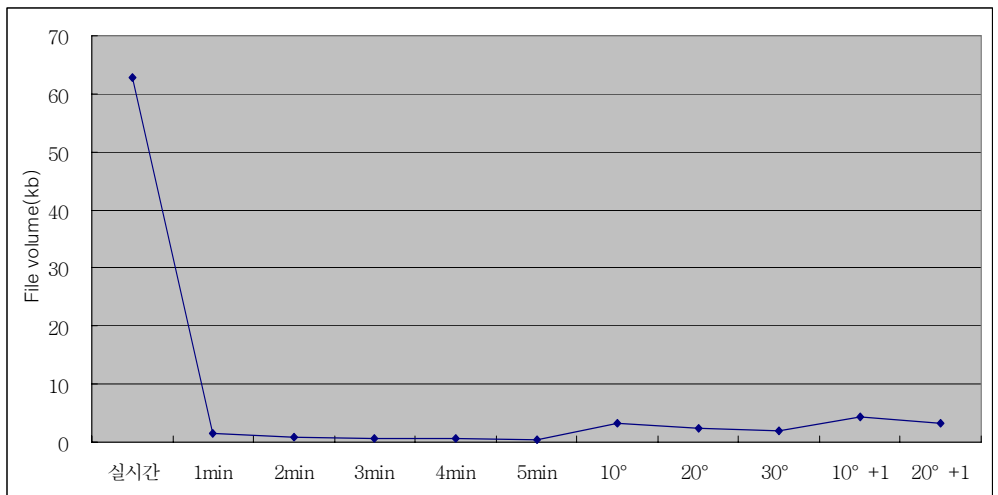


Fig. 125 Comparison of file sizes between time interval and similar trajectory in straightaway.

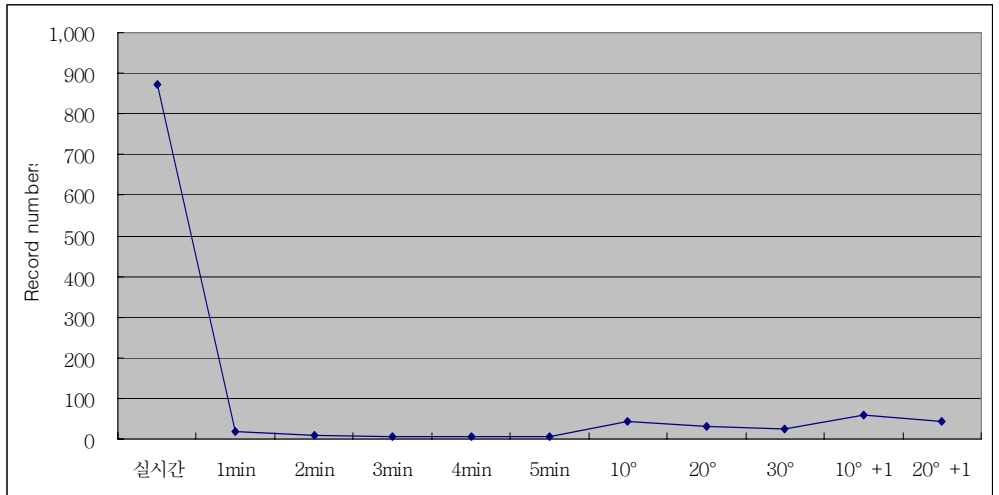


Fig. 126 Comparison of the number of records between time interval and similar trajectory in straightaway.

(8) 데이터베이스 구축

데이터베이스는 Oracle 9i를 이용하여 구축하였으며, 데이터베이스 테이블들은 4장 7절에서 설계한 차량 속성 테이블, 운전자 속성 테이블, 차량 이력 테이블, 차량 이동 시퀀스 테이블, 차량 빈발 시퀀스 테이블로 구성되어 있다.



Fig. 127 Vehicle attribute table structure and search.

Fig. 127은 차량 속성테이블 구조와 검색 결과 화면, Fig. 128은 운전자 속성 테이블 구조와 검색 결과 화면, Fig. 129는 차량 이력 테이블 검색 결과 화면, Fig. 130은 차량 위치정보에서 방향각 정보를 통해 차량 위치 정보를 추출하여 이를 다시 차량 이동 시퀀스 정보로 변환, 저장한 결과 화면이다.

Fig. 131은 차량 이동 시퀀스 테이블에서 차량 빈발 시퀀스를 저장한 화면으로, 하나의 경로를 추출하여 저장하였으므로 빈발 계수(tf)는 1로 나온 것이다. 따라서 계속적으로 차량이 운행할 경우에는 다양한 빈발 계수가 발생하게 된다.



Fig. 128 Driver attribute table structure and search.

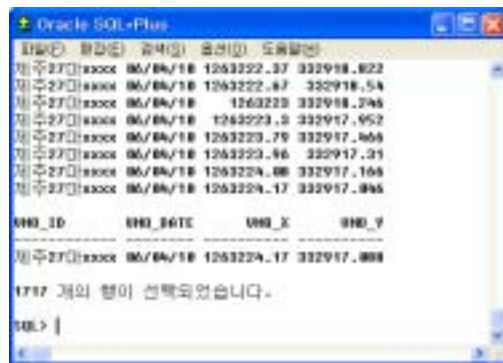


Fig. 129 Vehicle history data search result.

SEQ_NO	S_X	S_Y	E_X	E_Y	LANG	ANG	SPEED
336	1263111.32	332932.13	1263111.62	332932.004	9	116.6	31.4
337	1263111.62	332932.004	1263132.15	332931.398	531	92	57.9
338	1263132.15	332931.398	1263132.72	332931.164	20	116.1	18
339	1263132.72	332931.164	1263142.03	332925.29	301	127	30.1
340	1263142.03	332925.29	1263151.16	332917.478	337	135.6	50.5
341	1263151.16	332917.478	1263151.8	332917.286	18	109.5	32.4
342	1263151.8	332917.286	1263153.01	332917.43	31	81.9	37.7
343	1263153.01	332917.43	1263153.85	332917.898	26	56.4	46.9
344	1263153.85	332917.898	1263158.74	332922.374	187	42.5	67.3
345	1263158.74	332922.374	1263210.3	332930.084	381	51.5	49.1
346	1263210.3	332930.084	1263210.78	332930.312	14	60.5	25.6
347	1263210.78	332930.312	1263211.06	332930.33	7	85.5	25.7
348	1263211.06	332930.33	1263224.08	332917.166	527	140.3	51.3
349	1263224.08	332917.166	1263224.17	332917.046	4	146.2	8
350	1263224.17	332917.046	1263224.17	332917.088	3	353.2	.2

350 개의 행이 선택되었습니다.

SQL>

Fig. 130 Vehicle moving sequence table.

UND_ID	SEQ_NO	TF
제주270XXXX	340	1
제주270XXXX	341	1
제주270XXXX	342	1
제주270XXXX	343	1
제주270XXXX	344	1
제주270XXXX	345	1
제주270XXXX	346	1
제주270XXXX	347	1
제주270XXXX	348	1
제주270XXXX	349	1
제주270XXXX	350	1

350 개의 행이 선택되었습니다.

SQL>

Fig. 131 Vehicle frequent sequence table.

5) 결과 분석

본 연구에서 제안한 하이브리드 기법의 연구 결과는 다음과 같다.

첫 번째, 실시간 차량 운행으로 발생한 차량 위치정보를 제안한 기법의 차량 위치정보 추출 방법을 통해 차량 위치정보를 효율적으로 감소시켜 차량 위치정보 저장 공간을 절약할 수 있음을 보였다.

Table 28과 Fig. 132는 실시간 차량 운행에 따른 차량 위치정보의 파일 크기와 각각의 방향각별 파일 크기 그리고 하이브리드 궤적의 파일 크기를 각 차수별로 비교한 테이블과 그래프이다.

파일크기는 Fig. 132에서 보는 바와 같이 실제 파일크기에서 약 18%~20%로 차량 운행궤적을 표현할 수 있었으며, 저장 공간을 약 80 ~ 82% 정도를 절약 할 수 있음을 보였다. 또한 하이브리드 궤적의 파일 크기와 레코드 수도 실시간 차량 운행궤적의 파일크기보다는 많은 저장 공간을 절약 할 수 있었다. 하지만, 방향각 보다는 저장 공간이 많아지는 것을 볼 수 있다.

레코드 수를 정리한 Table 29와 Fig. 133은 Table 32와 Fig. 132와 거의 동일한 백분율을 보이고 있다.

Table 28. The file size of similar trajectory by real time vehicle running (units : %).

구분	1차	2차	3차	4차	5차	직선도로
방향각 10°	18.84	18.67	20.72	18.55	19.27	5.36
방향각 20°	9.39	8.42	10.28	8.87	10.16	3.65
방향각 30°	4.96	4.91	5.61	5.16	5.6	3.07
하이브리드 10°+1	20.66	19.97	22.08	20.58	21.20	7.08
하이브리드 20°+1	10.99	10.09	12.72	11.41	12.37	5.36

Table 29. The number of similar trajectory record by real time vehicle running (units : %).

구분	1차	2차	3차	4차	5차	직선도로
방향각 10°	18.77	18.55	20.77	17.90	19.22	5.15
방향각 20°	9.30	8.34	10.17	9.01	10.06	3.44
방향각 30°	4.87	4.84	5.51	5.02	5.54	2.86
하이브리드 10°+1	20.59	19.84	22.59	20.30	21.14	6.87
하이브리드 20°+1	11.02	10.00	12.60	11.23	12.30	5.15

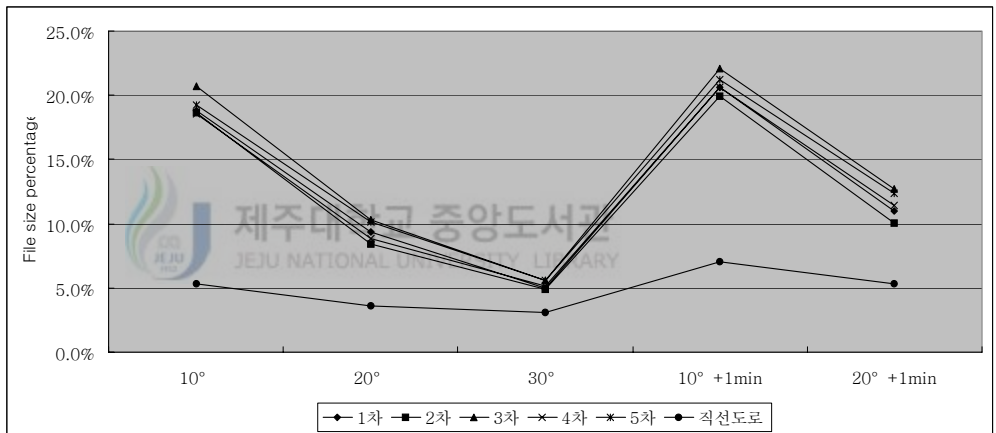


Fig. 132 Comparative graph of similar trajectory file size.

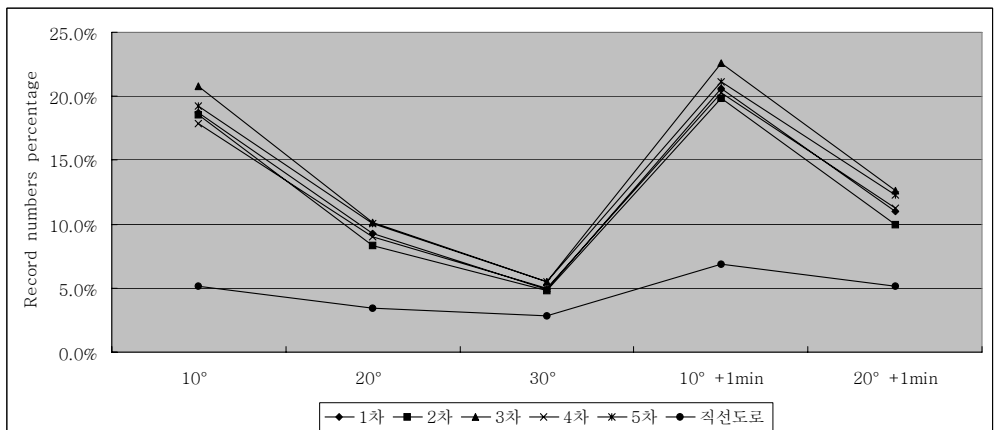


Fig. 133 Comparative graph of the number of similar trajectory record.

두 번째, 실시간 차량 운행궤적과 거의 동일한 차량 운행궤적을 그릴 수 있었다.

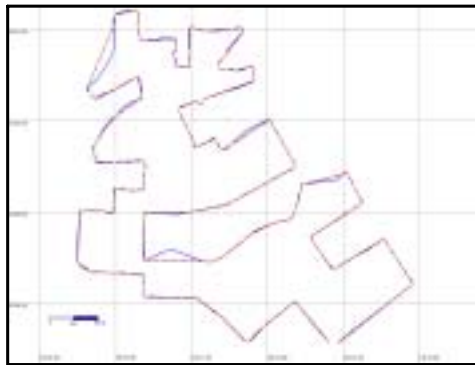
방향각 10°, 20°, 30° 범위로 차량 운행궤적을 표현한 결과 방향각 10°에서 실시간 차량 운행궤적과 거의 동일한 차량 운행궤적을 그렸으며 방향각 20°에서는 다소 차이를 보였지만 시간간격별 차량 운행궤적 보다는 보다 일치된 차량 운행궤적을 그렸다. 그리고 방향각 30°는 실시간 운행궤적과는 많은 차이를 보였다.

하이브리드 유사궤적은 방향각 유사궤적보다 더 실시간 운행궤적에 가까운 운행궤적을 그렸다.

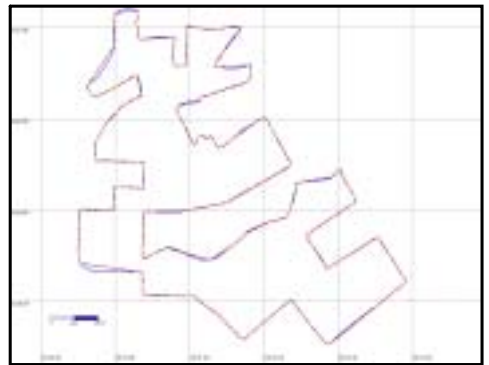
Fig. 134는 실시간 차량 운행궤적과 방향각 유사궤적의 유사성을 보이기 위해서 앞에서 서술한 실시간 차량 운행 궤적과 방향각 유사궤적 비교를 간략하게 도식화한 것이다.

세 번째, 차량 운행 궤적의 일관성을 보였다. 실시간 차량 운행궤적에서 시간 간격별 차량 운행궤적을 추출하여 차량 운행궤적을 그렸을 때 매번 다른 차량 운행궤적을 그렸다. 그러나 방향각 10°와 20°에서는 거의 일정한 운행궤적을 그림을 보였다. 따라서 시간간격의 문제점인 정확한 차량 운행경로를 파악하지 못하는 단점을 방향각 기법에서는 차량 운행 경로를 일관성 있게 실시간 운행 궤적과 거의 동일한 궤적을 그리는 것을 앞에서 보였다. 또한 하이브리드 궤적에서도 방향각보다 더 정확한 운행궤적을 그렸고, 일관성에서도 방향각과 같은 일관성을 보였다.

Fig. 135는 방향각 유사궤적의 일관성을 간략하게 표현한 화면이다.



The first similar trajectory 10°



The second similar trajectory 10°



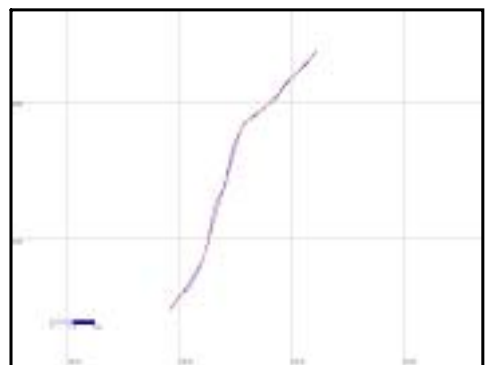
The third similar trajectory 10°



The 4th similar trajectory 10°



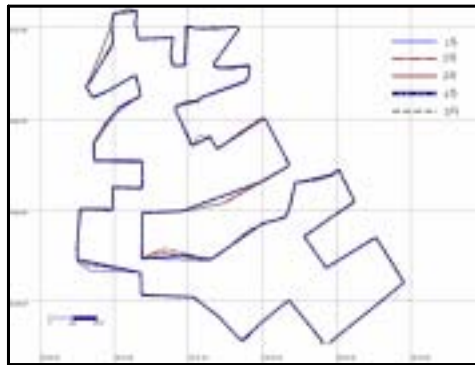
The 5th similar trajectory 10°



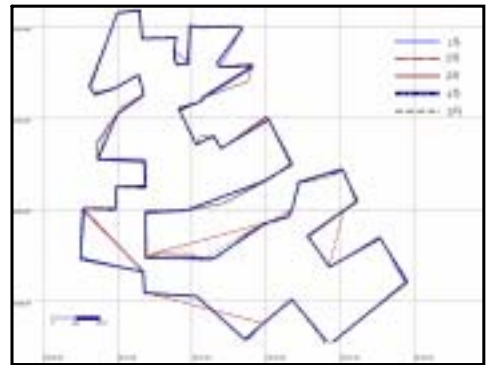
Straightaway similar trajectory 30°

— real time - - - similar trajectory

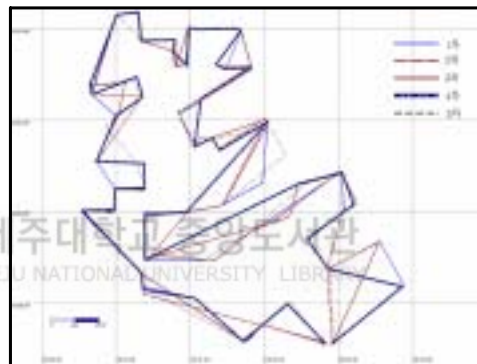
Fig. 134 Similarity between real time and similar trajectory of.



Similar trajectory(10°)



Similar trajectory(20°)



Similar trajectory(30°)

Fig. 135 Consistency of similar trajectory.

VI. 결론

본 연구는 시간 변화에 따라 방대하게 발생하는 차량 위치정보에 대해 방향각 정보를 이용하여 의미 있는 차량 위치정보를 추출할 수 있도록 하였다. 그리고 추출된 차량 위치정보와 시간간격을 이용하여 차량 운행 궤적을 표현할 수 있는 하이브리드 기법을 제안하였다.

제안 기법은 차량 이동객체가 좌회전, 우회전, 직진 등 방향 변환이 이루어지는 시점들에서의 위치 좌표값을 추출하여 데이터베이스에 저장하고 저장된 위치 좌표값을 이용하여 과거의 차량 운행궤적을 표현함으로써, 과거의 차량 위치 추적뿐만 아니라, 차량 이동객체의 운행 패턴을 추출할 수 있도록 하였다. 또한 기존의 시간간격 추출방법을 이용하여 차량 운행 궤적을 표현시 정확성이 떨어지는 문제점을 실시간 운행궤적과 거의 동일한 운행궤적을 표현할 수 있도록 정확성을 향상시켰다.

기존 카 네비게이션(Car-Navigation)에서는 차량 운행궤적을 파일형태 또는 이미지 형태로 저장되었기 때문에 질의를 통한 차량 위치추적과 검색은 불가능하다. 하지만 본 연구에서는 차량 운행궤적에 따른 위치정보를 데이터베이스에 저장·관리할 수 있도록 하여 과거의 차량 위치추적이 나 운행 패턴을 질의를 통하여 추출할 수 있도록 하였다. 또한 차량 운행 방향이 빈번히 변경되는 도심지역의 차량 운행일 경우 시간간격을 길게 설정할 경우 차량 방향이 변경된 자세한 운행 경로를 제대로 표현하지 못하는 경우가 발생한다. 하지만 본 연구에서 제안한 기법은 자세한 운행 경로궤적을 표현할 수 있었다.

기존 차량 추적 시스템 또는 차량관리 시스템들은 실시간으로 차량 위치정보를 수집·관리하였기 때문에 대용량 데이터베이스를 사용하였다.

따라서 초기 시스템 구축비용이 증가하는 문제를 발생시켰고, 또한 차량 객체마다 최근의 위치정보만을 저장·관리하여, 다양한 서비스를 제공하지 못하는 고비용, 저효율의 시스템을 구축하게 되었다.

반면, 제안 기법은 방대한 차량 위치정보에서 의미 있는 차량 위치정보만을 추출하여 저장하기 때문에 많은 저장 공간을 절약할 수 있었다. 그리고 차량 이동객체의 운행 경로 시퀀스를 통하여 다양한 텔레메틱스 서비스 개발을 위한 기본 자료로 활용할 수 있도록 하였다.

본 연구에서는 제안 기법의 효율성 검증을 위해 다양한 도로교통조건 하에서 차량을 운행하고, 그에 따른 차량 위치정보를 실시간으로 추출하였다. 또한 차량 이동객체에 GPS장비를 탑재하여 실시간으로 도로 주행한 차량 운행 위치정보와 제안한 기법으로 추출한 차량 위치정보를 비교 분석할 분석 방법에 대해 제시하였다.

분석 방법은 실시간 차량 운행궤적, 시간간격별 차량 운행궤적(1분, 2분, 3분, 4분, 5분), 방향각 차량 운행궤적(10° , 20° , 30°), 하이브리드 운행궤적($10^\circ+1$ 분, $20^\circ+1$ 분)을 추출하여 시각적 비교, 정량적 비교, 일관성 비교, 차량 운행거리 비교 등 다양한 비교·분석을 통하여 성능을 분석하였다.

다양한 분석방법을 이용하여 실시간 운행궤적과 유사성을 비교하였으며, 또한 효율성을 비교하여 제안 기법의 성능을 검증하였다.

따라서 다음과 같은 결과를 도출하였다.

본 연구에서 제안한 기법으로 차량 운행 위치정보를 추출하였을 경우, GPS로부터 수신된 실시간 차량 위치정보를 그대로 저장하는 것 보다 약 80% 정도 저장 공간을 절약할 수 있었다. 따라서 차량 위치 추적 시스템이나 관리 시스템 구축시 비용 절감 효과를 얻을 수 있었다.

또한, 차량 운행 궤적을 표현함에 있어 시간간격 차량 운행궤적보다 실시간 운행궤적에 가까운 운행궤적을 표현할 수 있었다. 이는 적은 양의

위치정보로도 실시간에 가까운 운행궤적을 표현할 수 있기 때문에 과거의 차량 운행궤적을 추적하거나, 운행 패턴을 추출하고자 할 때 유용하게 사용할 수 있음을 알 수 있다.

차량 운행궤적 일관성 비교에서 시간간격 차량 운행궤적은 도로 여건과 교통상황에 따라 차량 운행궤적이 매번 달라지는 것을 볼 수 있었으며, 방향각 유사궤적 기법의 차량 운행궤적은 시간간격 방법보다 일관성이 있음을 알 수 있다. 이는 차량 이동객체의 속도 변화에 따라 측정되는 위치 정보가 달라지기 때문에 일관성이 없지만, 방향각과 하이브리드 방식을 통한 위치정보는 도로 각이 변경되는 변경 각이 거의 일정하기 때문에 높은 일관성을 보였다. 또한 방향각 유사궤적 기법과 시간간격을 혼합한 하이브리드 기법은 차량 운행궤적을 표현함에 있어 방향각 유사궤적 기법보다 더 실시간 운행궤적에 가까운 운행궤적을 표현하였다. 그리고 일관성 표현에 있어서도 방향각 운행궤적 보다 높은 일관성을 보였고, 차량 운행거리 비교에서도 실시간에 가까운 운행거리를 보였다. 하지만, 정량적 분석인 파일 크기와 레코드 수는 방향각 기법보다 다소 많아지는 것을 보였다. 이는 방향각 정보로 추출한 차량 위치정보와 시간간격에서 추출한 위치정보가 병합된 것으로 방향각 정보보다 다소 많아진다. 그러나 방향각 정보로 추출된 차량 위치정보와 시간간격을 통해 추출된 차량 위치정보가 중복될 경우는 하나의 위치정보만 저장하였기 때문에 비교적 많은 데이터는 발생하지 않았다.

본 연구에서 제안한 하이브리드 기법의 기대 효과 및 활용 방안은 다음과 같다.

첫째, 소량의 차량 위치정보로 실시간으로 운행한 운행궤적과 거의 동일한 차량 운행궤적을 표현할 수 있기 때문에 차량 위치추적 시스템과 관리시스템 구축 시 비용 절감 효과를 기대할 수 있다.

둘째, 과거의 차량 위치정보를 데이터베이스에 저장·관리하기 때문에 질의를 통한 차량 위치추적과 차량 운행패턴을 추출할 수 있다. 따라서 차량 물류 수송 시스템에서 차량 배차와 수송 최적 경로 탐색 시 유용한 자료로 사용할 수 있다.

셋째, 과거의 차량 위치정보를 운행계적 시퀀스 빈발패턴 계수로 데이터베이스에 저장·관리함으로써, 시간에 따른 차량 이동객체의 차량 운행 패턴이나 그 밖의 유용한 정보들을 추출하여, LBS기반의 차량 관련 데이터마이닝 응용시스템에 활용할 수 있을 거라 사료된다.

넷째, 차량 위치정보와 운행 계적을 저장·관리함으로써, 차후 차량 보험료 산정시 개인별 맞춤형 보험료를 산정할 수 있을 거라 사료된다. 즉, 사고 다발지역을 자주 운행하는 차량, 또는 교통량이 많은 도심지역을 운행하는 차량에게는 높은 보험료를 산정하고 그에 반해 사고 다발지역이 아닌 곳과 도심 외곽지역을 운행하는 차량에게는 낮은 보험료를 산정하여 개인별 맞춤형 보험료를 산정할 수 있는 기준 자료로 사용될 수 있다.

다섯째, 데이터베이스에 과거 차량 운행에 따른 위치정보가 저장되어있기 때문에 범죄를 일으킨 차량을 검색할 때에도 유용하게 사용될 수 있다. 즉, A지점에 뺑소니 교통사고가 일어났다면 사고가 일어난 시점과 위치정보를 이용하여 A지점을 운행한 차량들을 검색하여 뺑소니 차량 여부를 검색할 수 있다.

본 연구에서 제안 기법의 개선되어야 할 문제점 및 향후 연구는 다음과 같다.

첫째, GPS수신 오류에 의해 무의미한 차량 위치정보가 수신되었을 경우 이를 인위적으로 처리하는 것이 아니라 자동으로 인지하여 제거, 처리할 수 있는 방법이 모색되어야 한다. 예를 들면, GPS 수신 불량지역에서는 무의미한 위치정보가 수신되었을 경우, 이를 자동으로 인지하여 제거

할 수 있는 알고리즘과 패턴 연구가 이루어 져야 할 것이다.

둘째, 본 연구에서는 차량의 방향각 변화와 거리 요소에 의해 차량 위치정보를 추출하였다. 이를 더 보완하여 차량 속도변화, 도로여건, 교통상황 등 추가 요소들을 입력받아 가변적으로 차량 위치정보를 추출할 수 있는 다양한 기법들이 연구가 되어야 할 것이다.

셋째, 차량 운행궤적을 비교하기 위해 차량 운행궤적 유사성 측정 방법들이 연구되어야 한다. 본 연구에서는 시각적으로 GPS TrackMaker와 차영상 기법 그리고 차량 운행 거리를 통하여 유사성 측정 방법을 선 보였지만, 다양한 방법을 통하여 실제 차량 운행궤적과 제안 기법에 의해 추출된 유사궤적과의 유사성을 시각적인 분석뿐만 아니라, 정량적인 분석으로 유사성 측정할 수 있는 다양한 방법들에 관한 연구가 이루어져야 할 것이다.

넷째, 차량 위치정보는 차량 이동속도, 도로여건, 교통상황에 따라 GPS에서 수신된 위치정보가 다르다. 즉, 동일한 운행 경로를 운행하더라도 차량 위치정보는 전혀 다른 위치정보들을 갖게 된다. 따라서 서로 다른 위치정보를 갖으면서 동일한 운행경로를 갖는 차량 운행궤적을 중복적으로 저장하는 것이 아니라 이를 판단하고 제거·관리할 수 있는 방법들이 연구되어야 할 것이다.

다섯째, GPS 위치검출 방법은 수신율이 좋은 상태와 그렇지 못한 상태에 따라 위치검출 오차가 커지고 작아진다. 따라서 GPS보다 위치 측위율이 높은 추측항법이나 GPS와 추측항법이 결합한 형태인 하이브리드 검출 방법을 통한 유사궤적 표현기법이 연구되어야 한다.

여섯째, 지역적인 특성을 가미한 가변적 차량 위치정보 추출기법이 연구되어야 할 것이다. 즉, 차량 소통이 원활하고 빠르게 움직이는 지역과 직선도로일 경우에는 시간간격과 방향각 각을 크게 하고, 차량 소통이 원

활하지 못하고 차량 속도가 저하되는 지역과 잦은 방향 변환이 이루어지는 도심지역일 경우에는 시간간격과 방향각 각도를 작게 하여 최적의 운행 궤적을 표현할 수 있는 기법들이 연구가 되어야 할 것이다.

마지막으로 과거의 차량 위치정보를 관리함으로써, 과거와 이를 활용한 미래의 차량 위치 예측에 있어서 기존 시스템보다는 효과적일 수 있다. 그러나 과거의 차량 위치정보 누적에 따라 질의 응답시 지연이 발생할 수 있다. 따라서 효율적인 차량 위치정보 검색을 위해 색인기법 연구가 다양하게 이루어져야 할 것이다.



참고 문헌

- "A Framework for Generating Network-Based Moving Objects", *GeoInformatica* 6, 2002.
- 안병익, "LBS기술동향과 전망 LBS 구조 및 구성", *한국지리정보*, 10월 호, pp.52-56, 2001.
- Brinkhoff T, "Generating Traffic Data", *Bulletin of the Technical Committee on Data Engineering*, IEEE Computer Society, vol. 26, no. 2, 2003.
- Dead Reckoning, "<http://myhome.shinbiro.com/~zippy/MapMatching.htm>"
- E. Mesrobian, R. R. Muntz, J. R. Santos, E. C. Shek, C. R. Mechoso, J. D. Farrara and P. Stolorz, "Extracting Spatio-Temporal Patterns from Geoscience Datasets," *IEEE Work-shop on Visualization and Machine Vision*, Seattle, WA, June, 1994.
- G. Berger and A. Tuzhilin, "Discovering unexpected patterns in temporal data using temporal logic," *Temporal Databases Research and Practice*, Springer-Verlag, 1998.
- GPS TrackMaker, "<http://www.gpstm.com>".
- G.Siouris, *Aerospace Avionics System - A Modern Synthesis*, Academic Press, 1993.
- H. Mannila and H. Toivonen, "Discovering generalized episodes using minimal occurrences," In *Proc. of the Int'l Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp.146-151, 1996.
- 한경복, 곽호영, "효율적인 차량 이력 데이터 저장을 위한 유사 궤적 저장 기법", *한국콘텐츠학회논문지*, 제6권 제1호, pp.114-125, 2006.1.
- I. Tsoukatos and D. Gunopulos, "Efficient Mining of Spatiotemporal Patterns," In *Proc. of the 7th Int. Symp. on Spatial and Temporal Databases (SSTD)*, 2001.
- Jean-Marc Saglio, Jos'e Moreira: Oporto: "A Realistic Scenario Generator for Moving Objects". *GeoInformatica* 5, pp. 71-93, 2001.

- Jeong J. D., Paek O. H., Lee J. W. and Ryu K.H., "Temporal Pattern Mining of Moving Object for Location-Based Service," In Proc. of International Conference on Database and Expert Systems Applications (Dexa2002), (LNCS2453), 2002.
- J. F. Roddick and M. Spiliopoulou, "Temporal data mining : survey and issues," Research Report ACRC-99-007, University of South Australia, 1999.
- J. F. Roddick and B. G. Lees, "Paradigms for Spatial and Spatio-Temporal Data Mining," Geographic Data Mining and Knowledge Discovery. Taylor and Francis. Research Monographs in Geographic Information Systems. Miller, H. and Han, J., Eds, 2001.
- 정홍진, 정영진, 류근호, "사용자 지정 시나리오에 기반 한 차량 위치 데이터 생성기의 설계", 한국정보과학회 2004 추계학술대회, vol. 31, no. 02, pp. 142-144, 2004.10.
- K. Koperski and J. Han, "Discovering of Spatial Association Rules in Geographic Information Databases," In Proc. of the 4th International Symposium on Large Spatial Databases, 1995.
- K. Koperski, J. Han and J. Adhikary, "Mining knowledge in geographical data," to appear in Communications of the ACM, 1998.
- 김동호, 이혜진, 이현아, 김진석, "e-Logistics 환경에서 차량관제를 위한 이동체 관리 시스템 개발", 정보처리학회논문지 D, 제11-D권 제6호, 2004.10.
- 김보련, 이상현, 이기준, "도로 네트워크 기반 이동객체의 궤적 데이터 생성", 한국정보처리학회 추계학술발표대회 논문집, 제12권 제2호, 2005.11.
- 김진석, 김동호, 류근호, "차량 궤적 추적을 위한 불확실성 처리기 구현", 정보처리학회논문지 D, 제11-D권 제5호, 2004.10.
- 김진원, 지규인, 이장규, 이영재 "GPS와 Dead-Reckoning을 이용한 항법 시스템 설계", 제어·자동화·시스템공학 논문지, 제2권 제3호, pp. 188-193, 1996.9.
- 이금우, "위치기반 서비스를 위한 개인화된 추천 시스템", 이학 석사학위 논문, 충북대학교, 2002.
- 이준욱, 이용준, 류근호, "시간데이터마이닝 프레임워크", 정보처리학회 논문지D, 제9-D권, 제3호, 2002.

- 이준욱, 백옥현, 류근호, “위치기반 서비스를 위한 이동객체시간 패턴 탐사 기법”, 정보과학회논문지, 제29권 제5호, 2002.
- 이준욱, 남광우, “이동객체 위치 일반화를 이용한 시공간 이동 패턴 탐사”, 정보처리학회논문지 D, 제10-D권 제7호, 2003.12.
- M. Erwig and M. Schneider, “Spatio-Temporal Predicates,” Technical Report 262, Fern University, 1999.
- M. N. Garofalakis, R. Rastogi, Shim K. S., “SPIRIT : Sequential Pattern Mining with Regular Expression Constraints,” In Proc. of the international conference on VLDB, 1999.
- R. Agrawal and R. Srikant, “Mining sequential patterns,” In Proc. of the 11th International Conference on Data Engineering, 1995.
- R. E. Valdes-Perez, “Systematic Detection of Subtle Spatio-Temporal Patterns in Time-Lapse Imaging. I. Mitosis,” Bioimaging. Vol.4 , No.4, pp.232-242, 1998.
- R. Snodgrass, “The Temporal Query Language TQuel,” ACM TODS, Vol.12, No.2, June, 1987.
- R. Srikant and R. Agrawal, “Mining sequential patterns : generalizations and performance improvements,” In Proc. of International Conference on Extending Database Technology, Avignon, France, Springer-Verlag, 1996.
- 신기수, 안윤애, 배종철, 정영진, 류근호, “GIS를 이용한 시공간 이동객체 관리 시스템”, 한국정보처리학회 논문지, 제8-D권 제2호, 2001.4.
- 심춘보, 장재우, “이동객체의 유사 부분궤적 검색을 위한 시그니처-기반 색인 기법”, 한국정보처리학회논문지 D, vol. 11, no. 02, pp. 247-258, 2004 . 04.
- Theodoridis, Y. and Nascimento, M.A., “Generating Spatiotemporal Datasets on the WWW”, SIGMOD Record, 29(3):39-43, September 2000.
- T. Abraham, “Knowledge Discovery in Spatio-Temporal Databases”, School of Computer and Information Science, University of South Australia, Ph. D dissertation, 1999.
- T. Bittner, “Rough sets in spatio-temporal data mining,” In Proc. of the 1st International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining(TSDM2000), 2000.
- X. Chen and I. Petrounias, “A framework for temporal data mining,” In

- Proc. of the 9th International Conference on Database and Expert Systems Applications, 1998.
- X. Lu, J. Han and B. C. Ooi, "Discovery of General Knowledge in Large Spatial Databases," In Proc. of Far East Workshop on Geographic Information Systems(FEGIS'93), pp.275-289, 1993.
- Y. Cai, N. Cercone and J. Han, Attribute-Oriented Induction in Relational Databases, in G. Piatetsky-Shapiro and W. J. Frawley (eds.), Knowledge Discovery in Databases, AAAI/MIT Press, pp.213-228, 1991.
- 양해정, 김태완, 이기준, "속도 모델을 이용한 차량 궤적의 효율적인 표현 방법", 한국정보처리학회 추계학술발표대회 논문집, 제11권 제2호, 2004.11
- "지능형교통시스템(ITS) 표준화 로드맵 연구", 한국전산원, 1998.12.



국 문 초 록

하이브리드 기법을 이용한 LBS기반의 유사궤적 추적 시스템

한 경 복

컴퓨터공학과

제주대학교 대학원

본 연구에서는 소량의 차량 위치정보를 통해 과거의 차량 위치를 추적하고, 운행궤적을 표현할 수 있는 방향각 정보를 이용한 하이브리드 기법을 제안하였다.

제안한 기법의 효율성을 검증하기 위해 다양한 조건 하에서 실제 차량 이동객체를 운행하여 차량 위치정보를 수집하였으며, 수집된 차량 위치정보에서 기존 시간간격별 차량 위치정보와 방향각 정보를 이용한 차량 위치정보를 추출하여 비교 분석하였다.

제안 기법으로 생성된 유사궤적을 GPS TrackMaker와 차영상 기법을 통해 실시간 운행 궤적과 비교 분석한 결과 시간간격별 차량 운행궤적보다 실시간 운행궤적과 거의 동일한 운행 궤적을 표현할 수 있었다. 또한 운행 궤적의 일관성을 비교 분석한바 시간간격별보다 일관성 있게 운행궤적을 표현하였다.

실시간 운행으로 생성된 파일 크기와 제안한 기법으로 생성된 파일 크기를 비교 분석한 결과, 약 80%정도 저장 공간을 절약할 수 있음을 정량적으로 검증하였으며. 또한 데이터베이스에 저장된 레코드 수도 같은 결과를 보였다.

제안 기법은 방향각 기법보다 위치정보를 더 발생시켰지만, 차량 운행 궤적을 표현함에 있어 방향각보다 더 실시간에 가까운 운행궤적을 보였다.

끝으로 차량 운행 거리를 비교 분석한 결과, 제안한 기법의 운행거리가 실시간 운행 거리와 거의 동일한 거리를 표현할 수 있음을 보여 그 효율성을 검증하였다.



감사의 글

사람의 일평생은 짧습니다. 그러나 짧다고 해서 어떤 일을 할 수 없는 것은 아닙니다. 자신이 마음먹기에 따라 수천수만 가지의 일도 할 수가 있는 것입니다. 매순간은 우리에게 일을 할 수 있는 시간이며, 결과를 확인할 수 있는 시간이고, 과정을 점검해야 하는 시간인 것입니다.

바로 지금 박사학위 과정을 무사히 마쳤다는 작은 성과에 대해 점검의 시간을 갖고자 합니다. 학위과정 시작과 끝에서 부족한 저에게 크고, 작은 도움을 주신 분들이 너무 많습니다. 다 찾아뵙고 감사의 인사를 드리기 전에 지면을 통해 감사의 인사를 드리고자 합니다.

처음 연구실 문을 열고 들어오면서부터 지금까지 많이 부족한 저에게 세 번째 사각모를 쓸 수 있도록 해주시고, 저를 알고 있는 분들께 논문을 내놓을 수 있도록 지도와 격려를 아낌없이 해주신 지도교수님이신 곽호영 교수님께 감사를 드립니다.

논문 심사와 지도를 위해 열정을 아끼지 않으신 김장형 교수님, 이상준 교수님, 이정훈 교수님께도 감사의 말씀을 올립니다. 그리고 바쁘신 일정에서도 먼 길을 마다 않으시고 내려오셔서 꼼꼼하게 지적해주신 원유현 교수님께도 이 지면을 통하여 감사의 말씀을 올립니다. 또한 안기중 교수님, 변상용 교수님, 송왕철 교수님, 변영철 교수님 그리고 김도현 교수님께도 보다 학술적인 접근을 바라는 말씀과 충고를 아끼지 않아 주셨음에 감사드립니다.

이 논문을 완성하기까지 주위의 모든 분들로부터 수많은 도움을 받았습니다. 같은 연구실에서 함께 공부하고 생활한 동료, 옆에서 논문의 타당성과 효율성에 대해 조언을 해준 김정희 박사, 수업과 연구에 바쁘면서도 논문 수정과 실험을 자기일 마냥 도와준 훈이, 그리고 연구실 홍일점인 혜선, 모두에게 감사의 말을 전합니다. 또한 성철 형님과 두화 형님, 그리고 형수, 명미, 관석이 에게도 좋은 결과가 있기를 이 지면을 통하여 전합니다. 그리고 제주도 IT발전을 위해 열심히 교육과 코딩하고 있는 용재, 혁준, 대환, 인석, 석건에게도 고마움을 전합니다. 또

한 서울에서 열심히 일하고 있는 행진 형과 동현, 병휘에게도 고마움을 전합니다.

연구실은 다르지만 영도, 강석 선배님, 그리고 변태보 선생님, 김정효 선생님 그리고 제출할 서류를 꼼꼼히 챙겨주었던 은경, 그리고 정아 선생님을 비롯한 학과 모든 연구실 식구들, 또한 좋은 벗들인 종훈, 기현, 민석, 준호에게도 감사의 말을 전합니다.

제주대학교 학과 조교 생활부터 지금까지도 계속해서 따뜻한 충고와 격려를 아낌없이 해주셨던 해양토목공학과와 양성기 교수님, 김남형 교수님, 남정만 교수님, 이병걸 교수님, 박상렬 교수님께 고마움을 전합니다. 또한 한라대학 컴퓨터 정보계열 교수님들, 컴퓨터정보활용과 교수님들, 행정정보과 교수님들, 관광대학 카지노 경영학과 교수님께도 그동안의 배려와 격려에 고마움을 전합니다.

와인은 깊은 숙성과정에서 좋은 와인이 탄생하듯이 사람 또한 사귄 수 록 깊은 믿음이 생기는 제주 와인향기 동호회의 휘노비노님, 오라방님, 삼매님, 등대님, 리텔님, 고경님, 오리님, 싸또님, 땡깡 공주님, 아시님, 영례님, 데이빗님, 별이님, 취생몽사님, 취한자님, 샤브님 등 그 외 와인향기 회원님들에게도 고마움을 전합니다. 또한 씨네아일랜드의 현충렬 대표이사님, 김정훈 이사님, 오주연 사무국장님, 이영운 사무차장님 그리고 형미, 인정, 진인에게도 고마움을 전합니다.

긴 시간 동안 동생을 위해 깊은 사랑으로 뒷바라지 해주신 형님과 형수님, 날 위해서 집안일과 잡다한 일에 신경 쓰지 않게 집안일을 잘 이끌어준 경애와 큰 매제 그리고 경욱과 작은 매제, 멀리서 묵묵히 바라보면서 지원을 아끼지 않은 동생 경찬과 제수씨 그리고 막내 경민, 사랑스런 나의 조카들에게도 고마움을 전합니다.

끝으로 오늘의 제가 있을 수 있도록 사랑으로 키워 주신 어머니, 공부한다, 시간 없다, 이런 저런 핑계로 자주 찾아뵙지 못하고 늘 걱정만 끼친 어머니께 이 작은 결실이 조금이나마 저의 마음을 대신할 순 없겠지만 감사의 마음으로 어머니께 바치고 그리고 저 먼 하늘나라에서 내려다보실 아버지께 이 작은 논문을 바칩니다.

2006년 6월 연구실에서...