

# 가중치 벡터합을 이용한 이동객체의 방향계산 및 미디어 검색방법

서 창 덕<sup>†</sup> · 한 기 태<sup>\*\*</sup>

## 요 약

본 논문은 기존 위치기반 서비스에서 최근접질의 및 한 지점에서의 방향성분을 고려한 최근접질의 단점을 해소하고자 가중치 벡터합을 이용하는 새로운 검색방법을 제안한다. 검색반경으로 1차 필터링된 영역에서, 2차 필터링을 위해 이용자의 이동방향, 관심방향 및 검색각도를 조합한 방향정보를 이용한다. 이동방향은 일정구간내 존재하는 벡터들의 가중치 합으로 계산하며, 검색각도를 0~360°까지 세분화하여 검색방향에 대한 범위를 조절 하도록 한다. 본 검색방법에 사용되는 데이터는 촬영위치가 기록된 정지영상 및 동영상, 업체나 관광지의 위치정보와 함께 소비자에게 제공되는 텍스트, 웹, 영상 등 각종 미디어 형태의 데이터가 될 수 있다. 제안하는 방법은 이동 중인 이용자가 현 위치를 기준으로 일정 반경 내에 있으면서 유사방향에 부합하는 미디어만을 검색하도록 함으로써, 이미 지났거나 혹은 관련 없는 방향의 미디어를 배제한 검색결과를 제공하기 때문에 기존의 위치만을 고려한 검색방법에 비해 보다 정확한 검색을 보장할 수 있으며, 방향성을 고려한 기존 최근접질의에 비해서도 보다 유연하고 포괄적인 검색결과를 보장한다.

키워드 : LBS, NN, GPS, TM좌표, 방위각, 이동방향, 검색방향, 방향계산, 가중치벡터합

## A Direction Computation and Media Retrieval Method of Moving Object using Weighted Vector Sum

Changduk Suh<sup>†</sup> · Gi-Tae Han<sup>\*\*</sup>

### ABSTRACT

This paper suggests a new retrieval method using weighted vector sum to resolve a problem of traditional location-based retrieval method, nearest neighbor (NN) query, and NN query using direction. The proposed method filters out data with the radius, and then the remained retrieval area is filtered by a direction information compounded of a user's moving direction, a pre-fixed interesting direction, and a pre-fixed retrieval angle. The moving direction is computed from a vector or a weighted vector sum of several vectors using a weight to adopt several cases. The retrieval angle can be set from traditional 360° to any degree you want. The retrieval data for this method can be a still and moving image recorded shooting location, and also several type of media like text, web, picture offering to customer with location of company or resort. The suggested method guarantees more accurate retrieval than traditional location-based retrieval methods because that the method selects data within the radius and then removes data of useless areas like passed areas or an area of different direction. Moreover, this method is more flexible and includes the direction based NN.

Key Words : LBS, NN, GPS, TM Coordinates, Azimuth, Moving Direction, Retrieval Direction, Direction Computation, Weighted Vector Sum

### 1. 서 론

GPS(Global Positioning System) 보급의 확대로 많은 저장 매체들이 위치정보와 함께 저장되었다가 인터넷 및 단말기 상의 전자지도와 결합하여 위치기반 서비스(LBS: Location-

based Service)로 제공되고 있다[1-3]. 이때 디지털카메라로 촬영한 사진을 전자지도 상에 매핑시켜 어디에서 찍은 사진 인지를 시각적으로 알려주는 서비스도 등장하였는데 특히 GPS를 이용하면 Geo-tagging이라는 기술을 이용하여 보다 편리하게 촬영위치를 자동 등록할 수 있다[4-6]. 근래에 PDA(Personal Digital Assistant), PMP(Portable Multimedia Player), Navigation 등의 단말기를 차량에 장착하는 추세이며, DMB(Digital Multimedia Broadcasting), TPEG(Transport Protocol Experts Group), WiBro(Wireless Broadband), HSDPA(High Speed Down-link Packet Access)처럼 이동

<sup>†</sup> 정 회 원: Foto Fiddlers(NZ), IT Manager

<sup>\*\*</sup> 정 회 원: 경원대학교 IT대학 소프트웨어학부 컴퓨터미디어전공 교수  
(교신저자)

논문접수: 2007년 9월 3일

수정일: 1차 2008년 2월 12일, 2차 2008년 3월 24일

심사완료: 2008년 4월 1일

중 방송, 교통정보, 인터넷 등과 같은 서비스 이용이 가능해졌다. 이러한 시점에서 사용자는 이동 중에도 현재의 위치를 기반으로 하는 각종 지능적인 서비스를 제공받거나 원하는 정보를 검색할 수 있게 되었다.

이제는 개인도 컴퓨터 앞에서 정보를 소비하는 입장에서 벗어나 카메라나 캠코더로 어디에서나 다양한 미디어를 직접 제작하여 제공하는 UCC(User-Created Content)의 생산자가 될 수 있다. 여기에, LBS가 일반화되고 여러 위치기반 서비스 콘텐츠들이 축적되면 텍스트, 문서, 정지영상, 동영상, 웹 등 여러 형태의 미디어로 구성된 데이터들이 혼재하게 되며, 사용자들은 이들 미디어 데이터 중에서 자신이 필요로 하는 데이터만을 검색할 필요가 생기게 된다. 즉, 특정 기준점으로부터 일정 거리 반경 이내에서 촬영한 사진을 검색한다거나 특정 위치나 매장에서 일정 반경 이내의 다수 고객에게만 광고하는 선별적 위치기반 push 서비스, 도로 운전자들에게 전방 수Km에서 일어난 도로교통사고로 인한 혼잡화면사진을 전송해 주거나 우회할 수 있도록 하는 서비스, 주변 관광지, 식당, 특산물 등의 사진을 주변을 지나는 운전자에게 홍보하는 서비스 등을 예로 들 수 있다. 이는 다가오는 유비쿼터스(Ubiquitous) 시대에 u-City 환경을 구축하기 위한 앞서의 각종 서비스 개발에 필요한 위치기반 기초 검색 기술로서 LBS에 필수적인 요소기술 중 하나가 될 것으로 기대하며 보다 지능적이고 정확한 검색방법이 요구된다.

## 2. 기존 위치기반 검색 서비스

### 2.1 LBS에서 기존 검색 방법

LBS는 그 서비스 대상의 미디어가 어떠한 형태이든지 간에 미디어의 촬영위치나 미디어내 객체의 위치를 위경도와 같은 정보를 이용하여 각종 서비스와 접목한 것을 말한다. 업체에서 제공하는 서비스 이외에 개인이 직접 위치와 함께 제공할 수 있는 가장 일반적인 형태로는 정지영상이 있다. 정지영상을 위한 디지털카메라나 카메라폰은 이미 대중화되었으며, 일부 모델이기는 하지만 GPS가 내장된 카메라나 외장형 GPS 기록장치로 말미암아 보다 더 쉽고 빠르게 위치가 기록된 영상을 제공할 수 있게 되었다.

디지털카메라로 촬영한 정지영상 파일에는 EXIF(EXchangeable Image File format)라는 표준 메타정보 기록 태그[7]가 마련되어 있다. 여기에는 촬영과 관계된 모든 정보(카메라 모델명, 촬영시각, 셔터속도, ISO설정값, 조리개값, 촬영모드 등 각종 카메라 설정값)가 영상파일규격인 JPEG이나 TIFF와 같은 포맷에 맞추어 기록된다. 또한 GPS의 위경도 좌표도 기록할 수 있도록 되어 있다. 이처럼 디지털카메라로 촬영된 영상과 관련된 많은 서비스 중에서 사진을 지도와 결합해 서비스 하는 위치기반 영상서비스가 점차 증가하고 있다.

위치정보가 기록된 각종 미디어가 제공되면 사용자는 자신이 원하는 조건에 맞는 데이터만을 검색할 수 있고, 이때 미디어의 위치정보뿐만 아니라 자신의 위치까지 고려해 검

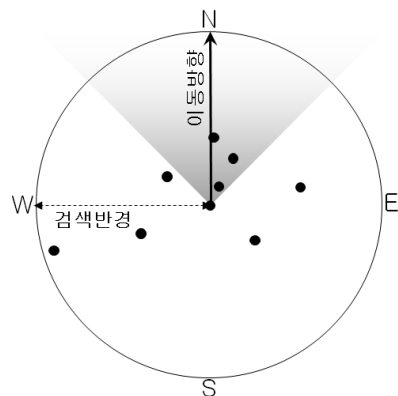
색하면 보다 신속하고 정확한 검색결과를 얻을 수 있게 된다. 즉, 자신이 위치한 곳을 중심으로 그 곳이 집이든 사무실이든 혹은 달리는 차 안이든 현재 위치를 기준으로 반경 몇m 이내에 위치한 곳의 정보만을 선별적으로 검색할 수 있는 최근접질의(NN: Nearest Neighbor)[8]에 의한 검색이 가능하다. 예를 들어 네비게이션에서 현재 위치에서 가장 가까운 곳의 주유소나 맛집 등을 검색해주는 경우가 한 예이다.

### 2.2 위치기반 검색의 한계

위치기반 검색에서의 근접질의에 의한 경우가 (그림 1)에 나타나 있다. 이 경우 9개의 데이터가 검색반경 내에 들어 오게 된다. 기존의 이같은 방법은 이용자의 현재위치가 고정된 상황에 적합한 방법이지만 이용자가 각종 운송수단으로 이동하는 경우에는 9개의 데이터 모두를 원하지 않을 수도 있다. 즉, 이용자가 북쪽(N) 방향으로 이동 중이라고 할 때 통상적으로 이동방향에 위치한 4개의 영상을 검색하고 싶어 하며 이미 지나온 곳의 영상은 관심 밖이다. 혹은 도심지의 경우라면 이동방향과 달리 우측 45°가 검색방향일 수도 있다. 이처럼 현 위치를 기준으로 360° 전범위가 아니라 특정 방향을 기준으로 검색하고 싶어하는 경우가 많은데 기존방법처럼 360° 전범위를 검색하게 되면 원하지 않는 데이터까지 검색되는 문제점이 있게 된다.

이러한 기존 NN방법의 문제점을 해소하기 위해 방향정보를 고려한 질의방법들[9,10]이 제안되었으며, 차량과 같은 이동객체의 방향을 양자화한 기법[9]도 제안되었다. 이는 동일방향으로 주행중 차선변경시 발생하는 방향차를 극복하기 위해 360°를 일정각도(예: 15°)로 나누어 그 각도 범위 내에서는 동일한 가중치를 갖도록 한 것이다.

이동객체와 검색대상 객체와의 거리뿐만 아니라 두 객체의 방향차까지 고려하는 DNN에서 더 나아가 차선변경에 효과적으로 대처하기 위한 양자화 기법을 도입한 QDNN이 그러하듯, 실세계에서는 매우 다양한 경우가 존재한다. 방향이 90°로 바뀌는 교차로가 많은 시내 도심지를 이동하는 경우도 있고 완만한 곡선으로 방향을 바꾸어가며 주행하는 국도나 일정구간 직선형태인 고속도로를 주행하는 경우도 있다. 사용자의 요구 또한 다양해서 방향전환에 민감하게 반



(그림 1) 위치기반의 검색대상(9개)

응하는 검색결과를 원할 수도 있지만 목표점을 기준방향으로 하여 현재의 방향에 너무 민감하게 반응하지 않는 검색결과를 원할 수도 있다.

기존 논문이 이러한 다양한 경우에 모두 대처하기 곤란한데 비해, 본 논문에서는 이를 해결하기 위한 방안으로 먼저 사용자의 관심방향과 검색각도를 정의하고, 가중치 벡터합에 의한 이동방향의 계산방법과 질의시점을 위한 참조점 설정 방법을 제안한다.

### 3. 가중치 벡터합을 이용한 미디어 검색

방향은 일반적으로 단독으로 사용되기 보다는 위치와 함께 사용되는데 위치와 방향은 데이터 및 이용자에 적용될 수 있다. 이들과의 적용여부에 따른 관계가 <표 1>에 나타나 있다. 서비스 데이터의 위치만을 제공했던 초기의 방법1에서 현재는 서비스 이용자의 현 위치까지 고려해 인접 관심지점(주유소, 맛집 등)을 검색해주는 방법2의 서비스가 제공되고 있다.

최근들어 GPS의 위경도를 포함한 미디어가 점차 확산되고 있지만 방향까지 포함된 미디어는 전무한 실정이다. 방향을 기록하려면 디지털나침반이 필요하며 GPS는 물론 디지털나침반까지 내장된 촬영기기가 등장 한다면 일반인들도 손쉽게 위치 및 방향이 자동 기록된 미디어를 생산할 수 있을 것이다. 그러나 현재는 GPS가 내장되어 촬영시 EXIF에 자동기록되는 카메라조차 극히 일부 모델에만 존재한다.

<표 1>에서 보듯이 방향정보는 데이터의 방향정보와 이용자의 방향정보로 구분되지만, 방향정보가 기록된 데이터가 거의 없는 실정으므로, 본 논문에서는 데이터의 방향정보 없이 이용자의 방향정보만으로도 검색에 활용할 수 있는 방법3을 대상으로 한다.

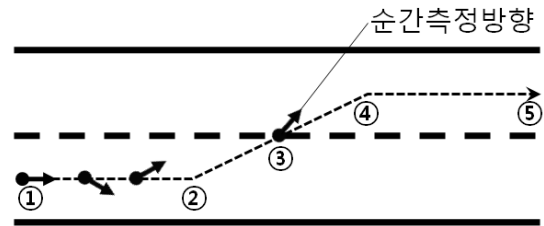
#### 3.1 이동방향, 관심방향 및 검색각도

본 논문에서 사용하는 방향정보란 360° 중 일정 범위각도만을 검색대상에 포함시키기 위해 사용하는 것으로 북쪽을 기준으로 우측으로 회전하면서 값이 증가하는 방위각을 사용한다. 이 검색영역의 방위각을 결정하기 위해서는 이동방향과 관심방향 그리고 좌우 검색각도를 조합하여 산출한다.

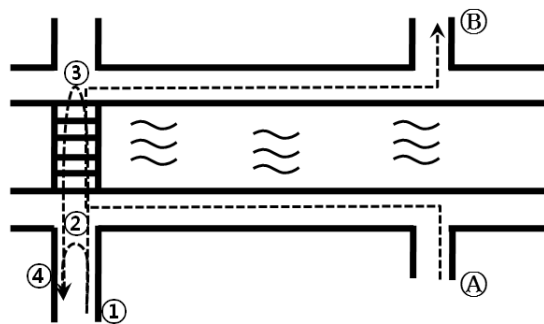
이용자의 이동방향이란 LBS기반 검색 단말기를 휴대하거나 차량에 장착하여 이동할 때의 진행방향을 말하는 것으로 이러한 이동방향을 취득하는 방법은 다음처럼 4가지 방법 중의 하나로 얻을 수 있다.

1. 수동으로 직접 입력
  2. 디지털나침반으로 얻음
  3. 지나온 궤적으로 계산함
  4. 지나갈 경로로 계산함

첫 번째 방법은 방위각 측정을 위한 디지털 장비가 없어도 되지만 매번 직접 입력해야 하는 불편함이 따른다. 나머지 방법들은 이용자가 이동하면서 일정간격으로 이동방향이 파악되고 검색이 자동으로 이루어질 수 있어 편리하다. 특히 두 번째 방법은 이동방향을 실시간으로 손쉽게 얻을 수 있는 장점이 있지만, 차선변경 같은 순간의 방향변화에 검색방향이 영향을 받아 원치 않는 방향에서 검색이 되는 문제가 있다. 이를 위해 QDNN이 제안되었지만 양자화된 각도 이내에서 분기하거나 전체적인 직선코스 내에서 일정각도 이상으로 방향을 번갈아 바꾸는 경우엔 적합하지 않을 수도 있다. (그림 2)에 실제 주행방향과 측정방향과의 차이를 보이고 있다. 디지털나침반의 특성상 이동객체에 의해서 뿐만 아니라 주변지형지물에 영향을 받아 급격히 변하는 경우가 존재하므로 직선으로 주행 중이라 할지라도 항상 일정 방향을 유지한다고 보기 어렵다. 세 번째 방법은 GPS를 사용해 과거의 경로를 바탕으로 앞으로의 이동방향을 예측하는



(그림 2) 실제주행방향과 측정방향의 차이



(그림 3) 직선거리와 실제거리의 차이

<표 1> 데이터와 이용자 간의 위치 및 방향 적용여부

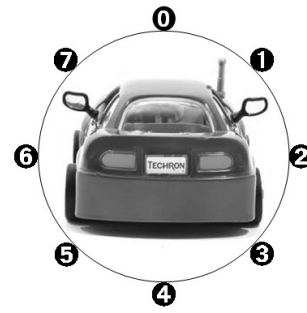
방법	데이터		이용자		설명	서비스
	위치	방향	위치	방향		
1	○	×	×	×	이용자의 위치와 무관, PC로 LBS 제공	위치 기반
2	○	×	○	×	이용자의 위치 고려, 단말기로 인접 관심지점 제공	
3	○	×	○	○	이용자의 방향정보를 이용, 데이터의 방향과 무관	위치 및 방향 기반
4	○	○	○	×	이용자의 방향과 무관, 데이터의 방향을 고려	
5	○	○	○	○	이용자의 방향정보와 데이터의 방향을 모두 고려	

것이고, 네 번째 방법은 네비게이션(GPS+Map+SW) 장비의 경로찾기 기능과 연계해 지나갈 경로로부터 앞으로의 이동 방향을 계산하는 것이다. 본 논문에서는 나침반 대신 최소 두 지점 이상을 연결하는 벡터를 이용해 계산하는 세 번째와 네 번째 방법을 사용한다.

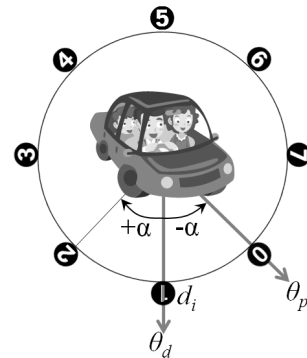
관심방향이란 이용자의 이동방향을 기준으로 전방, 우측 등과 같이 관심있는 방향을 말한다. (그림 3)과 같이 ①에서의 검색결과에 U턴해야만 하는 지점(좌전방,④)을 제외하고 싶은 경우나, 이동방향을 기준으로 특정방향을 검색하고 싶을 때 관심방향이 필요하며 도로조건에 따라 그 방향은 달리 설정될 수 있다. 같은 직선거리의 길 건너편 지점으로 이동할 때, U턴이 자유로운 국내 지방도로나 임의 U턴이 대부분 가능한 특정국가의 도로의 경우라면 거리증가가 미미할 것이다. 그러나 고속도로처럼 주변에서 U턴을 할 수 없는 곳이라면 직선거리와 실제 도달거리와는 많은 차이가 발생한다.

(그림 3)에 직선거리와 실제거리의 차이를 보이고 있는데 ①에서 ④가 검색된 경우 이들 사이의 직선거리는 짧지만 실제 이동거리는 도로 사정에 따라 달라지게 된다. 즉, ①→②→④로 이동하는 경우는 별 문제가 안되지만 ①→②→③→④로 이동할 수밖에 없는 경우는 이동거리가 길어져 검색에서 제외되거나 우선순위가 밀려나야 적합하다. 이러한 문제를 해결하기 위해 DNN에서는 이동객체는 물론 검색객체의 방향정보를 이용하였으나 이를 위해서는 모든 객체 데이터에 방향정보를 수록해야 하는 부담이 따른다. 뿐만 아니라 (그림 3)의 ㉠에서 진행방향과 동일한 ㉢검색객체는 DNN의 방법에 따라 직선거리도 짧고 두 객체와의 방향도 일치하므로 검색대상이 되지만 실제로는 ㉠→②→③→㉢와 같이 실제 이동거리는 길어질 수밖에 없다. 이에 대한 근원적인 문제해결은 네비게이션의 경로찾기 기능을 이용할 수밖에 없는데 본 논문은 <표 1>의 방법3에 해당되므로 모든 검색 데이터의 방향을 기록하는 대신 미리 설정된 이용자의 관심방향을 이용한다. 또한 본 논문에서는 여러 이유로 네비게이션의 경로찾기 기능을 이용하지만 이 기능 자체는 본 논문에서 다루고자 하는 대상이 아니므로 논외로 한다.

(그림 4)에 이용자가 설정하는 관심방향을 표시하였으며, 8방향인 ①~⑦ 중 하나를 미리 설정하게 된다. 일반적으로는 이용자의 이동방향(①)이 관심방향이 되지만 고속도로의 경우는 전방이라 할지라도 길 건너편이라면 방문하기가 곤란하므로 우전방(①)에 관심을 둘 수 있으며 도심지라면 ①이나 ①로 설정하는 경우가 일반적인 것이나 다른 방향의 설정이 필요할 수도 있다.



(그림 4) 관심방향 설정



(그림 5) 방향정보

검색각도란 검색방향을 기준으로 좌우 몇° 만큼 검색 허용각도로 할 것인지 설정하는 값으로 이 검색각도 또한 이용자가 사전에 0~180의 값을 입력한다. 일반적인 방법으로 현위치와 이동방향을 기준으로 모든 검색 데이터에 대해 질의를 수행하여 순위가 부여된 검색결과를 상위 일부분만으로 제한하게 되면 이용자가 원하는 검색반경과 검색각도내에서의 결과값을 얻을 수 있다. 그러나 본 논문은 PC가 아닌 모바일 단말기 상에서 수행되어야 하는 만큼 검색범위를 사전에 축소하기 위해 검색반경과 검색각도를 사전에 입력 받게 된다. 즉, 검색각도를 좌우 45°로 설정하게 되면 기존 NN 검색의 1/4 영역만을 대상으로 하기에, 관심없는 방향의 데이터는 설정값 범위만큼 검색에서 제외할 수 있게 되며, 180°로 하게 되면 NN과 동일하게 되어 이용자의 선택에 따라 검색범위를 자유자재로 조절할 수 있다.

이동방향을 자주 변할 수 있기 때문에 검색지점에 이동방향을 계산하고 (식 1)로부터 상대적 관심방향에 의한 절대방위각 즉, 검색방향을 계산하고 여기에 검색각도를 적용해 최종적으로 검색영역을 계산한다. 본 논문은 방향성분이 없는 <표 1>의 방법3에 해당되는 데이터를 대상으로 하지만 이동객체와 데이터 사이에는 거리뿐만 아니라 이동방향과의 각도 차이가 존재하므로 이들은 검색결과에 영향을 미친다.

(그림 5)에 검색영역을 계산하기 위한 방향정보들 간의 관계를 보이고 있다. 만약 이동방향이 135°(θ<sub>p</sub>=135)이고 관심방향이 ①로 설정(d<sub>i</sub>=1)되어 있다면 검색방향은 180°(θ<sub>a</sub>=180)가 되며, 좌우 검색각도가 ±45°로 설정(α=45)되어 있다면 검색영역(θ)은 135°~225°가 된다.

$$\theta_a = (\theta_p + d_i * 45^\circ) \text{ modulo } 360$$

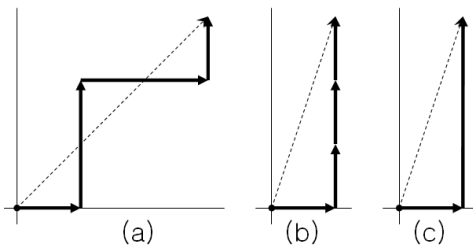
$$\theta_i = \theta_a - \alpha \sim \theta_a + \alpha \quad (1)$$

- θ<sub>p</sub>: 이동방향(0° ≤ θ<sub>p</sub> < 360°)
- d<sub>i</sub>: 관심방향 설정번호(0,1,2,...,7)
- θ<sub>a</sub>: 검색방향(0° ≤ θ<sub>a</sub> < 360°)
- ±α: 좌우 검색각도(0° < α ≤ 180°)
- θ: 검색영역

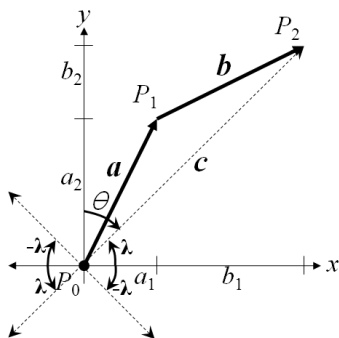
### 3.2 이동방향의 평균

3.1에서 언급한 세 번째와 네 번째 방법에 의한 이동방향은 최소 두 점의 좌표로 구할 수 있으며 현재좌표를 기준으로 이전좌표 혹은 다음좌표를 선택할 수 있다. 이미 지나온 경로상의 이전좌표들과 현재좌표로 계산된 방향은 과거의 이동방향 성격이 짙으며 현재좌표와 앞으로 지나갈 경로상의 다음좌표들로 계산된 방향은 미래의 이동방향 성격이 짙다. 네비게이션의 경로찾기 기능을 통해 앞으로 지나갈 경로들을 알 수 있다면, 두 개의 좌표보다는 여러 개의 좌표들을 통해 앞으로 진행할 방향의 평균을 구하는 것이 유리하다. 시골길처럼 구불구불한 도로나 도심지의 각 교차로, 골목길 등을 지그재그로 통과하는 경우처럼 이동방향이 짧은 시간 내에 바뀌는 경우, 이를 그대로 반영해 시시각각 변하는 방향을 얻기보다는 평균을 구하게 되면 완만한 이동방향의 변화를 얻을 수 있기 때문이다.

(그림 6)에서 굵은 실선의 화살표 하나가 두 좌표에 의한 이동방향이라면 이 각각의 이동방향은 벡터가 되며 (a)에서 4개 벡터에 대한 방위각(90°,0°,90°,0°)의 평균은 점선 화살표에 해당하는 45°로 이는 벡터합으로 계산된다. 이제 두 벡터 (a, b)의 평균을 위한 벡터합  $c = a + b$ 는 (그림 7)과 같이 나타낼 수 있으며 c의 x축에 대한 각도  $\lambda$ 는 알려진 (식 2a)로, 구하고자 하는 방위각  $\theta$ 는 (식 2b)로 구한다.



(그림 6) 벡터합에 의한 이동방향의 평균



(그림 7) 두 벡터합

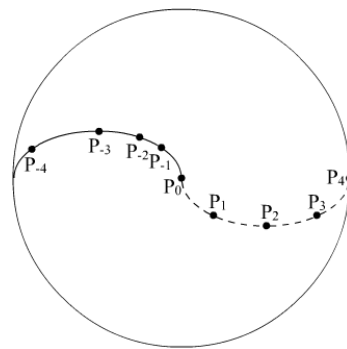
### 3.3 궤적의 참조점 설정

이동방향의 평균을 구하기 위해서는 현 위치를 포함한 최소 3개 이상의 경로상 지점을 설정해야 2개 이상의 벡터로부터 벡터합을 계산할 수 있다. 여기서 설정된 각 참조점들로 이루어진 경로를 궤적이라 하며, 현 위치에서 가장 멀리 떨어진 참조점이라도 검색반경을 넘어서는 것은 무의미하므로 검색반경 내에서 일정 개수로 위치를 선정한다.

궤적은 검색반경 내에서 과거 지나온 경로상의 참조점들로 이루어진 과거궤적과, 현재 위치에서 앞으로 지나갈 경로상의 참조점들로 이루어진 미래궤적 두 종류가 있다. 그러나 과거궤적을 통해 앞으로의 이동방향을 예측한다는 것은 어려운 일이다. 앞으로의 이동방향은 목적지와 관련있을 뿐 이전좌표들과는 무관하기 때문이다. 따라서 과거궤적의 참조점들로 벡터합을 구한 이동방향이나 이전좌표와 현재좌표 두 지점만으로 구한 이동방향이나 올바른 이동방향을 예측하기에는 무리이므로 계산량이 적은 두 좌표에 의한 방향을 구하도록 한다. 네비게이션이 있는 경우에는 목적지까지의 경로찾기 기능을 이용해 지나갈 경로상에 참조점을 설정한 미래궤적이 가능하다. 현 위치를 포함한 몇 개의 참조점으로부터 벡터합을 구하면 현 위치에서 가까운 미래의 이동방향을 계산할 수 있다.

(그림 8)은 현재위치 P0를 중심으로 검색반경 내에서 이미 지나온 경로(실선)와 앞으로 지나갈 경로(점선) 상에서의 궤적을 보여주고 있다. GPS만 있는 경우라면 과거궤적으로 2개의 참조점 P-1, P0만으로 방향을 구하며, 네비게이션이 있는 경우라면 미래궤적으로 최소 2개의 참조점 P0, P1(혹은 그 이상)으로 이동방향을 구한다.

궤적의 참조점을 설정하는 방법은 궤적종류에 따라 다른데 GPS만 사용하는 과거궤적의 경우 일정시간 단위(예: 10초)로 위치를 기록하되 현 위치에서 이전 위치의 좌표만 기억하면 된다. 네비게이션을 사용하는 미래궤적은 지도정보



(그림 8) 과거, 미래의 궤적 데이터

$$\tan \lambda = \frac{a_2 + b_2}{a_1 + b_1}, \quad \lambda = \tan^{-1} \frac{a_2 + b_2}{a_1 + b_1} = \tan^{-1} \frac{P_{2y} - P_{0y}}{P_{2x} - P_{0x}} \text{ (degree)} \quad (2a)$$

$\therefore$  방위각  $\theta$ (degree)는  
 if  $(P_{2x} - P_{0x}) \geq 0$  then  $\theta = 90^\circ - \lambda$   
 else  $\theta = 270^\circ - \lambda$  (2b)

를 이용하여 일거리 단위로 측정하여 설정하는 방법과 교차로 단위로 설정하는 방법, 또 이 둘을 혼합한 방법 등을 고려해 볼 수 있다. 본 논문은 혼합한 방법을 사용하며, 미래 궤적의 참조점은 경로찾기에 의한 경로상에서만 존재한다.

미래 궤적의 참조점을 선정하는 경우 시내에서의 교차점 간에는 대부분 직선도로이므로 교차점 단위로 참조점을 설정하면 되지만 때에 따라서는 교차점 간의 거리가 길어서 설정된 참조점 수보다 적을 수도 있다. 특히 국도에서는 검색반경내 교차점이 없으면서 곡선구간인 경우도 많은데 이처럼 교차점 간의 거리가 길 경우에는 거리단위로 참조점을 설정할 필요가 있다. 다음의 알고리즘 1은 미래 궤적 상의 참조점( $P_i$ ) 설정을 위한 것으로, 주기적인 검색이 일어날 때마다 미리 설정된 검색반경과 참조점 수에 따라 참조점들의 위치를 얻게 된다.

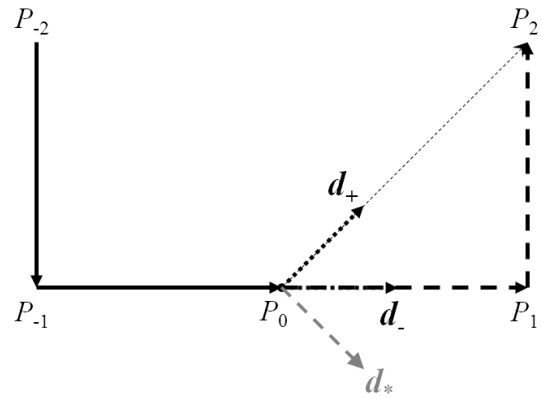
[알고리즘 1]

```

R = 검색반경;
n = 설정된 참조점의 수;
AvrP = R/n; //참조점간의 평균거리
for (i=1; i<=n; i++) {
    C =  $P_{i-1}$ 에서 진행경로의 인접 교차점;
    d =  $|P_{i-1}-C|$ ; //두 지점간 거리
    if (d <= AvrP)
         $P_i = C$ ; //교차점 C를  $P_i$ 로 설정
    else
        // $P_{i-1}$ 에서 AvrP만큼 떨어진 위치를  $P_i$ 로 설정
         $P_i = P_{i-1} + AvrP \cdot i$ ;
}
    
```

3.4 가중치 벡터합

이동방향의 평균값을 계산하기 위해 2개 이상의 벡터로부터 단순 벡터합을 사용할 경우, 이에 의한 이동방향의 결과는 (그림 9)와 같은 문제점을 보여준다. 이용자는  $P_{-2}$ ,  $P_{-1}$ 을 지나  $P_0$ 에 현재 위치하고 있으며 경로탐색에 의해  $P_1$ ,  $P_2$ 로 가고자 한다.  $P_0$ 에서 과거궤적을 이용한 이동방향은  $d_{-}$ 이며 미래궤적을 이용한 이동방향은  $d_{+}$ 가 된다. 만약 과거궤적에 벡터합을 적용했다면 이 경우에는 이동방향과 더욱 거리가 먼  $d_{*}$ 가 되었을 것이다.



(그림 9) 단순 벡터합에 의한 평균 이동방향

미래 궤적에서 머지않아  $P_2$ 로 방향전환할 것을 알고 있으면서도 네비게이션이 없어 과거 궤적의 방향인  $d_{-}$ 로 하는 것은 불가피한 문제로 큰 문제가 되지는 않는다. 그러나 미래 궤적 상에서 이동방향을 단순히 벡터합에 의한 이동방향인  $d_{+}$ 로 하는 것은 아쉬움이 있는데 그것은 가장 가까운 미래의 이동방향인  $P_0 \sim P_1$ 에 의한  $d_{-}$ 와 차이가 크기 때문이다.

이와 같은 문제점을 해결하고자 가중치에 의한 벡터합으로 이동방향의 평균을 계산하여 검색하는 방법(VNN: weighted Vector sum based NN)을 제안한다. (그림 9)에서 바람직한 형태는 이동방향이  $d_{-}$ 에서  $d_{+}$ 사이가 되는 것으로 현재의 방향에 보다 더 가중치를 주는 즉,  $\overline{P_0P_1}$  벡터에 가중치를 더 높게 주는 방법이다.

인접한 두 좌표  $P_i$ ,  $P_j$ 에 의한 벡터를  $V_i$ 라 할때 현재좌표  $P_0$ 와 인접좌표  $P_1$ 의 벡터  $V_0$ 에 대한 가중치를  $\omega(0 \sim 1)$ 라 한다면  $P_1$ ,  $P_2$ 의 벡터  $V_1$ 의 가중치는  $1-\omega$ 가 되며, 현재좌표  $P_0$ 에서의 가중치 벡터합  $d_{+}$ 는 현 위치를 포함한 궤적 데이터를  $n+1$ 개( $P_0 \sim P_n$ ,  $V_0 \sim V_{n-1}$ ) 사용한다고 할때 (식 3a)로 계산된다. 이로서  $d_{+}$  벡터의 방향성분인  $\lambda_{+}$ 는 (식 2a)로부터 (식 3b)와 같이 나타낼 수 있으며 이로부터 방위각  $\theta$ 는 (식 2b)로 구한다.

(식 3a)의  $d_{+}$ 에서 보듯이 세 좌표에 의한  $V_0$ 와  $V_1$ 가 가장 큰 영향을 미치며 나머지 벡터들은 뒤로 갈수록 그 영향력이 미미해지므로 필요 이상의 참조점 설정은 무의미하며

$$d_{-} = V_{-1}$$

$$d_{+} = \omega V_0 + (1-\omega)V_1 + (1-\omega)^2 V_2 + \dots + (1-\omega)^{n-1} V_{n-1} = \omega V_0 + \sum_{i=1}^{n-1} (1-\omega)^i V_i \tag{3a}$$

$$\lambda_{-} = \tan^{-1} \frac{y_1 - y_0}{x_1 - x_0}$$

$$\lambda_{+} = \tan^{-1} \frac{\omega(y_1 - y_0) + (1-\omega)(y_2 - y_1) + \dots + (1-\omega)^{n-1}(y_n - y_{n-1})}{\omega(x_1 - x_0) + (1-\omega)(x_2 - x_1) + \dots + (1-\omega)^{n-1}(x_n - x_{n-1})} \tag{3b}$$

$$= \tan^{-1} \frac{\omega(y_1 - y_0) + \sum_{i=1}^{n-1} (1-\omega)^i (y_{i+1} - y_i)}{\omega(x_1 - x_0) + \sum_{i=1}^{n-1} (1-\omega)^i (x_{i+1} - x_i)}$$

PC에 비해 계산능력이 떨어지는 단말기 특성을 고려한다면 3~5개가 적절할 것으로 생각된다.

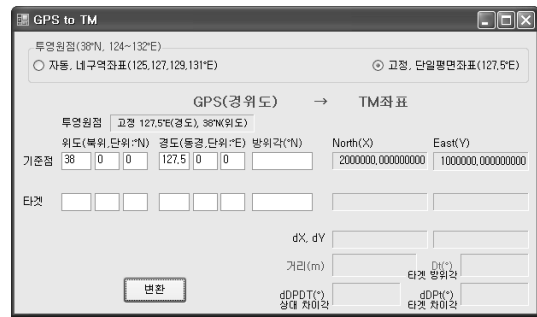
$d$ -나  $\omega=1$ 인  $d+$ 인 경우는 나침반을 사용하여 현재의 방향을 사용하는 기존 방법과 유사하며 벡터합을 계산하더라도  $V_0$ 에 높은 가중치( $\omega=0.9$ )를 주거나 아니면  $V_1$ 과 같은 가중치( $\omega=0.5$ )를 주는 등  $\omega$ 값을 자유자재로 조절함으로써 다양한 경우에 적용 할 수 있다. 예를 들어 검색 미디어가 많이 분포되어 있는 지역이면서 방향변화가 잦은 도심지를 지날 때, 많은 미디어 중에서 일단 현재 이동방향에서 근거리에 위치한 정보만을 더 보고 싶어한다면, 현재벡터의 비중을 높게 두기 위해  $\omega$ 를 높게 설정하면 된다.

### 3.5 좌표변환

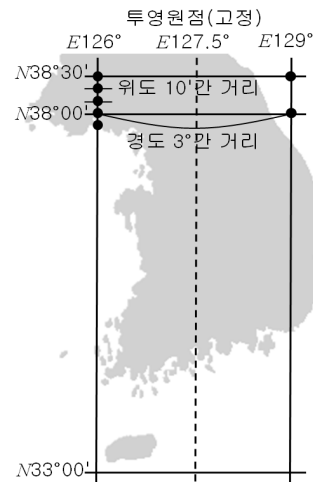
앞서 언급한 참조점들의 좌표는 GPS로부터 위경도 좌표를 얻게 되는데 이는 지구 표면의 특정 위치를 나타내는 좌표로서 좁은 면적에서는 평면으로 간주해도 무방하나 국가 전체를 대상으로 한다면 위경도 특성상 같은 수치의 위경도 거리에 따른 실제 거리차이가 발생한다. 극적인 예로, 위도 90°의 북극/남극에서 경도 1°의 거리는 최소(0m)이지만 위도 0°인 적도에서 경도 1°의 거리는 최대(약 111Km)가 된다. 따라서 위경도 1°간의 실제거리는 측정지점의 위도에 따라 크게 차이가 나며 측정지점의 경도에 따라서도 약간의 차이가 발생하지만 위도에 비해 미미하다. 또한 위도 1°와 경도 1°간의 거리는 서로 같지 않다. 이는 지구가 완전한 구(sphere)가 아니라 타원체이며 이 또한 완전한 수학적 모형이 아니기 때문이다[11]. 이를 위해 실제 지구타원체에 최대한 가깝게 수학적 모델링을 세워서 근사치를 얻을 수 있는 방법들이 제공되고 있으며, 이러한 변환공식을 이용하면 본문에서 대상으로 하는 위치 및 이동방향 기반 검색 서비스에 사용하기에는 충분한 정확도를 가진다.

현 위치를 기준으로 반경 몇m 이내의 내용을 검색하고자 하는 위치기반 검색 서비스를 위해서는 위경도가 아닌 미터법에 의한 거리표기가 필요하다. 이때 경위도 좌표(경도 $\lambda$ , 위도 $\phi$ )를 TM(Transverse Mercator) 좌표(East:Y, North:X)로 변환하는데 이는 지구타원체를 모델링하여 직교 평면 좌표 형태로 맵핑한 것으로 국토지리정보원에서는 다음과 같은 변환공식을 사용한다. 보다 세부적인 공식은 관련문헌[12]에 나와 있다.

구면상의 좌표인 경위도 좌표를 평면직각좌표로 변환할 때는 투영도법을 적용하는데 우리나라는 4개의 구역좌표별 투영원점과 1개로 된 고정된 투영원점이 있다[13]. (그림 10)은 (식 4)를 이용하여 만든 좌표변환 프로그램이다.



(그림 10) 좌표변환 프로그램



(그림 11) TM좌표값 측정위치

거리변환은 이동중에 GPS로부터 얻는 위경도 좌표를 대상으로 수시로 이루어져야 하므로 단말기의 CPU 성능이 좋은 경우라면 (식 4)의 변환공식을 이용하여 직접 변환해도 되지만 그렇지 못한 경우라면 CPU에 많은 부하를 주므로 좌표변환 프로그램을 이용해 위도별로 미리 계산된 테이블을 이용하는 방식을 사용한다.

측정간격은 테이블크기에 따른 메모리 증가와 정확도를 고려해 (그림 11)과 같이 위도 N38°30'에서 N33°00'까지 10' 단위로 변경해가며 측정하였으며 경도 변화에 대해서는 위도에 비해 거리차가 미미하기 때문에 고정된 투영원점 127.5°를 기준으로 좌우 1.5°씩 떨어진 두 지점만으로 측정하였다. 이렇게 계산된 TM좌표값은 <표 2>의 좌측영역(좌표값)에 기록하고 (식 5)에 의해 각 위도별 변화에 따른 위도 1", 경도 1"간 거리(m)로 계산되어 <표 2>의 우측영역(음영)에 기록하였다. 여기서 ←은 E126°의 N좌표와 동일함을 뜻한다.

$$\begin{aligned}
 Y(E) &= \Delta Y + k_0 \cdot N \cdot \left[ A + \frac{A^3}{6}(1 - T + C) + \frac{A^5}{120}(5 - 18T + T^2 + 72C - 58e'^2) \right] \\
 X(N) &= \Delta X + k_0 \cdot \left\{ M - M_0 + N \tan \phi \cdot \left( \frac{A^2}{2} + \frac{A^4}{24}(5 - T + 9C + 4C^2) \right) \right. \\
 &\quad \left. + \frac{A^6}{720}(61 - 58T + T^2 + 600C - 330e'^2) \right\} \quad (4) \\
 \Delta Y, \Delta X: & Y\text{축}(East), X\text{축}(North) \text{원점 가산값}, k_0: \text{원점축적계수}, \dots
 \end{aligned}$$

〈표 2〉 좌표변환 테이블

경도 좌표 위도	E126°		E129°		1"간 거리(m)	
	N좌표(X)	E좌표(Y)	N좌표(X)	E좌표(Y)	위도( $d_{\phi_i}$ )	경도( $d_{\lambda_i}$ )
N38°30'	2056131.259	897635.721	←	1102364.279	<b>30.81868</b>	<b>18.95635</b>
N38°20'	2037640.052	897163.264	←	1102836.736	<b>30.81786</b>	<b>19.04384</b>
N38°10'	2019149.337	896691.466	←	1103308.534	<b>30.81704</b>	<b>19.13121</b>
N38°00'	2000659.113	896220.342	←	1103779.658	<b>30.81622</b>	<b>19.21846</b>
N37°50'	1982169.379	895749.910	←	1104250.090	<b>30.81541</b>	<b>19.30557</b>
⋮	⋮	⋮	⋮	⋮	⋮	⋮
N33°00'	1446174.670	882478.463	←	1117521.537	<b>30.79271</b>	<b>21.76325</b>
N32°50'	1427699.042	882036.017	-	-	-	-

$$\begin{aligned}
 D_{\phi_i} &= |N_{\phi_i} - N_{\phi_i-10'}| \times 6 \\
 D_{\lambda_i} &= |E_{\phi_i, \lambda_{129}} - E_{\phi_i, \lambda_{126}}| / 3 \\
 d_{\phi_i} &= D_{\phi_i} / 3600, \quad d_{\lambda_i} = D_{\lambda_i} / 3600
 \end{aligned}
 \tag{5}$$

- $N_{\phi_i}$  : 위도  $i$ 에서의 N좌표
- $D_{\phi_i}$  : 위도  $i$ 에서의 위도 1°간 거리(m)
- $E_{\phi_i, \lambda_j}$  : 위도  $i$ , 경도  $j$ (=129,126)에서의 E좌표
- $D_{\lambda_i}$  : 위도  $i$ 에서의 경도 1°간 거리(m)
- $d_{\phi_i}, d_{\lambda_i}$  : 위도  $i$ 에서 위도 및 경도 1"간 거리(m)

〈표 2〉에서 보듯이 위도 10'단위의 변화에도 1"간 위도 및 경도 거리가 달라지는 것을 볼 수 있으며 특히 경도 거리가 크게 변하는 것을 알 수 있다. 〈표 2〉에서 단말기에 기억되는 내용은 전체가 아니라 위도별 측정위치에 따른 1"간 거리값이며 이는 두 위경도 좌표의 거리계산에 이용된다.

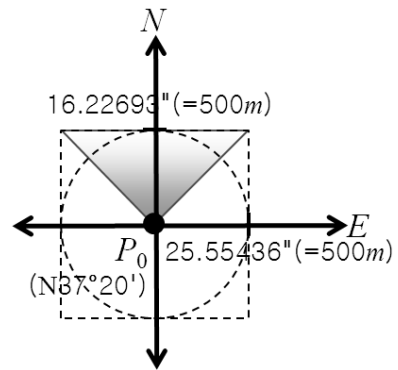
### 3.6 검색영역 및 검색 알고리즘

검색영역은 2단계에 걸쳐서 필터링 되며 첫 단계는 전체 미디어 중 검색반경 내에 존재하는 데이터만을 추출하며 두 번째 단계는 여기서 다시 검색방향에 부합하는 일정 범위 각도 내의 영역내에 들어오는 데이터만을 골라낸다. 여기서 이들 간의 검색순위를 부여할때 거리뿐만 아니라 데이터와 검색방향과의 각도차이를 반영하게 되며 이는 방향성분을 고려한 최근접점의인 DNN과의 비교를 위해 동일한 방법[9]을 사용한다.

본 논문에서는 첫 단계의 거리계산을 위해 두 지점의 위경도 좌표로부터 거리를 얻을 수 있는 (식 6)을 사용하지 않고 (그림 12)와 같은 방법을 사용한다. 즉, 이용자에 의해 주어진 검색반경(m)을 〈표 2〉를 이용해 같은 거리의 위도 및 경도 값을 먼저 알아낸 후 두 지점의 위경도값 차이가 모두 이 범위 내에 존재하는지의 여부를 확인하는 것이다. 이로 말미암아 (식 6)의 -, \*, +, ()<sup>2</sup>, √ 연산 대신 -연산으로 대체가 가능하다.

$$d_{0,T} = \sqrt{(|\phi_0 - \phi_T| * d_{\phi_i})^2 + (|\lambda_0 - \lambda_T| * d_{\lambda_i})^2}
 \tag{6}$$

- $\phi_0, \lambda_0$  : 현위치의 위도 및 경도
- $\phi_T, \lambda_T$  : 데이터의 위도 및 경도
- $d_{\phi_i}, d_{\lambda_i}$  : 위도  $i$ 에서 위도 및 경도 1"간 거리(m)
- $|\phi_0 - \phi_T|$  : 두 지점의 위도차("단위)
- $|\lambda_0 - \lambda_T|$  : 두 지점의 경도차("단위)
- $\phi$  :  $\phi_0$ 에 인접한 테이블 상의 위도  $i$



(그림 12) 검색영역

(그림 12)에서 검색영역은 일반적인 원형이 아닌 정사각형으로 설정되어 계산량을 줄일 수 있으며 이때 검색반경이 500m로 주어졌다면 현 위치  $P_0$ (N37°20')에서 실제측정거리가 N, E축 모두 동일한 500m가 되기 위해 계산한 결과, 위도(N축)와 경도(E축)별 거리는 각각 16.22693", 25.55436"가 됨을 알 수 있다. 물론 이 값들은 위도에 따라 달라진다.

본 논문에서의 거리에 의한 검색영역은 1단계에서 (그림 12)의 정사각형 영역으로 제한되며, 검색방향이 0°이고 검색각도가 ±45°로 주어진다면 2단계의 검색영역은 -45°(315°)~+45°에 해당하는 역삼각형이 된다. 이 영역 내의 미디어 데이터에 한해서 (식 6)에 의한 거리계산을 하게 되므로 전체 영역에서의 데이터는 물론 정사각형이나 원형 영역에서의 데이터보다 축소된 영역에서의 거리계산으로 계산량이 줄어든다.

다음 알고리즘 2는 이에 대한 검색방법으로 먼저 정사각형 영역안의 데이터를 출력(resultROI)하고 그 중에서 다시 역삼각형 영역안의 데이터를 출력(resultDOI)한다. (그림 13)은 이 알고리즘 2로 시뮬레이션하기 위한 프로그램의 화면 예이다. (그림 9) 및 (그림 13)의  $P_0 \rightarrow P_1 \rightarrow P_2$ 처럼 경도 10" 이동후 위도 10" 이동하는 경로에서 현재벡터의 방위각은 90°이지만, 단순 벡터합의 방위각은 45°가 아닌 그보다 작은값(그림 13의 계산결과 32.4°)이 된다. 그 이유는 같은 위경도 10"라도 실제 m단위 거리는 위도가 더 멀기 때문이다. 그러나 현재벡터에 더 큰 가중치를 주는 가중치 벡터합의 방위각은 (그림 7)의  $d_+ \sim d_-$  사이(그림 13의 계산결과 56.0°)가 된다.



[알고리즘 2]

```

A1, A2 = 검색영역; //범위:A1~A2
P0 = 현위치 위도 및 경도;
meterPerCho = 위경도 1"당 거리(m);
R = 주어진 검색반경(m);
Rlat = R / meterPerCho.위도; //R의 위도(")
Rlon = R / meterPerCho.경도; //R의 경도(")
for (i=0; i<데이터갯수; i++) {
    Xi = i번째 미디어데이터의 위경도;
    Cho = distGPSbyCho(P0, Xi); //위경도 차(")
    //Cho가 "단위의 검색반경 내 인가?
    if (|Cho.위도|≤Rlat && |Cho.경도|≤Rlon) {
        //정사각형 영역안에 존재(1차필터링)
        resultROI.text += (char)('A'+i)+" ";
        //조단위 거리차를 meter단위로 환산
        Met = Cho2Met(Cho, meterPerCho);
        //P0와 Xi와의 meter단위(N,E축) 거리로
        //방위각 및 거리 계산
        dd = getDirDist(Met); //방위각&거리
        if ( (A1≥0 && (A1≤dd.방위각≤A2))
            || (A1<0 && (360+A1≤dd.방위각
                || dd.방위각≤A2)) )
            //검색방위각 영역안에 존재(2차필터링)
            resultDOI.text += (char)('A'+i)+" ";
    }
}
    
```

```

DirDist getDirDist(Distmeter Met)
{
    //두 지점 사이의 방위각 및 거리
    lam = tan-1(Met.위도/Met.경도);
    theta=(Met.경도>0) ? 90-lam : 270-lam;
    dist = sqrt(Met.위도2+Met.경도2);
    DirDist dd; //방위각&거리
    dd.방위각 = theta;
    dd.거리 = dist;
    return dd;
}
    
```

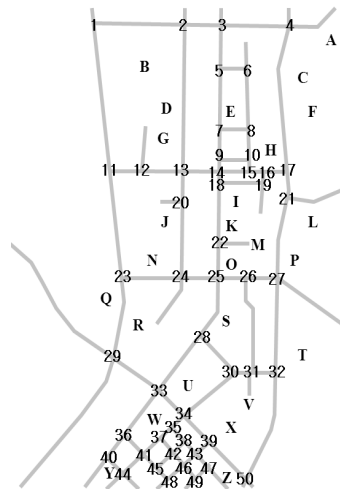


(그림 13) 검색 시뮬레이션

본 알고리즘 2에 의한 검색은 사용자가 이동 중 원하는 시점에 직접 검색을 할 수도 있지만 주기적으로 자동검색하는 방법이 적합하다. 즉, 이동시 현 위치를 기준으로 빠른 검색이 일어나되 그 검색 간격이 짧을수록 좋으나 단말기에 많은 부하를 주므로 아래에 제시한 두 가지 기준에 맞추어 검색시점이 결정된다.



(그림 14.a) 실험영역의 지도



(그림 14.b) 교차로 및 데이터 위치

첫 번째는 GPS만 있는 경우로 이때는 일정 시간(예: 10 초) 단위로 현 위치를 기록해 나감과 동시에 검색이 일어난다. 따라서 현 위치와 이전위치(예: 10초전)로 방향을 설정하고 검색이 일어난다. 따라서 이전위치의 참조점만 기억하면 된다. 두 번째는 네비게이션이 있는 경우로 이때는 방향이 바뀔 때마다 검색이 이루어져야 한다. 이를 위해 교차로 단위로 검색을 실시하며 만약 교차로가 멀리 떨어져 있는 경우는 일정 시간이 경과후 교차점에 도달하지 않았더라도 그 지점을 현 위치로 하여 검색이 일어난다. 그것은 일정 시간이 흐르면 현위치가 달라지고 이로부터 검색반경에 의한 검색영역이 변화되기 때문이다. 또한 방향성분으로 검색영역이 세분화되므로 일정거리만큼 떨어져 있으면서 방향이 바뀔 수 있는 교차로를 검색위치로 보는 것이다.

4. 실험결과

실험은 지도상의 각 교차로를 통과하며 이때의 GPS값을 바탕으로 검색반경 내에 존재하는 미디어 데이터 중에서 검색 방향에 부합하는 데이터를 검색하는 과정을 시뮬레이션 하였다. 본 실험에서는 네이게이션으로부터 교차로 정보를 얻는다고 가정하며 검색시점은 교차로 단위가 된다. 실험을 위해 먼저 (그림 14a)의 지도 영역에서 각 교차로 영역에 1부터 50까지 번호를 부여하고 미디어 데이터의 위치를 A~Z로 표기(그림 14b)하였다. 이들 각각의 위경도 좌표는 <표 3>과 <표 4>에 기록하였고, <표 5>는 시뮬레이션할 이동경로 5개를 선정 한 것이며, <표 6>은 실험에 사용된 설정값이다. (그림 14)에

<표 3> 각 교차로에서의 위경도 좌표

	위경도:37°,127°		위경도:37°,127°
1	24'19", 7'09"	26	23'41", 7'36"
2	24'19", 7'25"	27	23'41", 7'42"
3	24'19", 7'32"	28	23'31", 7'28"
4	24'19", 7'43"	29	23'29", 7'13"
5	24'13", 7'32"	30	23'26", 7'34"
6	24'13", 7'36"	31	23'26", 7'37"
7	24'03", 7'32"	32	23'26", 7'42"
8	24'03", 7'36"	33	23'24", 7'21"
9	23'59", 7'32"	34	23'20", 7'26"
10	23'59", 7'47"	35	23'18", 7'24"
11	23'57", 7'12"	36	23'17", 7'15"
12	23'57", 7'18"	37	23'16", 7'21"
13	23'57", 7'25"	38	23'16", 7'26"
14	23'57", 7'32"	39	23'16", 7'30"
15	23'57", 7'36"	40	23'14", 7'12"
16	23'56", 7'39"	41	23'14", 7'18"
17	23'57", 7'43"	42	23'14", 7'23"
18	23'55", 7'32"	43	23'14", 7'28"
19	23'56", 7'38"	44	23'11", 7'15"
20	23'52", 7'25"	45	23'12", 7'21"
21	23'53", 7'44"	46	23'12", 7'25"
22	23'45", 7'32"	47	23'13", 7'30"
23	23'41", 7'14"	48	23'10", 7'22"
24	23'41", 7'24"	49	23'10", 7'27"
25	23'41", 7'32"	50	23'10", 7'36"

<표 4> 각 미디어 데이터의 위경도 좌표

	위경도:37°,127°		위경도:37°,127°
A	24'17", 7'50"	N	23'43", 7'19"
B	24'12", 7'17"	O	23'43", 7'33"
C	24'11", 7'46"	P	23'44", 7'45"
D	24'07", 7'22"	Q	23'38", 7'12"
E	24'06", 7'33"	R	23'34", 7'17"
F	24'06", 7'47"	S	23'35", 7'33"
G	24'02", 7'21"	T	23'30", 7'46"
H	24'01", 7'40"	U	23'25", 7'26"
I	23'53", 7'34"	V	23'22", 7'37"
J	23'49", 7'22"	W	23'20", 7'20"
K	23'49", 7'33"	X	23'19", 7'34"
L	23'50", 7'48"	Y	23'12", 7'12"
M	23'46", 7'38"	Z	23'11", 7'33"

<표 5> 이동경로

	이동경로(미래계적)											
	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>	P <sub>9</sub>	P <sub>10</sub>	P <sub>11</sub>
①	1	11	23	29	33	34	39	50				
②	2	13	20	24	25	28	33	36	40			
③	3	5	7	9	14	18	22	25	26	27	32	50
④	4	17	21	27	32	31	30	34	33	29		
⑤	32	31	26	25	22	18	14	15	10	8	6	

<표 6> 실험 설정값

	설정값
검색반경	500m
관심방향	0(정면)
검색각도	±45°(총90°)
참조점수	3(미래계적)
가중치 $\omega$	0.7

서 위경도 1"간거리는 각각 30.81298m와 19.56613m가 되는데 이는 (그림 14)나 <표 3>, <표 4>의 위도영역인 37°24'~37°23'에 해당하는 <표 2>의 37°20'에서의 값에 따른 것으로 (그림 13)에도 표시되어 있다.

<표 7~11>은 <표 5>의 5개 이동경로에 대한 실험 결과이다. 기존 NN에 의한 검색결과는 알파벳으로 표시하였으며, 이 중에서 기울임 글꼴은 제한한 VNN에서  $\omega$ 를 1.0으로 하여 기존 DNN과 같은 방법으로 한 것이고 역상 원문자는  $\omega$ 를 0.7로 하여 2차 필터링한 결과이다. 이때 이동방향도 각각  $d_{+DNN}$ 과  $d_{+VNN}$ 에 표시하였으며  $\omega$ 가 1에서 작아질수록  $\omega=1$ 인 경우의 이동방향과 달라지게 된다. DNN과의 비교를 위한 동일 조건으로 관심방향을 0, 검색각도를 ±45°(검색영역은  $d_{-}(\theta)-45^{\circ} \sim d_{+}(\theta)+45^{\circ}$ )로 제한하였다. 또한 두 객체와의 방향각 차이에 대한 가중치를 DNN과 동일하게 하여 평가한 결과의 순위를 나타낼때, DNN은 아래첨자로 VNN은 위첨자로 표기하였다. 각 표의 맨 아래줄에는 26개 데이터 중 검색된 개수의 평균을 나타내었다. 도로구조에 따라 현재벡터( $\omega=1$ )와 가중치 벡터합( $\omega=0.7$ )에 의한 방향과는 차이가 큰 경우도 있고 그렇지 않은 경우도 존재한다. 차이가 큰 경우는 검색순위로 달라지게 된다. <표 8>의 P<sub>3</sub>이 그러한 예로 DNN의 경우 O,P,M,L,T 순으로 검색되었지만 VNN의 경우 S,T,U 순으로 검색되었다.

(그림 15)는 이동경로 ③에서의 검색결과(표 9) 중 교차로 P<sub>8</sub>에서 검색이 일어난 경우의 예이다. 교차로 P<sub>8</sub>(26번)에서 1차 필터링을 위한 검색반경(실선 정사각형)과 26, 27, 32번 교차로를 참조점 P<sub>0</sub>,P<sub>1</sub>,P<sub>2</sub>로 하는 경로에 대해 가중치 벡터 합에 의한 방위각  $d_{-}(\theta)=149.3^{\circ}$ (화살표)과 ±45°에 의한 검색 영역의 범위(음영)를 보이고 있다. 이 결과 1차 필터링에 의한 검색결과 12개 중 1개(T)만이 2차 필터링을 위한 음영영역에 속함을 보이고 있다. 반면, 현재벡터만을 이용한 기존 DNN과 같은 경우엔 P만이 검색됨을 알 수 있다. 이처럼 제한한 VNN은 가중치값을 조정함으로써 다양한 도로상황에 맞추어 적용할 수 있다.

(그림 16)은 <표 7~11> 각각의 평균값에 대한 차트이다. 사용자가 설정한 검색각도 내에서만 찾고자 하는 데이터가 100%

<표 7> 이동경로 ①에서의 검색결과

	검색결과	$d_{+DNN}$	$d_{+VNN}$
P <sub>0</sub>	B, D, E, G, I, K, O, S	175.1	175.1
P <sub>1</sub>	B, D, E, G, I, K, O, S	175.5	177.3
P <sub>2</sub>	I, J, K, M, N, O, S, Y	183.0	173.8
P <sub>3</sub>	N, O, Q, R, S, Y	134.5	136.2
P <sub>4</sub>	Q, R, S, T, U, W, Y	141.6	143.3
P <sub>5</sub>	R, S, T, U, V, W, X, Y	147.6	147.6
Avr	8.33(NN), 2.83(DNN), 2.83(VNN)		

<표 8> 이동경로 ②에서의 검색결과

	검색결과	$d_{+DNN}$	$d_{+VNN}$
P <sub>0</sub>	A, C, D, F	180.0	180.0
P <sub>1</sub>	B, C, D, E, F, G, H, I, L, M, N, O, P	180.0	181.6
P <sub>2</sub>	D, E, F, G, H, I, K, L, M, N, O, P	183.3	172.0
P <sub>3</sub>	I, J, K, L, M, N, O, P, Q, R, S, T	90.0	137.0
P <sub>4</sub>	I, J, K, L, M, N, O, P, Q, R, S, T	194.3	198.9
P <sub>5</sub>	M, N, O, P, Q, R, S, T, V, X	212.4	211.3
P <sub>6</sub>	Q, R, S, T, U, V, X, Z	208.6	209.2
Avr	11.86(NN), 4.17(DNN), 3.14(VNN)		

<표 9> 이동경로 ③에서의 검색결과

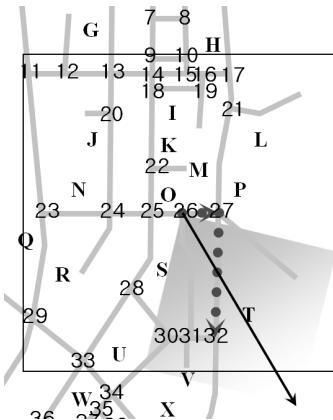
	검색결과	$d_{+DNN}$	$d_{+VNN}$
P <sub>0</sub>	A, B, C, D, F	180.0	180.0
P <sub>1</sub>	A, B, C, D, F, G, H	180.0	180.0
P <sub>2</sub>	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	180.0	180.0
P <sub>3</sub>	B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	180.0	180.0
P <sub>4</sub>	B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	180.0	180.0
P <sub>5</sub>	C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	180.0	180.0
P <sub>6</sub>	H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	180.0	164.8
P <sub>7</sub>	I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	90.0	90.0
P <sub>8</sub>	I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	90.0	149.3
P <sub>9</sub>	I, J, K, L, M, N, O, P, R, S, T, U, V, X, Z	180.0	184.3
Avr	12.10(NN), 4.00(DNN), 3.90(VNN)		

<표 10> 이동경로 ④에서의 검색결과

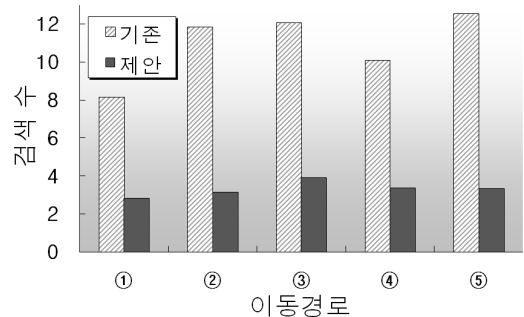
	검색결과	$d_{+DNN}$	$d_{+VNN}$
P <sub>0</sub>	A, D, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	180.0	179.3
P <sub>1</sub>	C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	171.0	179.4
P <sub>2</sub>	D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	186.0	183.9
P <sub>3</sub>	I, J, K, L, M, N, O, P, R, S, T, U, V, X, Z	180.0	185.2
P <sub>4</sub>	S, T, U, V, X, Z	270.0	270.0
P <sub>5</sub>	Q, R, S, T, U, V, X, Z	270.0	237.8
P <sub>6</sub>	Q, R, S, T, U, V, X, Z	220.3	236.4
P <sub>7</sub>	S, T, U, V, W, X, Y, Z	321.6	318.9
Avr	10.13(NN), 3.38(DNN), 3.38(VNN)		

<표 11> 이동경로 ⑤에서의 검색결과

	검색결과	$d_{+DNN}$	$d_{+VNN}$
P <sub>0</sub>	S, T, U, V, W, X, Z	270.0	331.8
P <sub>1</sub>	Q, R, S, T, U, V, W, X, Y, Z	357.6	353.4
P <sub>2</sub>	I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	270.0	304.0
P <sub>3</sub>	I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	0.0	0.0
P <sub>4</sub>	I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	0.0	0.0
P <sub>5</sub>	D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	0.0	28.6
P <sub>6</sub>	B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	90.0	81.2
P <sub>7</sub>	B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, X, Z	74.0	47.1
P <sub>8</sub>	C, D, E, F, G, H, I, J, K, L, M, O, P, Q, R, S, T, U, V, X, Z	299.8	319.9
Avr	12.56(NN), 3.33(DNN), 3.33(VNN)		



(그림 15) 표 9의 P8 예



(그림 16) 필터링된 검색 개수

존재하며 그 이외의 방향에서는 존재하지 않는다고 가정한다. 사전에 설정한 검색각도  $\pm 45^\circ$ 는  $360^\circ$  영역의 1/4에 해당하는 것으로, 실험에 사용된 데이터의 분포가 이상적으로 균일하다면 실험결과는 1차 필터링된 결과의 1/4만 남게 되어 불필요한 데이터 3/4은 제거된다. 그러나 미디어 데이터 26개가 (그림 14b)처럼 균일하게 분포되지 않은 경우 분포형태와 이동경로에 따라 (그림 16)처럼 그 결과값은 달라진다.

26개의 미디어 데이터 중 5가지의 이동경로에 대해 1차 필터링한 결과 전체평균은 10.96개(기존 NN에 해당)이며 2차의 전체평균은 약 3.32개(DNN은 3.54)로서 NN 대비 약 30%만 남기고 불필요한 70%는 제거되었다. 여기서 DNN과 VNN은 검색각도와 방향차이각에 대한 가중치를 동일한 조건으로 하였기에 최종 검색개수의 평균값은 큰 차이가 없어 도 개별 경로에서의 검색결과는 차이가 있음을 알 수 있다.

### 5. 결론

본 논문에서는 기존 NN 방법에서 벗어나 가중치 벡터합을 이용한 이동객체의 방향을 계산하고 이를 검색에 반영해 불필요한 방향의 데이터는 제거하는 방법을 제안 하였다. 단말기의 부하를 최소화하기 위해 이용자의 위치를 기준으로 정사각형의 검색영역으로 1차 필터링후, 미디어 데이터 자체의 방향성분이 없는 상태에서도 이용자의 방향정보(이

동방향+관심방향+검색각도)를 이용하여 검색영역을 사전에 축소하는 2차 필터링 단계를 거치는 검색방법을 제안하였다. 이때 이동방향 계산을 위해 현 위치에서의 측정방향을 사용하는 부정확한 방법 대신, 가장 가까운 과거벡터( $V_{-1}$ )를 사용(GPS만 있는 경우)하거나, 가장 가까운 미래벡터들( $V_0, V_1, V_2, \dots$ )간에 가중치를 둔 벡터합을 사용(네비게이션이 있는 경우)하여 이동방향을 계산할 수 있도록 하였다. 제안한 VNN은 벡터와 관심방향을 사용함으로써 데이터의 방향 성분이 기록되어 있지 않은 데이터에 효과적으로 적용할 수 있었다. 또한 가중치 조절을 통해 기존방법을 포괄하거나 다양한 도로환경에 유연하게 대처할 수 있음을 보였다.

### 참 고 문 헌

- [1] Kentaro Toyama, Ron Logan, Asta Roseway and P. Anandan, "Geographic Location Tags on Digital Images," *ACM MM'03*, California, USA, pp.156-166, Nov., 2-8, 2003.
- [2] Noah Snavely, Steven M. Seitz and Richard Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," *ACM 0730-0301/06/0700-0835*, pp.835-846, 2006.
- [3] Davide Carboni, Stefano Sanna and Pietro Zanarini, "GeoPix: Image Retrieval on the Geo Web, from Camera Click to Mouse Click," *ACM MobileHCI'06*, Helsinki, Finland, pp.169-172, Sep., 12-15, 2006.
- [4] MediaMapper <http://www.mediamapper.com>.
- [5] World Wide Media eXchange <http://wmx.org>.
- [6] Flickr <http://flickr.com>.
- [7] EXIF 표준 <http://www.exif.org/specifications.html>.
- [8] Apostolos N. Papadopoulos and Yannis Manolopoulos, *Nearest Neighbor Search - A Database Perspective*, Springer, 2005.
- [9] 이용재, 정영진, 최현미, 류근호, 이성호, "모바일 객체의 방향성을 고려한 최근접 질의 처리", *개방형지리정보시스템학회논문지 제6권 제1호*, pp.59-71, 2004.
- [10] Ken C.K. Lee, Wang-Chien Lee, Hong Va Leong, "Nearest Surrounding Queries," *IEEE ICDE'06*, 2006.
- [11] Andrew Jones and Graeme Blick, "Where in the World are We?, New Zealand Version 1.0," July 1999.
- [12] 2006.11.14 고시 공공측량의 작업규정 세부기준 운영세칙(국토지리정보원 제2006-608호)을 준용한 좌표변환
- [13] 측량법 제5조 2항의 규정에 의거 측량법 시행령 제2조 6항(개정 2004.7.20)



### 서 창 덕

e-mail : scd@daum.net

1988년 한양대학교 전자공학과(학사)

1990년 한양대학교 대학원 전자공학과  
(공학석사)

2000년 한양대학교 대학원 전자공학과  
(공학박사)

2000년~2001년 한양 BK21 Post-Doc.

2002년~2006년 한양대 HCI 연구원/경원전문대 겸임교수

2007년~현 재 Foto Fiddlers(NZ), IT Manager

관심분야 : 영상검색, LBS, 유비쿼터스 등



### 한 기 태

e-mail : gthan@kyungwon.ac.kr

1982년 충남대학교 계산통계학과(학사)

1990년 한양대학교 대학원 전자계산학과  
(공학석사)

2001년 한양대학교 대학원 전자공학과  
(공학박사)

1985년~1992년 한국주택은행 전산부 전산역

1992년~현 재 경원대학교 IT대학 소프트웨어학부 컴퓨터미디어전공  
교수

관심분야 : 영상처리, DB, LBS, 유비쿼터스 등