Update Policy and Estimation of Uncertain Position Using Trajectory Information

Tai-Jung Sim*, Jae-Hong Kim**, Won-Il Jung*, Yong-II Jang*, Hae-Young Bae*
*Dept of Computer Science & Engineering, In-Ha University
**Dept of Computer Science, Young-dong University

요 약

이동 단말의 보편화로 인해 이동 객체의 위치 정보를 기반으로 사용자에게 사물이나 사람, 장소등과 같은 이동 객체의 위치를 파악하여 그에 대한 정보를 제공해 주는 시스템이 필요로 하게 되었다. 이러한 이동 객체의 위치 정보를 제공하는 시스템은 주로 GPS 또는 DBMS와 같은 위치 정보를 제공하는 시스템을 활용하여 위치 정보를 제공하는 방식으로 구분된다.

이동 객체의 위치 정보는 주로 위치 정보를 제공하는 시스템에서 제공되며, 이는 위치 정보를 제공하는 시스템의 성능에 크게 영향을 미친다. 또한 이동 객체의 위치 정보는 이동 객체의 이동 경로를 추정하거나 예측할 수 있는 중요한 정보로 사용된다.

1. 서론

이동 객체의 위치 정보를 제공하는 시스템은 주로 위치 정보를 제공하는 시스템에서 제공되며, 이는 위치 정보를 제공하는 시스템의 성능에 크게 영향을 미친다. 또한 이동 객체의 위치 정보는 이동 객체의 이동 경로를 추정하거나 예측할 수 있는 중요한 정보로 사용된다.

이동 객체의 위치 정보를 제공하는 시스템은 주로 위치 정보를 제공하는 시스템에서 제공되며, 이는 위치 정보를 제공하는 시스템의 성능에 크게 영향을 미친다. 또한 이동 객체의 위치 정보는 이동 객체의 이동 경로를 추정하거나 예측할 수 있는 중요한 정보로 사용된다.

이동 객체의 위치 정보를 제공하는 시스템은 주로 위치 정보를 제공하는 시스템에서 제공되며, 이는 위치 정보를 제공하는 시스템의 성능에 크게 영향을 미친다. 또한 이동 객체의 위치 정보는 이동 객체의 이동 경로를 추정하거나 예측할 수 있는 중요한 정보로 사용된다.
2. 관련 연구
2.1 이동 개체 관리 시스템에서의 개선 정책

이동 개체 관리 시스템은 계절적으로 변화하는 이동 개체의 정보를 효율적으로 관리하여 DBMS에 저장하는 것이 중요하다. 이는 인터넷 경계로 인한 다수의 접근 비율이 증가하고 이

따라서는 DOMINO에서는 버전을 고려한 개선 정책을 이용한 개선 정책을 제공하고 있는 DBMS에 저장하는 것이 중요하다. 적절한 접근 위험이 있는 DBS의 접근 영향도 클록 보편적인 예측 모델을 적용해야 한다. 또한 서술의

변화율은 이용한 개선 정책은 이용 개체의 속도에 따라 단위

가장 것 같은 시점은 계산하여 개선 정책을 결정하기 위해서

의 정책적인 개선 정책의 영향력은 크고 단위 거리

당 개선의 영향을 입으므로 위치 정책의 정확성을 보장해 줄 수

있다고 클록 비용은 줄일 수 있다.

이와 같은 개선 정책으로는 개선 정책을 고

려하여 개선 정책을 결정하는 방법이 있다. 이는 이용

가져와의 현

재 상태에 대한 점의 처리를 위한 점의 벨레임은 두어 이동 개

체로부터 계산된 위치 정보와 점의 벨레임에는 저장된 위치 정보

을 비교하여 계산된 오차값을 더할 경우에만 벨레임을 사용할

다. 그러나 이 기법은 이용 개체에 적용되는 한계를 설정하는

것에 의하여 이용 개체로 고려되지 않으면 불확실한 위치 정

책의 추정에 있어 정확성을 보장하기 어렵다.

2.2 불확실한 위치 정보의 추정

DBMS에 정적으로 저장하지 않은 위치 정보를 사용자가 요구하였을 경우 시스템에서는 이를 계산하여 위치에 대한 결과를 제공해 주어야 한다.

위치 정책

\[ t \leq t' \leq t_{t+1} \]의 조건을 만족하는 시점 \( t \)의 위치 정책에 대한 추정을 할 경우 다음과 같은 식으로 계산한다.

\[ x(t) = x_k - \frac{x_k - x_{k+1}}{y_{k+1} - y_{k+1}} (t - t_k + y_{k+1} - y_{k+1}) + x_k \]

\[ y(t) = y_k - \frac{y_k - y_{k+1}}{x_{k+1} - x_{k+1}} (t - t_k + x_{k+1} - x_{k+1}) + y_k \]

미래의 위치 추정

미래의 위치 추정은 어떤 위치 정보를 저장하지는 않아 나

양한 개선 방법이 사용되지만 대부분의 위치 정보 추정은 수학적

방법을 이용하여 계산한다. 현재 위치 정보에 저장된 속도 속도

과 거리의 시간을 곱하여 이동 거리를 구한 후 방향 속도와 고려

하여 미래의 위치를 추정한다.

그러나 이러한 위치 추정 방법은 이동 정보를 고려하지 않고 있기 때문에 위치 검색을 사용할 경우 위치를 잘못 정렬할 경우 나올 수 있고 도로 상에서 이동 할 수 없는 지점을 미래의 위치값으로 추정할 수 있다.

3. 위상 정보를 고려한 개선 정책 및 불확실한 위치 추정
3.1 위상 정보를 고려한 개선 정책

본 논문에서는 최소한의 위치 정보를 정적으로 저장하여 집

의를 처리하기 위해 도로상의 개인 지점과 교차로의 차량이 도착

할 시점을 개선 시점으로 정한다.

<그림 1> 삼각형 도로와 연관정보

<그림 1>은 삼각형 도로와 그 위치정보를 나타낸 것이다. 이 도로는 노드 D에서 노드 B 방향으로 이동할 경우 우회로 즉 노드 A로의 이동이 가능하지만 좌회로 노드 D로의 이동은 불 가능하다. 위상 정보를 나타낸 각 노드와 라인의 정보를 다음과 같이 간단하게 나타낼 수 있다.

현재 전체의 이동 방향이 노드 D에서 노드 B 방향으로 속도 V로 이동하고 있다고 가정하면 교차로 노드 B에 도착하는 예상 시간 T1 = L/V로 계산할 수 있다.

<표 1> 라인 정보

<table>
<thead>
<tr>
<th>line ID</th>
<th>연결 노드 ID</th>
<th>라인 길이</th>
<th>라인 목</th>
</tr>
</thead>
<tbody>
<tr>
<td>line 1</td>
<td>(A, C)</td>
<td></td>
<td>L1</td>
</tr>
<tr>
<td>line 2</td>
<td>(B, C)</td>
<td>L2</td>
<td>W2</td>
</tr>
<tr>
<td>line 3</td>
<td>(B ,D)</td>
<td>L3</td>
<td>W3</td>
</tr>
</tbody>
</table>

<표 2> 교차점 노드 B의 정보

<table>
<thead>
<tr>
<th>노드 ID</th>
<th>FROM</th>
<th>TO</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>line 1</td>
<td>line 2</td>
</tr>
<tr>
<td>B</td>
<td>line 3</td>
<td>line 2</td>
</tr>
<tr>
<td>B</td>
<td>line 2</td>
<td>line 3</td>
</tr>
</tbody>
</table>

차량이 B요에 도착하면 다음 개선 시점을 결정하게 되는데 이때 차량은 ‘line 1’로 이동을 할 수 있으므로 노드B에서 노드A까지의 라인 길이를 토록하여 다음 개선 시점을 결정한다.

만일 교차점 차량이 도착했을 때 다음 예상 도로 시점의 속도

나머지 경우 가장 가까운 노드의 예상 도로 표지로 정하여 개

선 시점을 정하고 개선을 수행한다. 이때 개선을 수행한 지점

이 노드가 아닌 개인 상에 위치한 지점임 경우 다시 노드에 도착

할 시점로 계산하여 개선을 수행한다. 알고리즘은 <그림 2>와 같다.

<그림 2>의 알고리즘 상의 CalNextUpdateTime는 현재 위치, 라인 ID, 차량의 속도를 이용해 도로의 길이를 추정하고 이에 따

른 교차로 도로 시간을 추정한다. 이때 시간을 지정 후 도착한 지점 이 교차점이 아닌 위치 정보의 라인에 위치 할 경우 다음

1652
교차점까지의 시간을 다시 계산한다.

// 경신 시점 결정 알고리즘
POSITIONCurnPos : 현재 차량의 위치
LINE_ID = LINEID : 현재 차량이 위치한 LINE
begin
Update( CurnPos );
bOnNode = isOnNode( CurnPos );
if( bOnNode ) {
  nNodeNum = FindNextNodeNum();
  if( nNodeNum > 1 )
    nLineID = CalShortestLine();
    else
      nLineID = FindNextLine();
    T = CalNextUpdateTime( CurnPos, nLineID, V );
  }
else{
  nLineID = FindNextLine();
  T = CalNextUpdateTime( CurnPos, nLineID, V );
}
end

그림 2. 갱신 시점 결정 알고리즘

3.2 위상 정보를 고려한 불확실한 위치 추정

<그림 1>과 같은 도로에서 노드 D에서 노드 B로 이동하는 차량을 가정하고 노드 B에서 미래의 위치를 고려해보자. 노드 B에서의 현재 이동 방향은 line 3과 수평한 방향이다. 기존의 기법에서 사용하는 방법처럼 현재 위치의 이동 방향만을 고려하여 미래의 위치를 추정한다면 노드 B 지점에서부터 이 상 line 3과 수평한 방향으로 이동할 수 있는 경로가 존재하지 않기 때문에 차량이 이동하기에는 불가능한 지점이 위치로 추정될 것이다.

<그림 2>에서 노드 A와 노드 B에서의 차량 객체가 도착하였을 경우 이 두 지점의 정보가 저장되었다고 가정하자.

노드D에서의 도착 시간을 T1라고 하자. 가령 노드 A에서의 도착 시간은 T2이고 시점 사이의 중간 시점을 T3라 할 때 만약 정확적으로 저장된 두 점을 이용해 시점 T3의 위치 정보를 알면 이러한 기존의 기법에서는 불가능한 경로를 판단할 것이다. 그러나 실험에서는 노드 B와 노드D의 차량 상에 객체가 위치하고 있었을 것이다.

현재 차량의 위치와 위상 정보를 이용한 미래의 불확실한 위치 추정 알고리즘은 다음과 같다.

// 미래 위치 추정 알고리즘
// T1 : 미래 시점, V : 현재 속도, CurPos : 현재 위치
begin
// 시점까지 이동 거리 계산
nLength = CalLength( CurPos, V, T1 );
if( nLength <= CalLengthToNode( CurPos, CurLineID ) )
  FurPos = CurPos + V + T1;
else
  if( CalLineNum( ) == 1 )
    FurPos = CurPos + V + T1;
end

그림 4. 미래 위치 추정 알고리즘

미래 위치 정보를 추정할 때는 위치 추정이 가능한 한계값을 계산한다. 한계값은 T1 시점까지 차량이 이동할 거리와 차량의 거리를 비교하여 결정한다. 차량이 이동할 거리가 다음 도착 노드까지의 길이보다 작은 경우 위치 정보를 갱신할 수 있으나 길 경우 다음 도착 노드의 정보를 연계하여 이동 가능한 경로가 하나 없을 경우에만 위치 정보를 추정해본다. 이는 미래의 위치를 결정하고자 할 경우 다수의 결과값이 나오는 것을 막기 위함이다.

과거의 위치 추정도 위상 정보를 이용할 수 있다. 위치 정보는 도로상의 위치 정보 교차점으로 저장하여 최소한의 정보를 저장하는 경우에 시점 시점에 대한 시점에 해당하는 경우 이 두 시점 사이에는 미래의 위치로 추정할 수 있다. 따라서 과거의 위치를 추정할 때 두 시점 사이의 위치 정보에 노드가 존재하지 않아 갱신 방법이 달라진다.

// 과거 위치 추정 알고리즘(시점 Tb 정보가 저장되어 있지 않음)
// Tp : 과거 시점
begin
// Tb<Tp<Tn 시점 Tn 검색
SelTime( Tp, Tb, Tn ) : nNodeID = ExistNodeID( Tp, Tn )
 Va = ( nLengthp + nLengthm ) / ( Tn - Tp )
if( nNodeID != -1 )
  nLengthm = CalLength( Tb, nNodeID )
  nLengthp = CalLength( Tn, nNodeID )
  if( Va * ( Tp - Tb ) < nLengthm )
    PastPos = PosB + Va * ( Tp - Tb )
  else
    PastPos = PosNode + Va * nLengthm / Va;
  else
    PastPos = PosB + Va * ( Tp - Tb )
end

그림 5. 과거 위치 추정 알고리즘

본 논문에서는 제안한 위치 추정 알고리즘을 적용할 경우 과거 의 경신 속도의 갱신하기 위해 디스플레이된 필요 없고 차량의 이동 경로를 고려하고 있기 때문에 위치의 정확성을 사용자에게 보장해 줄 수 있다.

4. 결론

본 논문의 실험은 GMS/LBS 사용에 강점 지역 교통통보 이용용 시뮬레이션에 데이터를 저장하였다. 지도의 각 데이터는
전송, 도로, 도로명등의 레이어 파일로 이루어져있고 이와 더불어 위상 정보도 파일로 관리된다.

5초 간격 위치 횟목 | 위상 정보 위치 횟목
--- | ---
Select FoxX, FoxY | Select FoxX, FoxY
From KangNam0 | From KangNam
Where | Where

5. 결론 및 향후 연구
이동 객체의 위치 관리를 위해 지금까지 연구된 시스템들은 일정 시간 간격으로 위치 정보를 저장하거나 방향과 속도 등의 변화량을 고려해 위치 정보를 저장하였다. 이는 특정 시간대의 접근 권한에 의해 이동 정보가 고려하지 않아 정확한 위치 정보의 제공이 어렵고 데이터의 양이 기하급수적으로 늘어난다. 본 논문에서는 이러한 문제점을 해결하고자 위상 정보를 이용한 개선 정책과 불확실한 위치 정보에 대한 추정 기법을 제안하였다.

제안한 기법은 위상 정보를 이용한 갱신 정책은 교차로나 클러스터의 위치 정보만을 저장함으로써 데이터의 양을 줄이고 디스크의 반전한 접근을 덜어준다. 따라서 최소한의 정보만으로 정확한 위치를 얻기 위한 응답을 하므로 빠른 응답시간을 보장할 수 있다. 또한 위상 정보를 이용한 불확실한 위치 정보에 대한 추정 기법은 차량 객제의 특성을 고려하여 정확성으로 저장되지 않은 데이터에 대해 사용자 간의 공유를 요청한 결과 보다 정확한 위치 정보를 제공해 줄 수 있다.

본 논문에서의 실험은 시뮬레이션을 이용했기 때문에 향후 실제로 차량이 이동할 경우 전송되는 위치 정보의 오차를 고려해 데이터를 저장할 수 있는 보정 기법이 연구되어야 하며 대량의 시공간 데이터의 효율적으로 처리할 수 있는 인프라 기법이 연구되어야 한다.

참고문헌
[2] A. P. Sisistla "Querying the Uncertain Position of Moving Objects"
[6] 안요한 "차량 위치 추적을 위한 이동 객체 관리 시스템의 설계" D 제1-1번 제5호, 2002년 10월