

# 상호 이동성을 갖는 이동 객체의 연속 범위 질의 처리

## A Continuous Range Query Processing of Reciprocal Moving Objects

최길성, 서상석\*, 복경수\*, 유재수\*  
 동아방송대학 정보통신계열  
 충북대학교 정보통신공학과\*

Choi Kil-Seong, Seo Sang-Seok\*,  
 Bok Kyoung-Soo\*, Yoo Jae-Soo\*  
 Division of Telecommunication, Dong-Ah  
 Broadcasting College  
 Dept of Computer & Communication,  
 Chungbuk National University\*

### 요약

이동 객체의 지속적인 위치 변화로 인해 일정 시간 동안 동일한 질의를 계속적으로 수행하는 연속 질의에 대한 필요성이 증가되고 있다. 이러한 연속 질의는 지속적인 이동 객체의 위치 변화에 따라 서버에서 계속적인 질의 처리를 수행하기 때문에 많은 질의 처리 비용을 소요할 뿐만 아니라 서버 및 네트워크에 많은 부하를 초래할 수 있다. 본 논문에서는 연속 질의의 하나인 연속 범위 질의를 효과적으로 수행하기 위한 질의 처리 기법을 제안한다. 제안하는 질의 처리 기법은 상호 이동 중인 이동 객체에 대한 연속 질의 처리를 수행하기 위해 질의에 대한 응답의 유효성과 질의 결과를 미리 예측한다. 또한, 객체의 동적 변화에 따른 질의 처리를 효과적으로 수행할 수 있도록 한다.

### Abstract

Processing the continuous queries which perform the identical query in given time interval is required because of the continuous change of location of the moving object. Thus the continuous query may cause loads to the server or network and take the cost of processing time because the continuous query is performed in server as the moving object's location changes. In this paper, we propose the query processing technique to perform the continuous range query that is a sort of continuous query effectively. The proposed query processing technique predicts the query result and the validity of query answer to perform the continuous query for reciprocal moving object. And it enables to process effectively the query that moves dynamically.

## I. 서론

최근 무선 네트워크 기술의 발전으로 인해 이동성을 기반으로 하는 서비스들에 관심이 집중되고 있다. 그리고 휴대폰, PDA와 같은 휴대용 단말기의 사용이

일반화된 현재 시점에서 위치 기반 서비스(Location-based Service)나 GPS(Global Positioning System)와 같은 서비스들이 많은 활용이 되고 있다. 이런 서비스들을 효과적으로 제공하기 위해서는 이동 객체의 특징을 고려한 질의 처리 방법이 필요하다. 이동 객체는 시간의 변화에 따라 그 위치가 연속적으로

본 연구는 한국과학재단 목적기초연구(R01-2003-000-10627-0)지원으로 수행되었음

로 변경되는 특성을 가지는 시공간 데이터이다. 지금까지 연구된 이동 객체 관리 기법들은 이동 객체의 과거 및 미래 위치 정보를 개별적으로 관리하고 있다. 이러한 기법은 이동 객체의 위치 정보가 변경되었을 때 모든 위치 정보를 재계산해야 한다. 연속 질의 요청이 들어오면 서버는 주어진 질의 시간동안에 동일한 내용의 질의를 반복적으로 수행하고 각 시간 간격의 질의 결과를 사용자에게 전송한다. 기존의 질의 처리 기법은 매 시간 간격마다 질의를 처리해야 하기 때문에 처리 비용이나 결과 관리 비용에 대한 오버헤드가 있다. 또 질의나 이동객체의 움직임에 변화가 있는 경우 질의를 재처리한다. 질의를 다시 수행하는 것은 질의 처리 시간이나 질의 수행 도중 이동객체의 움직임을 질의에 반영하지 못하는 경우가 발생할 수 있다. 본 논문에서는 상호 이동하는 객체에 대한 연속 범위 질의 처리 기법과 이들 객체를 관리하는 기법을 제시한다. 제안하는 기법은 질의 요청시 질의와 이동객체간의 현재 위치와 미래 위치, 거리를 계산하고 질의 결과를 서버에 저장한다. 서버는 질의나 이동객체의 움직임이 질의 시간동안에 변하지 않는다는 가정을 하고 서버에 저장된 정보를 사용해서 질의 처리를 수행한다. 또한 제안하는 기법은 질의나 이동객체의 움직임에 변화가 생길 경우 질의를 재수행하는 것이 아니라 변화된 시점에서 질의를 수행하는 방법을 제시한다. 본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해서 기술하고, 3장에서는 연속 범위 질의 처리 기법에 대해서 설명한다. 그리고 마지막으로 4장에서 결론을 맺는다.

## II. 관련연구

본 장에서는 연속 질의의 종류와 처리 기법에 대해서 기술한다. 연속 질의는 같은 내용의 질의를 매 시간 간격마다 서버에 요청하는 질의의 한 형태이다. 일반적인 연속 질의는 범위 질의(Range Query)와 k-최근접 질의(kNN Query)를 예로 들 수 있다.

Yufei Tao[1]는 범위 질의의 형태를 질의  $q$ 를 중심으로 하고 검색 반경  $r$ 을 반지름으로 하는 원으로 구성했다. 각 시간 간격마다 이 원에 포함되는 객체를 연속 질의의 결과로 반환한다. Iosif Lazaridis[2]는 시간에 따라 위치가 변하는 모바일 객체에 대한 동적 질의를 위한 질의 처리기법을 제안하였다. 먼저 질의의 형태를 예측가능한 질의(predictive query)와 예측불가능한 질의(non-predictive query)로 분류하였다. PDQ는 질의의 궤적을 알고 있기 때문에 결과를 미리 계산할 수 있고 NPDQ는 PDQ와는 반대로 미리 질의의 궤적을 알 수 없기 때문에 미리 결과를 계산할 수 없는 일반적인 경우이다. NDPQ에 대한 알고리즘은 이전 스냅샷 질의의 결과가 재사용될 수 있기 때문에 많은 디스크 접근을 피할 수 있다. Divyakant Agrawal[3]은 주어진 범위내에 이동객체가 포함되는지를 결정하는 효과적인 필터링 알고리즘을 포함하는 이동객체에 대한 범위 질의 알고리즘을 제안하였다. 이동 객체의 삽입에 영향을 미치는 변수에는 두 가지를 설정했는데 하나는 그리드의 단위이고 다른 하나는 하나의 셀에 유지될 수 있는 이동객체의 수에 의존하여 이동객체가 움직일 수 있는 최대 속도를 반환하는 최대 속도 함수의 임계값이다.

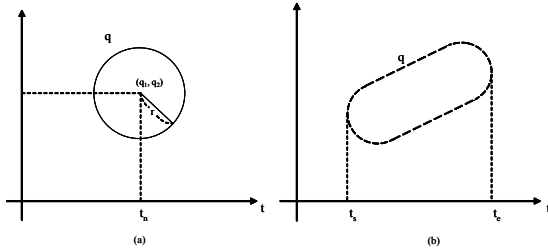
## III. 연속 범위 질의 처리 기법

본 장에서는 각각 자신만의 움직임을 갖는 질의 객체와 검색 대상 객체에 대한 연속 질의 처리 기법에 대해 알아본다. 제시하는 연속 질의 처리 기법에서 질의 객체의 이동 패턴은 일정하다고 가정한다.

### 3.1 범위 질의의 종류

범위 질의의 형태는 그림1로 표현할 수 있다. 기본적인 범위 질의는 (a)와 같은 형태이다. 특정 시간  $t$ 에서 질의 객체  $q$ 는 중심좌표가 ( $q_1, q_2$ )이고 반지름이  $r$ 인 원으로 구성된다.(a)와 같은 질의 형태의 예로 '현재 시각에서 반경 1km 내의 주유소를 검색하라'

를 들 수 있다. (b)는 전체 질의 시간 간격  $[t_s, t_e]$ 에서의 연속 범위 질의이다.



▶▶ 그림 1 범위 질의의 종류

$[t_s, t_e]$  내의 각 시간 간격에서의 범위 질의의 집합 형태이다. 연속 범위 질의는 '10분 동안 반경 1km 내의 주유소를 검색하라'와 같은 형태의 질의를 예로 들 수 있다.

질의 요청은  $(Q, r, t)$ 의 형태로 표현된다. 이는 시간  $t$ 에서 질의 반경  $r$ 에 속하는 모든 객체를 검색하는 질의를 나타낸다. 연속 범위 질의는 질의 객체가 일정 시간 간격 동안에 동일한 내용의 질의를 계속 수행하는 것이다.

질의 요청시 서버로 전송되는 데이터의 형태는  $\langle O_{id}, p_{cur}, v_{cur}, t_{cur} \rangle$ 과 같은 형태로 전송되고 각 데이터의 설명은 다음과 같다.

[표 1] 서버로 전송되는 이동 객체의 데이터 형태

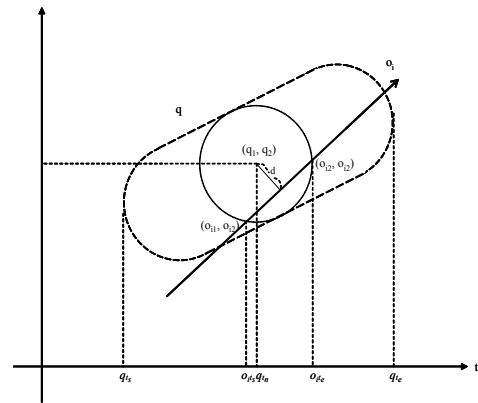
전송 데이터	설명
oid	이동 객체에 대한 식별자
p <sub>cur</sub>	이동 객체의 현재 위치( $p_1, p_2, \dots, p_n$ )
v <sub>cur</sub>	이동 객체의 현재 속도( $v_1, v_2, \dots, v_n$ )
t <sub>cur</sub>	전체 시간 간격( $t_1, t_2, \dots, t_n$ )에서의 현재 시간

미래 시점  $t$ 에서 객체의 위치는 (식1)과 같이 객체의 위치와 속도를 통해 계산한다.

$$p_t = p_{cur} + v_{cur}(t - t_{cur}) \quad \text{식(1)}$$

### 3.2 연속 범위 질의 처리 기법

본 절에서는 연속 범위 질의 처리 기법에 대해 설명한다. 질의의 이동 패턴은 일정한 것으로 가정한다. 사용자의 질의  $q$ 는 시간  $[q_t_s, q_t_e]$  동안 중심 좌표가  $(q_1, q_2)$ 이고 반지름이  $r$ 인 원의 형태로 표현된다. 객체  $o$ 의  $t$ 시점에서의 위치  $p_t$ 는 좌표  $(p_1, p_2)$ 로 표현된다. 그림2는 현재 시간  $t$ 에서의 질의와 객체간의 거리를 나타낸 것이다.



▶▶ 그림 2 질의와 객체의 거리

두 점간의 거리는 식(2)를 이용해서 구할 수 있다. 두 점을 각각  $(a, b)$ 와  $(x, y)$ 로 가정했을 경우 두 점간의 거리는

$$\sqrt{(a-x)^2 + (b-y)^2} \quad \text{식(2)}$$

와 같다.

$t$ 시점에서의 질의와 이동객체의 위치를  $(q_1(t), q_2(t))$ 와  $(p_1(t), p_2(t))$ 라 했을 때 질의의 위치는

$$\begin{aligned} q_1(t) &= q_1 + v_{q_1}(t - t_{cur}) \\ q_2(t) &= q_2 + v_{q_2}(t - t_{cur}) \end{aligned}$$

이고 이동객체의 위치는

$$p_1(t) = p_1 + v_{p_1}(t - t_{cur})$$

$$p_2(t) = p_2 + v_{p_2}(t - t_{cur})$$

이다. 질의와 이동객체간의 거리를 구하기 위해 식(2)에 대입한 후 이 식을 시간 t에 대해서 정리하면 식(3)과 같다. 서버에 질의 요청이 들어오면 먼저 질의와 객체간의 거리(d)를 식(3)을 이용하여 계산한다.

$$d = At^2 + Bt + C (\leq r^2) \quad \text{식(3)}$$

식(3)에서 A, B, C의 값은 식(4)과 같다.

$$A = (v_{q_1} - v_{p_1})^2 + (v_{q_2} - v_{p_2})^2$$

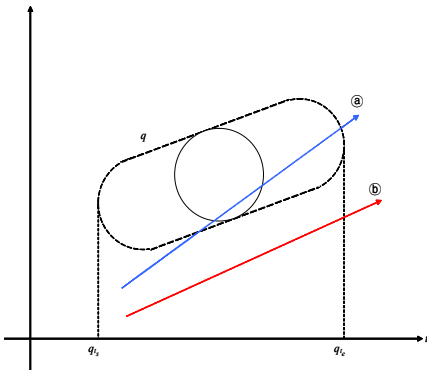
$$B = 2\{(q_1 - p_1)(v_{q_1} - v_{p_1}) + (q_2 - p_2)(v_{q_2} - v_{p_2})\}$$

$$C = (q_1 - p_1)^2 + (q_2 - p_2)^2 \quad \text{식(4)}$$

식(4)에서 (q<sub>1</sub>, q<sub>2</sub>), (p<sub>1</sub>, p<sub>2</sub>)는 각각 질의 q와 객체 o의 시간 t에서의 좌표이고, (v<sub>q1</sub>, v<sub>q2</sub>)는 질의 q의 x, y 축의 속도, (v<sub>p1</sub>, v<sub>p2</sub>)는 객체 o의 x, y 축의 속도이다.

또한 식(3)을 이용하여 객체 o가 질의 q에 포함되는지를 판별할 수 있는데 이는 식(5)를 이용해 알 수 있다.

$$B^2 - 4AC \quad \text{식(5)}$$

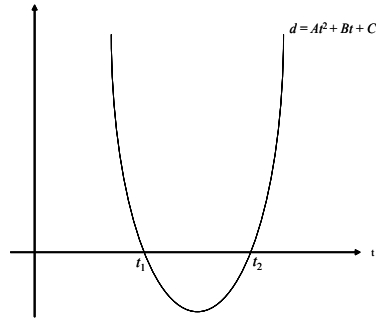


▶▶ 그림 3 객체의 질의 포함 여부 판별

식(5)의 값이 0보다 작은 값이 나오는 객체는 그림 3의 (b)의 경우이며 이 객체는 질의 처리 과정에서 제외시킨다. 0보다 크거나 같은 값이 나오는 객체는 그림3의 (a)이고 이 경우 식(3)의 해를 구한다. 이 t값이 객체 o의 전체 시간 [t<sub>s</sub>, t<sub>e</sub>]값을 나타낸다. 해를 t<sub>1</sub>, t<sub>2</sub>라 하면 식(6)과 그림4로 나타낼 수 있다.

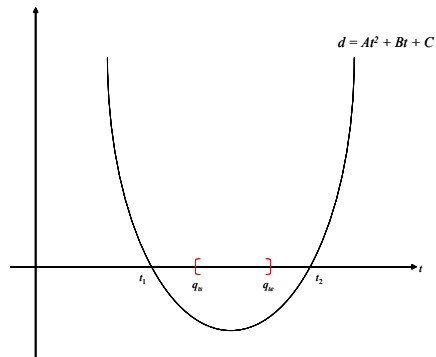
$$t_1 = \frac{-B - \sqrt{B^2 - 4AC}}{2A}$$

$$t_2 = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad \text{식(6)}$$



▶▶ 그림 4 객체 o의 전체 시간 t1과 t2

그림 5는 실제 질의 결과 시간을 나타낸 것으로 객체 o는 시간 [q<sub>ts</sub>, q<sub>te</sub>]에 질의의 결과가 된다.

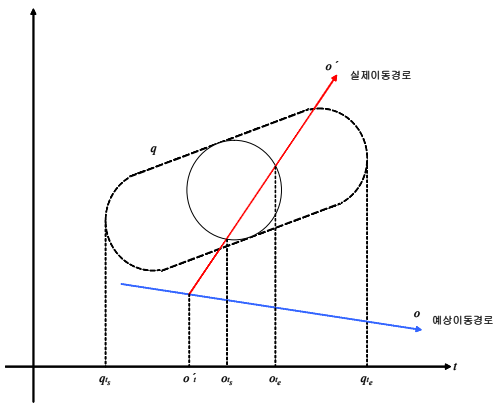


▶▶ 그림 5 질의 결과 시간 qts와 qte

### 3.3 이동객체의 움직임 변화시 질의 처리

#### 3.3.1 이동객체의 삽입

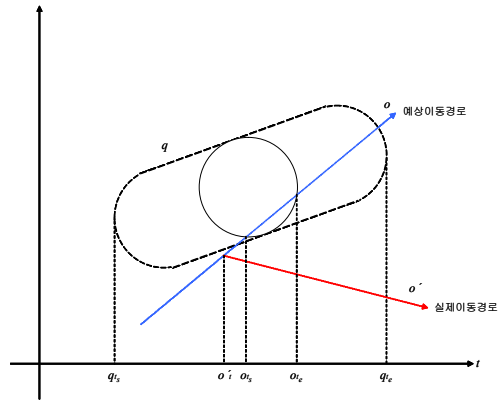
그림6은 질의 시간  $[q_{ts}, q_{te}]$  동안 객체  $o$ 의 움직임에 변화가 발생한 경우 중 삽입을 나타낸 것이다. 그림7에서 이동객체는 예상이동경로를 벗어나 실제이동경로  $o'$ 으로 이동했다. 예상이동경로로 이동했을 경우 이 이동객체는 식(5)를 이용하여 질의의 결과가 되지 않지만 이동객체의 움직임이 변화된 시점  $o't$ 에서 다시 식(5)를 이용해 질의의 결과가 되는지 판별했을 때 경로를 바꾼 객체  $o'$ 은 앞으로 질의  $q$ 의 결과가 된다. 따라서 움직임에 변화가 일어난 시점  $o't$ 에서 이동객체  $o'$ 에 대한 질의를 수행한다.



▶▶ 그림 6 객체의 이동 변화(삽입)

#### 3.3.2 이동 객체의 삭제

그림7은 질의 시간  $[q_{ts}, q_{te}]$  동안 객체  $o$ 의 움직임에 변화가 발생한 경우 중 삭제를 나타낸 것이다. 삽입의 경우와 마찬가지로 식(5)를 이용하여 움직임에 변화가 일어난 시점  $o't$ 에서 질의의 결과에 영향을 미치는지 판별한다. 그림8에서는 경로를 바꾼 객체  $o'$ 은 앞으로 질의  $q$ 에 대한 결과가 되지 않기 때문에 질의 처리 과정에서 제외된다.



▶▶ 그림 7 객체의 이동 변화(삭제)

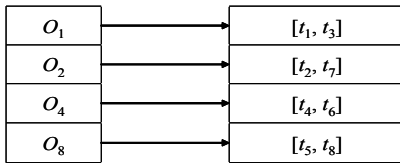
### 3.4 이동객체의 관리

이동객체는 서버에 그림8과 같이 해시 테이블에 저장된다. 이동객체를 해시 테이블에 저장할 때는 시간 단위로 저장하지 않고 이동객체별로 저장을 한다. 시간 단위로 저장을 할 경우에는 이동객체의 움직임에 변화가 생겼을 때 해당 시간 간격 내에 있는 모든 이동객체가 질의의 결과에 영향을 미치는 지를 판별해야 한다. 이동객체 단위로 저장을 하면 이동객체의 움직임이 변한 해당 객체의 시간 간격을 재계산해서 해시 테이블을 수정하면 된다.

$O_1$	→	$[t_1, t_3]$
$O_2$	→	$[t_2, t_7]$
$O_5$	→	$[t_6, t_8]$

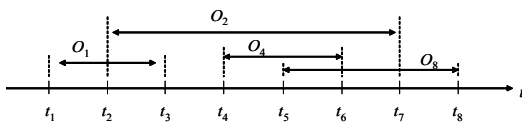
▶▶ 그림 8 객체 관리 해시 테이블

또한 삽입이나 삭제된 이동객체에 대해서도 해시 테이블 내의 해당 객체의 값을 수정함으로써 해결할 수 있다. 그림9는 객체  $O_5$ 가 삭제되고  $O_8$ 이 삽입됐을 경우 수정된 해시 테이블이다.



▶▶ 그림 9 수정된 객체 관리 해시 테이블

사용자에게 결과를 전송할 때는 그림10과 같이 각 시간 간격에 대한 객체를 전송한다.



▶▶ 그림 10 각 시간 간격에서의 객체

#### IV. 결론 및 향후 연구

본 논문에서는 질의 객체와 검색 대상 객체 모두 이동하는 경우에 대한 연속 질의 처리 기법을 제시하였다. 그러나 질의 객체의 이동성은 변화가 없다는 제한을 두었다. 즉 질의 객체는 질의 시간 동안에 방향을 바꾸지 않는다. 향후 연구에서는 질의 객체 역시 방향을 바꾸었을 때 질의를 처리하는 기법에 대한 연구에 초점을 맞출 것이다.

#### ■ 참고문헌 ■

- [1] Yufei Tao, Dimitris Papadias, Qiongmao Shen, "Continuous Nearest Neighbor Search", Proceedings of the 28th VLDB Conference, 2002.
- [2] Iosif Lazaridis, Kriengkrai Porkaew, Sharad Mehrotra, "Dynamic Queries over Mobile Objects". EDBT 2002, pp.269-286
- [3] Hae Don Chon, Divyakant Agrawal, Amr El Abbadi, "Query Processing for Moving Objects with Space-Time Grid Storage Grid", Third International Conference on Mobile Data Management, 2002.