

이동 객체 데이터베이스 상에서 정보 비용 테스트를 위한 테스트 베드의 설계 및 구현

김태원⁰ 김경식 김창화 강태원
강릉대학교 컴퓨터공학과

ktw2000⁰@hanmail.net, (problem, kch, twkang)⁰@kangnung.ac.kr

Design and Implementation of Testbed for Information Cost Test

Tae-Won Kim, Kyung-Sik Kim, Chang-Hwa Kim, Tae-Won Kang
Dept. of Computer Science & Engineering, Kangnung National University

요 약

실시간으로 움직이는 많은 객체들에 대한 index를 데이터베이스에 저장하기란 막대한 비용을 초래 하게 된다[4]. 따라서 데이터베이스 서버에 움직이는 객체에 대한 정보량을 줄이는 것이 필수적이며, 그러기 위한 클라이언트 쪽에서의 이벤트 발생을 줄여 주고 데이터베이스에서의 update 비용을 줄여 주어야 한다. 그러므로 실시간으로 움직이는 객체들에 대한 이벤트 발생 시점을 계산하기 위한 deviation(threshold)를 정하는 방법을 개발하고 이벤트 발생 횟수를 줄임으로써 데이터베이스의 update 비용을 줄일 수 있다. 따라서 이 논문에서는 기존의 기법과 보다 나은 기법을 비교하여 우위를 증명하기 위한 Testbed 구현에 대해 설명한다.

1. 서 론

지구상에 실시간으로 움직이는 객체(moving object) 들은 수없이 많이 있다. 특정 한 객체에 대한 움직임을 파악할 수도 있고 불특정 다수의 움직임을 파악할 수도 있다. 하지만 이런 데이터들을 모두 데이터베이스에 저장하는 것은 불가능하다. 따라서 이런 움직이는 객체들에 대한 효율적인 저장 기법이 필요하다. MOD(moving object databases)는 이런 움직이는 객체들을 데이터베이스에 저장하는 연구로써 1990년 후반부터 활발하게 진행되어 왔다[4]. MOD의 목적은 데이터베이스의 indexing 처리에 들어가는 비용을 최소화 하여 효율적이고 정확한 질의 처리를 할 수 있게 하는 것이다. 그러기 위해서는 데이터베이스 기록에 필요한 정보 비용(information cost)이 있어야 한다. 정보 비용에는 3가지가 있다[1]. Deviation cost, Update cost, Uncertainty cost가 있고 이 3개를 이용해 움직이는 객체의 위치정보를 데이터베이스에 기록한다. 따라서 이런 비용을 감소 시키고 보다 정확한 질의를 할 수 있는 방법이 필요하다. 이동하는 객체를 클라이언트라고 했을 때 클라이언트에서 이벤트를 발생 시키고 통신에 의해 데이터베이스로 이동하는 물체의 위치를 저장하게 하게 된다. 따라서 이벤트 횟수를 줄여서 정보 비용을 줄여주는 요소인 3가지 비용에 대한 효율적인 기법 개발이 필요하다. 각각의 기법은 다른 2가지 비용에도 영향을 주게 되고 이 논문에서는 3가지 비용에서 즉 정보비용에 대해 검증할 수 있고 테스트 할 수 있는 Testbed 구현을 목적으로 한다 2절에서는 Cost의 평

Testbed 구현을 목적으로 한다 2절에서는 Cost의 평가 방법에 대해서 설명하고 3절에서는 testbed가 갖추어야 할 기능에 대해서 설명 한다. 4절에서는 testbed의 구조에 대해 설명하고 5절에서는 각 구성요소의 기능에 대해 설명한다. 6절에서는 testbed의 구현에 대해 설명한다. 마지막 7절에는 결론 및 향후 연구 과제에 대해 설명한다.

2. Cost 평가 방법

데이터베이스에서 indexing 처리를 하기 위한 3가지 비용은 중 Update cost는 이벤트 발생시 MOD에 객체의 위치정보 메시지를 보내는 비용이고, Uncertainty cost는 객체의 단위 시간당 객체의 위치정보 불확실성을 나타내는 방법으로 사용한다. 이 세가지 방법은 서로 유기적으로 작동한다. deviation cost의 평가 방법은 시간 (t1, t2)동안에 이동하는 물체에 대한 deviation(threshold)를 두어 deviation을 벗어 남고 동시에 이벤트를 발생시켰다. 식은 다음과 같다[1][2][3].

$$COST_d(t_1, t_2) = \int_{t_1}^{t_2} d(t)dt$$

여기서 d(t)는 deviation 함수를 말한다. 기존의 deviation 함수보다 더 효율적인 함수를 만드는 것이 목적이므로 새로운 함수를 적용 시켜 기존의 함수와 비교 이벤트 발생 횟수를 계산하고, 적은 이벤트 발생과

나머지 정보 비용 요소인 Update Cost, Uncertainty Cost 등 다른 정보 비용 요소에도 영향을 주어서 데이터 베이스 질의에 대한 정확한 응답을 나타내야 한다. 따라서 평가 방법과 비교 할 수 있는 Test 기능을 갖추고 있는 Testbed를 만들어야 한다.

3. Testbed가 갖추어야 할 기능

움직이는 객체의 실시간 정보를 가지고 있어야 하고 미래 시점에 대한 예측을 가능하게 해야 한다. 따라서 정보 비용 계산을 위한 정보들을 포함하고 있어야 한다. 그러기 위해서는 시간 값과 좌표 값 물체의 이동 방향에 대한 정보를 위지 시켜주어야 한다.

- (1) x, y 평면에 움직이는 객체를 표현 할 수 있는 그래픽 기능 제공
- (2) x, y 평면 대한 시간을 추가 하여 시간 t 대한 x 그래프와, 시간 t 대한 y 그래프 기능 제공
- (3) deviation cost(threshold) 변경가능 하게 해주고 적당한 deviation cost를 찾을 수 있게 해준다.
- (4) 단위 시간당 거리를 유동적으로 변경 가능하게 해준다.
- (5) 새로운 기법에 대한 기능을 프로그램 모듈로 첨가 사용할 수 있으며 여러 가지 기법들을 비교 할 수 있게 한다.

4. Testbed 구조

Testbed구조는 4개의 폼과 하나의 모듈을 첨가 하고 있다. 각각의 폼은 움직이는 마우스 포인터를 받아 각각의 정보를 표현해 준다. Form1에서 계산되어진 moving object의 위치변환에 정보를 시간 t를 첨가해 두 개의 좌표로 표현해준다. 모듈은 이런 좌표 값 정보와 각각의 연산을 수행한다.

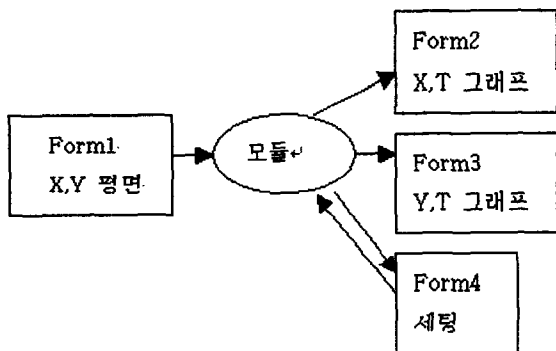


그림 1, 프로그램 구조

5. 각 구성요소의 기능

x, y 평면에 대한 움직이는 객체를 픽셀과 선으로 표현한다. 픽셀과 픽셀의 간격은 거리 단위가 되고 속도는

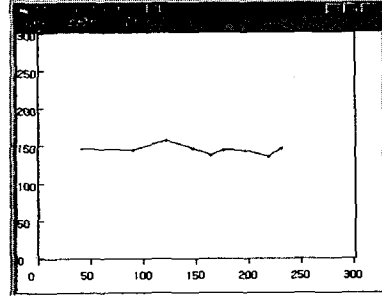


그림 2, moving object의 x, y좌표

- (1) 시간 t에 대한 각각의 x, y 그래프를 표현 한다. 시간은 초기 값이 주어지나 움직이는 객체의 성격에 맞게 조정할 수 있다.

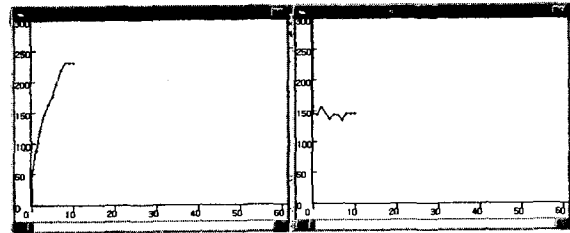


그림 3. 시간 t에 대한 x, y 그래프
(y축은 거리, x축은 시간을 나타낸다.)

- (3) 시간, 눈금간격(거리), 가중치를 설정할 수 있다. 여기서 가중치란 물체의 deviation(threshold)값으로써 물체가 시간 t동안 실제 포인터가 위치해 있어야 하는 거리를 나타내는 것이다. 적당한 가중치 설정은 정 moving object의 정보비용을 줄이는데 많은 영향을 미친다. 따라서 testbed를 이용한 적절한 가중치 설정도 중요하다.

그림 4, 기본 설정

여기서 시간은 초기 설정은 1초로 되어있으며 1/1000초로 계산되어 진다. 가중치는 deviation cost의 값으로써 적합한 threshold값을 찾는 것이다. 여기서 주어진 Threshold 값에 따라서 cost의 비용이 많이 달라 질것이다. 눈금 간격은 객체가 움직이는 영역을 조절할 수 있다. 즉 거리를 나타낸다. 영역은 픽셀 단위 300 * 300, 600 * 600 모두가 주어지며, 각각의 영역에 대해서 눈금 간격을 조절 해서 움직이는 객체의 거리를 알아 볼 수 있다.

6. Testbed 구현

```

Private Sub form_1()
{
    x_point = i

    y_point = j

    time_interval = t
    call function(x_point, y_point, time_interval)
    call form2(x_point, time_interval)
    call form3(y_point, time_interval)
}

Private Sub form_2(x_point, time_interval)
{
    //시간에 따른 x축의 위치정보 표현
}

Private Sub form_3(y_point, time_interval)
{
    // 시간에 따른 y축의 위치정보 표현
}

Public Module()
{
    function1() // 선을 그린다.
    function2() // 점을 그린다.
    function3()// 기존 기법
    function4()//향상된 기법
}
    
```

7.결론 및 향후 연구 과제

실시간으로 움직이는 객체들에 대한 MOD를 처리하기 위한 다양한 기법들이 있다. 그 중에서 이동하는 물체 즉 자동차, 비행기, 배 등에 대한 위치 정보를 Update 해주는데 있어서, 데이터베이스에 정보 비용과 통신 비용이 들어 간다. 이런 비용들은 물체의 위치정보를 저장하는데 있어 줄여 나가야 할 부분이다. 이런 문제들을 해결하고자 본 논문에서는 활용하고자 하는 기법에 우선하여 기존의 기법들을 테스트 할 수 있는 Testbed를 만들어야 했다. 본 논문의 Testbed를 이용 물체의 움직임에서 시간을 추가하여 시간에 대한 x, y에 대해 알아 보았다. 따라서 이런 움직임들에 대한 정보에서 비용 감소를 위한 정책을 앞으로 연구하여야 하며, 많은 요소 중에서 물체가 이동하면서 발생하는 이벤트에 대한 발생 수를 줄이고 질의에 의한 데이터베이스에서의 정보 추출이 향상되어야 함을 알았다. 정보 비용을 줄이기 위해서는 deviation cost에 대한 향상된 기법을 연구 해야 하고 Testbed를 이용 하여 기법의 성능을 분석해야 할 것이다.

참고문헌

[1] "Moving Object Databases : Issues and Solutions" the Proceedings of the tenth International Conference on Scientific and Statistical Database Management (SSDBM98), Capri(Italy), July 1-3, 1998

[2] "Updating and Querying Databases that Track Mobile Units" invited paper, special issue of the distributed and Parallel Databases Journal on Mobile Data Management and Applications, 1999, Kluwer Academic Publishers.

[3]"Cost and Imprecision in Modeling the Position of Moving Objects" Proceeding of the 14-th International Conf. On Data Engineering, Feb 1998 Orlando, FL

[4]"Indexing moving point" Written with Pankaj K, Agarwal and Lars Arge, Proceedings of the 19th ACM Symposium on Principles of Database Systems, 175-186, 2000.