

유비쿼터스 센서 네트워크에서 불확실한 데이터의 효율적 검색을 위한 그룹화 기법† A Grouping Technique for Efficient Search of Uncertain Data in Ubiquitous Sensor Networks

김동오*, 홍동숙, 한기준
Dong-Oh Kim*, Dong-Suk Hong, Ki-Joon Han
건국대학교 컴퓨터·정보통신공학과
{dokim*, dshong, kjhan}@db.konkuk.ac.kr

요 약

유비쿼터스 센서 네트워크 관련 기술이 급속히 발전하고 센서가 여러 분야에서 활용됨에 따라 센싱되는 데이터를 효율적으로 관리하기 위한 연구가 활발히 진행되고 있다. 일반적으로 유비쿼터스 센서 네트워크에서 센싱되는 데이터는 계통적 오차와 측정 방법의 부적합 등으로 인한 불확실성을 가지며, 또한 센싱하는 환경 및 제약으로 인해 데이터가 유사하거나 반복되는 제한성을 가진다. 그러나 기존의 연구들은 센싱되는 데이터의 이러한 특성을 고려하지 못함으로 인해 검색이 비효율적이다.

따라서, 본 논문에서는 불확실한 데이터를 고려한 기존의 인덱스를 확장하여 유비쿼터스 센서 네트워크에서 센싱되는 데이터 중 제한성을 가지는 데이터를 그룹화함으로써 효율적인 검색을 지원하는 그룹화 기법을 제시한다. 본 논문은 센싱된 데이터를 그룹화하는 기법으로써 처음에 그룹으로 설정된 영역을 이용해 그룹화하는 최초 그룹화 기법, 한 그룹 내에 최대한 많은 데이터를 그룹화하는 최적 그룹화 기법, 센싱된 데이터를 최대한 근접하게 그룹화하는 최근접 그룹화 기법을 제시한다. 마지막으로, 성능 평가를 통해 본 논문에서 제시한 그룹화 기법을 이용한 인덱스에 대한 검색 성능의 우수성을 입증한다.

1. 서 론

최근 온도 센서, RFID, GPS 등과 같은 다양한 데이터를 센싱하는 센서 기술과 CDMA, WiFi, WiBro 등과 같은 무선 통신 기술 등이 발전함에 따라 환경 모니터링, 차량 도난 감지 등과 같은 유비쿼터스 센서 네트워크(USN) 관련 기술에 대한 관심이 증대되고 있다[1]. 또한 이러한 환경에서 센싱되는 데이터를 효율적으로 관리하기 위한 연구도 활발히 진행되고 있다

[2,3,4].

일반적으로 유비쿼터스 센서 네트워크에서 센싱되는 데이터는 검색의 효율성을 위해 중앙 서버(또는 외부 서버)에 저장하는데 이 때 비용 감소를 위한 갱신 지연과 같은 계통적 오차나 측정 방법의 부적합 등으로 인한 불확실성과 센싱하는 환경 및 제약으로 인해 데이터가 유사하거나 반복되는 제한성을 가진다. 그러나 기존의 연구들은 센싱되는 데이터의 이러한

†본 연구는 서울시 산학연 협력사업의 신기술 연구개발 지원사업의 지원으로 수행되었음.

특성을 모두 고려하지 못함으로 인해 검색이 비효율적이다[1,5].

따라서, 본 논문에서는 유비쿼터스 센서 네트워크에서 센싱되는 데이터 중 제한성을 가지는 데이터를 그룹화하기 위한 그룹화 기법을 제시하고 이를 적용하여 불확실한 데이터를 고려한 기존의 인덱스를 확장하여 효율적인 검색을 지원하기 위한 그룹 불확실성 인덱스(Grouping Uncertainty R-tree: GUR-tree)를 제시한다. 그룹화란 특정 영역 내에 존재하는 여러 개의 데이터를 하나의 그룹으로 관리하는 것을 의미한다.

본 논문은 센싱된 데이터를 그룹화하는 기법으로써 처음에 그룹으로 설정된 영역을 이용해 그룹화하는 최초 그룹화 기법, 한 그룹 내에 최대한 많은 데이터를 그룹화하는 최적 그룹화 기법, 센싱된 데이터를 최대한 근접하게 그룹화하는 최근접 그룹화 기법을 제시한다. 마지막으로, 성능 평가를 통해 본 논문에서 제시한 그룹화 기법을 이용한 인덱스에 대한 검색 성능의 우수성을 입증한다.

본 논문의 구성은 다음과 같다. 제 2 장에서는 본 논문에서 확장한 불확실한 데이터를 고려한 인덱스인 UR-tree에 대해 설명한다. 제 3 장에서는 본 논문에서 제시한 그룹화 기법과 그룹화 기법을 적용한 GUR-tree의 인덱스 구조에 대해서 설명하고, 제 4 장에서는 실험을 통해

GUR-tree의 성능을 분석한다. 마지막으로, 제 5 장에서는 결론에 대해 언급한다.

2. UR-tree

UR-tree(Uncertainty R-tree)[6]는 다양한 센서에서 센싱된 데이터의 불확실성을 고려하여 인덱스의 갱신 성능을 개선한 불확실한 데이터를 위한 인덱스이다. UR-tree의 인덱스 구조는 R-tree와 유사하나 검색 시에 불확실성 영역을 이용하는 점과 갱신 시에 갱신을 지연시키기 위한 갱신 영역을 이용하는 점에서 구분된다.

UR-tree에서 갱신 요청은 센서 노드의 아이디와 새로 센싱된 데이터로 구성된다. 여기서, 해당 아이디를 가지는 리프 노드에 빠르게 접근하기 위한 방법으로 보조 인덱스를 사용한다. 보조 인덱스의 각 노드는 아이디와 리프 노드 포인터로 구성되는데, Hash나 B-tree로 구현될 수 있다. 그림 1은 UR-tree와 보조 인덱스를 구성한 예를 보여준다.

유비쿼터스 센서 네트워크를 구성하는 각 센서 노드 N 은 센서 노드를 식별할 수 있는 아이디와 센싱을 담당하는 센서 R 과 O 를 가진다고 가정하자. 이때, 아이디가 i 인 센서 노드를 N_i 라고 하고, 센서 노드 N_i 의 센서 R 과 O 에서 센싱된 데이터는 $(N_{i,r}, N_{i,o})$ 이며, 센서 R 과 O 의 불확실성 영역은 각각 $(-U_r, U_r)$, $(-U_o, U_o)$ 이라고 하

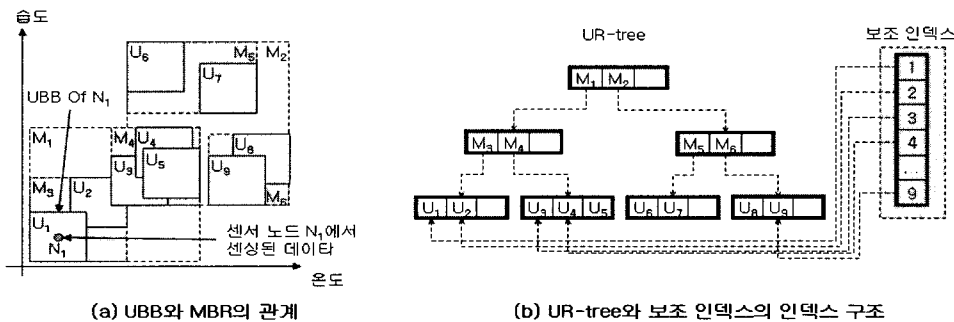


그림 1. UR-tree와 보조 인덱스의 구성 예

자. 이때, 센서 노드 N_i 의 불확실성 영역을 위한 인덱스의 엔트리는 $(N_i, (N_{i,r}-U_r, N_{i,o}-U_o), (N_{i,r}+U_r, N_{i,o}+U_o))$ 이다. 즉, 센싱된 데이터는 2차원 점 $(N_{i,r}, N_{i,o})$ 으로 표현되며, 해당 데이터의 불확실성 영역은 2차원 사각형 $((N_{i,r}-U_r, N_{i,o}-U_o), (N_{i,r}+U_r, N_{i,o}+U_o))$ 으로 표현된다. 이때, 센싱된 데이터에 대한 불확실성 영역을 Uncertainty Bounding Box(UBB)라고 한다.

그림 1(a)는 센서 노드 N_i 의 UBB인 U_i 상호간, 트리 노드의 MBR인 M_j 상호간 또는 U_i 와 M_j 간의 contain 및 overlap 관계를 보여준다. 그림 1(b)는 그림 1(a)를 인덱스로 표현한 그림으로써, UR-tree의 리프 노드는 각 센서 노드의 UBB로 구성되며, 보조 인덱스는 해당 노드 아이디의 리프 노드를 가리키고 있다.

3. GUR-tree

본 장에서는 불확실한 데이터를 고려한 인덱스인 UR-tree에 그룹화 기법을 적용함으로써 효율적인 검색을 지원하기 위한 그룹 불확실성 인덱스인 GUR-tree를 설명하고 이때 사용한 그룹화 기법을 설명한다.

유비쿼터스 센서네트워크에서 센싱되는 데이터는 센싱하는 환경 및 제약 등으로 인해 데이터가 유사하거나 반복되는 제한성을 가진다. 이러한 성질은 특정 조건의

센서 노드에서 센싱된 데이터는 그 값이 유사하다는 그룹성을 포함한다. 즉, 유사한 환경에서 다수의 센서 노드를 사용하는 유비쿼터스 센서 네트워크의 경우 센서 노드들에서 센싱되는 데이터 값들이 유사하고, 그 변동 패턴도 유사할 확률이 높다. 따라서, 이러한 데이터를 그룹(즉, 센서 노드 그룹)으로 관리함으로써 검색의 효율성을 높일 수 있게 된다.

아이디가 j 인 센서 노드 그룹을 G_j , G_j 에 속하는 아이디가 i 인 노드를 $G_j.N_i$ 라고 하자. 이때, 센서 노드 $G_j.N_i$ 의 센서 R과 O에서 센싱된 데이터는 $(G_j.N_{i,r}, G_j.N_{i,o})$ 이다. 그리고 센서 노드 N_i 의 불확실성 영역을 위한 인덱스의 엔트리는 $(G_j, (MIN(G_j.N_{i,r})-U_r, MIN(G_j.N_{i,o})-U_o), (MAX(G_j.N_{i,r})+U_r, MAX(G_j.N_{i,o})+U_o))$ 이다. 여기서 이러한 그룹 영역을 Grouping Uncertainty Bounding Box(GUBB)라고 한다.

GUR-tree의 인덱스 구조는 UR-tree와 유사하나 검색 및 갱신 시에 그룹 영역을 이용하는 점에서 구분된다. GUR-tree에서 갱신 요청은 센서 노드의 아이디와 새로 센싱된 데이터로 구성된다. 여기서, 해당 아이디를 가지는 리프 노드에 빠르게 접근하기 위한 방법으로 보조 인덱스를 사용한다. 그림 2는 GUR-tree와 보조 인덱스를 구성한 예를 보여준다.

그림 2(a)는 센서 노드 N_i 의 UBB인 U_i ,

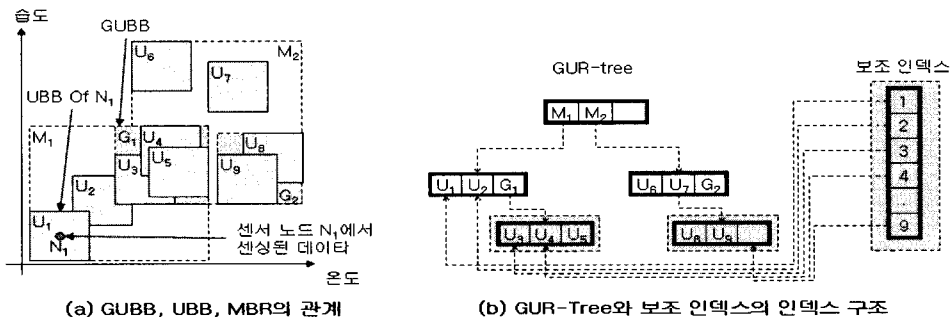


그림 2. UR-tree와 보조 인덱스의 구성 예

트리 노드의 MBR인 M_i , GUBB인 G_i 들 간의 관계를 보여준다. 그림 2(b)는 그림 2(a)를 인덱스로 표현한 그림으로써, GUR-tree의 리프 노드는 각 센서 노드의 UBB와 GUBB로 구성되며, 보조 인덱스는 해당 노드 아이디의 리프 노드를 가리키고 있다.

GUR-tree에서 그룹화 기법에서 그룹 영역을 설정하는 방법에 따라 최초 그룹화 기법, 최적 그룹화 기법, 최근접 그룹화 기법이 있다.

그림 3은 GUR-tree의 최초 그룹화 기법을 보여준다.

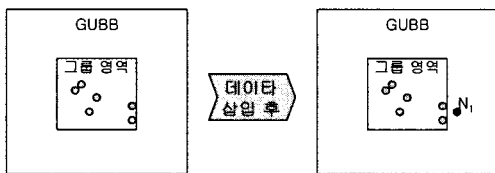


그림 3. 최초 그룹화 기법

그림 3에서 보듯이 센서 노드 N_1 의 데이터가 입력되었을 때 최초 그룹화 기법에서는 처음 설정된 그룹 영역 내에 데이터가 들어오지 않아서 해당 그룹으로 설정되지 않는 모습이다. 즉, 최초 그룹화 기법은 처음 설정된 GUBB는 그룹에 속한 객체가 한 개가 될 때까지는 변경되지 않는다. 최초 그룹화 기법은 그룹 영역을 설정 시 적은 시간이 들지만 그룹 영역이 최적화되기 힘들다는 단점이 있다.

그림 4는 GUR-tree의 최적 그룹화 기법을 보여준다.

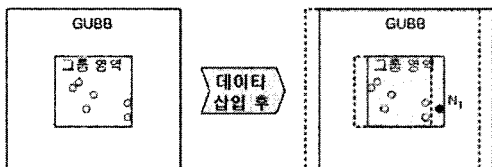


그림 4. 최적 그룹화 기법

그림 4에서 보듯이 센서 노드 N_1 의 데이터가 입력되었을 때 최적 그룹화 기법

에서는 그룹 영역에 최대한 많은 데이터가 올 수 있도록, 즉 하나의 그룹에 최대한 많은 데이터가 그룹화되도록 그룹 영역을 재설정한다. 또한, 그룹 영역이 재설정됨에 따라 GUBB도 재설정된다. 최적 그룹화 기법은 그룹 영역 설정 시 최초 그룹화 기법보다 더 많은 시간이 들지만, 그룹 영역이 최적화되는 장점이 있다.

그림 5는 GUR-tree의 최근접 그룹화 기법을 보여준다.

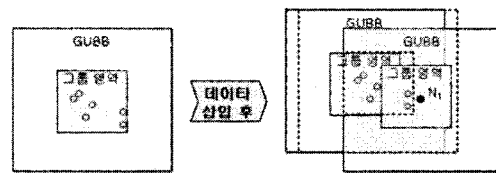


그림 5. 최근접 그룹화 기법

그림 5에서 보듯이 센서 노드 N_1 의 데이터가 입력되었을 때 최근접 그룹화 기법에서는 그룹 영역이 최대한 밀집되도록 설정한다. 즉, 그룹의 밀집도가 높아지도록 필요에 따라서는 노드의 분할이 발생할 수 있다. 최근접 그룹화 기법은 그룹 영역 설정 시 노드 분할 등으로 인해 많은 그룹화에 시간이 많이 소요되고, 그룹 영역이 최적화 되지 못하는 단점이 있으나, 검색 시 빠른 결과를 얻을 수 있다.

4. 성능 평가

본 장에서는 GUR-tree에서 갱신 영역에 따른 검색의 정확성을 비교하여 갱신 및 검색 질의의 성능을 분석한다. 본 논문에서 성능 평가에 사용한 데이터는 표1과 같다.

표 1. 성능 평가 데이터

이름	센서 노드 개수	분포 함수	시작 데이터 범위	평균 이동 범위	최소/최대 이동 범위	전체 데이터 범위	센서 개수
DataSet1	500	Gaussian	10,20	0.2	-0.5,0.5	10,20	2
DataSet2	500	Gaussian	15,20	0.2	-0.5,0.5	10,20	2

본 논문에서 성능 평가는 갱신 지연을 불확실성의 원인으로 보고, 임의의 시간 T_i 에 해당하는 데이터를 불확실한 데이터로 설정하여 인덱스를 구성하였으며, T_{i+1} 에 해당하는 데이터를 확실한 실제 데이터로 간주하여 비교하였다. T_i 의 데이터에서 검색한 결과의 개수는 R_i , T_{i+1} 의 데이터에서 검색한 결과의 개수는 R_{i+1} , T_i 와 T_{i+1} 의 검색 결과에 공통으로 나타난 결과의 개수는 R_e 라고 할 때, 다음 두 가지는 검색의 정확성을 측정하기 위한 인자이다.

$$\text{Omission error} = (R_{i+1} - R_e) / R_{i+1}$$

$$\text{Commission error} = (R_i - R_e) / R_i$$

Omission error는 실제 검색 결과에 나타나지만 인덱스 검색 결과에는 나타나지 않을 확률로써 이 확률이 낮을수록 정확도는 증가한다. Commission error는 인덱스 검색 결과에 나타나지만 실제 검색 결과에 나타나지 않을 확률로써 이 확률이 낮을수록 정확도는 증가한다.

예를 들어, 시간 T_5 에서 질의 윈도우 q 로 검색한 결과, 즉 인덱스에서 검색한 결

과 아이디어가 {3,5,8,9,11,15}이고, T_6 에서 질의 윈도우 q 로 검색한 결과, 즉 각 센서 노드에서 검색한 결과 아이디어가 {5,8,12}이라고 하면, R_5 , R_6 , R_e 는 각각 6, 3, 2가 된다. 따라서 Omission error는 $(3-2)/3=0.33$ 이고, Commission error는 $(6-2)/6=0.67$ 이다. Omission error가 0.33이므로 검색 결과에 있어야 할 아이디어가 검색이 안되고, Commission error가 0.67이므로 검색 결과에 없어야 할 아이디어가 검색이 됨으로써 검색의 정확도가 감소함을 알 수 있다.

그림 6은 불확실성 영역과 그룹 영역을 변화시키며 GUR-tree의 성능을 실험한 결과를 보여준다. 그림 6(a)~그림 6(d)는 불확실성 영역을 0.0에서 0.5까지 변화시키며 R-tree, UR-tree, GUR-tree의 성능을 실험한 결과를 보여준다. 본 논문에서 UR-tree<0.5>는 갱신 영역이 0.5인 UR-tree를 나타내고, UR-tree(최초)는 최초 그룹화 기법을 나타내며, GUR-tree(최적)는 최적 그룹화 기법을 나타내고, GUR-tree(최근점)는 최근점 그룹화 기법을 나타낸다.

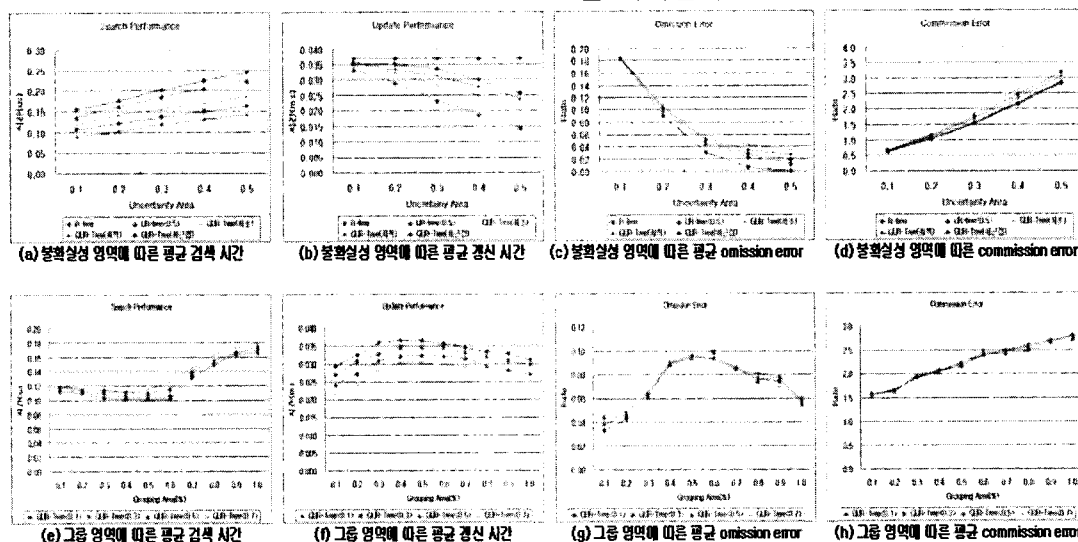


그림 6. GUR-tree의 성능 실험 결과

그림 6(a)에서 보듯이 불확실성 영역이 커질수록 GUR-tree의 검색 시간이 감소하나, 그림 6(b)에서 보듯이 불확실성 영역이 커질수록 GUR-tree의 갱신 시간이 UR-tree에 비해 증가한다. 그림 6(c)나 그림 6(d)에서 보듯이 불확실성 영역이 클수록 error가 증가하는데 이는 검색 결과로 나와야 할 데이터가 검색되지 못하거나 검색 결과로 나오지 않아야 할 데이터가 검색되기 때문이다.

그림 6(e)~그림 6(h)는 그룹 영역을 0.0에서 1.0까지 변화시키며 갱신 영역에 따른 GUR-tree의 성능을 실험한 결과를 보여준다. 본 논문에서 GUR-tree<p>는 갱신 영역이 p인 GUR-tree를 나타낸다. 즉, GUR-tree<0.1>은 갱신 영역이 0.1임을 나타낸다. 그림 6(e)나 그림 6(f)에서 보듯이 그룹 영역이 약 0.5일 때 검색 시간은 최저이나 갱신 시간이 최고이다. 이는 0.5보다 작을 경우 그룹의 개수가 증가하기 때문이며, 0.5보다 클 경우는 그룹 영역이 커짐에 따라 그룹내의 데이터들에 대한 갱신이 줄어들기 때문이다. 그림 6(g)와 그림 6(h)에서 보듯이 검색의 정확도는 크게 차이가 나지 않는다.

5. 결 론

본 논문에서는 유비쿼터스 센서 네트워크에서 불확실한 데이터 처리를 위한 인덱스인 UR-tree를 살펴보고, 이 인덱스에 그룹화 기법을 적용함으로써 효율적인 검색을 지원하기 위한 GUR-Tree를 제시하였다.

성능 평가를 통해 GUR-tree는 불확실성 영역이 크면 검색 성능은 좋아지나, 정확도가 약간 감소되었다. 또한, R-tree에 비해서는 갱신 속도가 빠르나, UR-tree에 비해 갱신 성능이 떨어졌다. 그리고 GUR-tree는 그룹 영역이 커질수록 검색

성능은 좋아지나, 그룹 영역이 특정 범위를 벗어날 경우 그룹의 개수가 증가함으로 인해 검색 성능이 오히려 저하되었다.

참고문헌

- [1] Cheng, R., Prabhakar, S., "Managing Uncertainty in Sensor Databases," SIGMOD Record, Vol.32, No.4, 2003, pp.41-46.
- [2] Cheng, R., Xia, Y., Prabhakar, S., Shah, R., Vitter, J.S., "Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data," Proc. of the 30th Intl. Conf. on Very Large Databases, 2004, pp.876-887.
- [3] Civilis, A., Jensen, C.S., IEEE, Pakalnis, S., "Techniques for Efficient Road-Network-Based Tracking of Moving Objects," IEEE Transactions on Knowledge and Data Engineering, Vol.17, No.5, 2005, pp.698-712.
- [4] Dai, X., Yiu, M.L., Mamoulis, N., Tao, Y., Vaitis, M., "Probabilistic Spatial Queries on Existentially Uncertain Data," Proc. of the 9th Intl. Symposium on Spatial and Temporal Databases, 2005, pp.400-417.
- [5] Trajcevski, G., Wolfson, O., Hinrichs, K., Chamberlain, S., "Managing Uncertainty in Moving Objects Databases," ACM Transactions on Database Systems, Vol.29, No.3, 2004, pp.463-507.
- [6] Kim, D.O., Hong, D.S., Kang, H.K., Han, K.J., "UR-Tree : An Efficient Index for Uncertain Data in Ubiquitous Sensor Networks," Proc. of the Int. Conf. on Grid and Pervasive Computing, 2007, pp.603-613.