

## VoR-나무에서의 근방질문처리를 위한 자료삽입과 삭제알고리즘에 대한 연구

손 명 속

경애하는 최고령도자 김정은동지께서는 다음과 같이 말씀하시였다.

《현시대는 과학기술의 시대이며 과학기술의 발전수준은 나라의 종합적국력과 지위를 규정하는 징표로 됩니다.》(《조선로동당 제7차대회에서 한 중앙위원회사업총화보고》 단행본 38페이지)

선행연구[1, 2]들에서는 공간자료를 비롯한 다차원자료의 색인과 질문처리의 효율을 높이기 위한 R-나무(region-tree)자료구조와 그것의 변종들이 많이 연구되였다.

나무는 계층구조로서 린접한 자료모임(영역)들에 대한 국부근방구역을 탐색할 때 잎들사이를 순회하는데 많은 마디들을 방문하므로 국부근방구역탐색에서 효율이 떨어진다.

한편 보로노이도는 국부근방구역탐색속도가 빠르지만 탐색구역을 찾아내는것은 어렵다.

론문에서는 R-나무와 보로노이도를 결합하여 근방질문처리효율을 높일수 있는 VoR-나무(voronoi-tree)를 정의하고 VoR-나무에서의 자료삽입, 삭제에 대한 알고리즘을 제안하였다.

### 1. VoR-나무의 정의

여기서는 공간색인과 국부근방탐색성능을 높이기 위하여 R-나무(영역나무)와 보로노이도를 결합하여 VoR-나무를 구축한다.

보로노이도[1]는 기하공간에서 국부근방탐색에 아주 효율적인 자료구조이다. 모든 자료점들에 대한 R-나무와 보로노이도  $VD(P)$ 가 이미 구축되어있고 R-나무의 잎들에는 자료점모임  $P$ 의 부분모임과 해당하는 점들에 대한 추가정보를 포함하는 기록들이 보관된다. 이로부터 어떤 점  $p$ 의 기록에  $p$ 의 매 보로노이근방( $VN(p)$ )에로의 지적자와  $p$ 의 보로노이세포의 정점들( $V(p)$ )을 추가적으로 보관할수 있다.

론문에서는 VoR-나무를 다음과 같이 정의하였다.

정의 다음과 같은 나무를 차수  $(m, M)$ 인 VoR-나무라고 부른다.

① 뿌리가 아닌 모든 잎은  $M$ 개까지의 항목을 가질수 있다. 최소항목수는  $m$ 이다. 매 항목은  $(point, p)$ 의 형식을 가진다. 여기서  $point$ 는 마디에 속하는 점들의 자리표이고  $p$ 는 그 점의 기록에로의 지적자이다.

② 모든 내부마디가 보관할수 있는 항목수는  $m$ 과  $M$ 사이의 값이다. 매 항목은  $(mbr, p)$ 의 형식을 가진다. 여기서  $p$ 는 마디의 자식에로의 지적자,  $mbr$ 는 공간상에서 자식들을 포함하는 MBR(Minimum Bounding Rectangle)를 둘러싼 MBR이다. 나무의 잎들은 자

료기지객체들의 지적자를 포함한다.

③ VoR-나무의 모든 잎들은 같은 준위에 있다.

그림 1에 VoR-나무의 실례를 보여주었다.

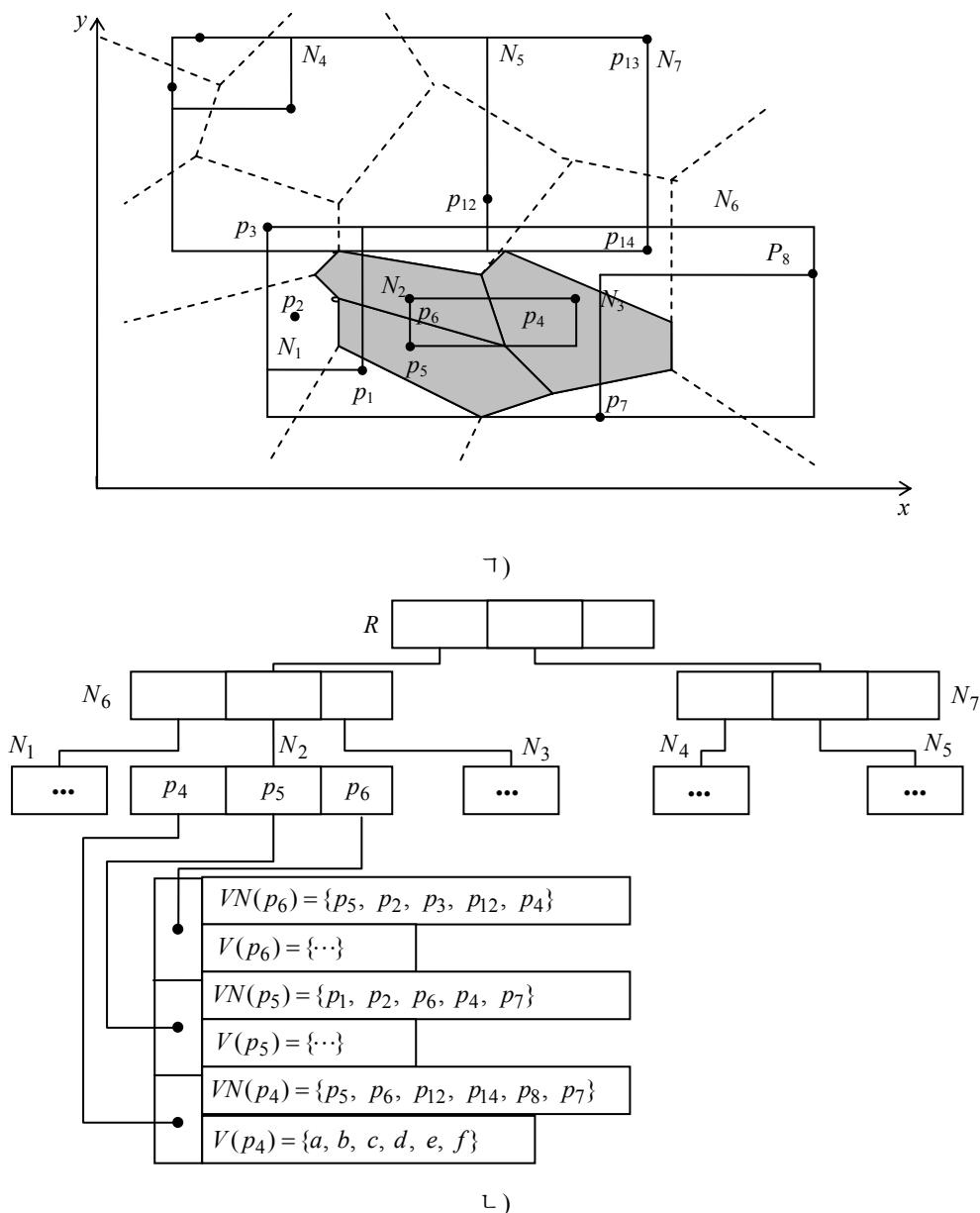


그림 1. VoR-나무의 실례

1) 보로노이도, 2) 1)에 대응하는 VoR-나무

큰 4각형구역을 리용하여  $V(p_3)$ 과 같은 보로노이세포들을 유한구역들로 제한한다. 그림 1에서는 간단히 점  $p_4, p_5, p_6$ 을 포함한 잎마디  $N_2$ 의 내용만을 보여주었다.  $N_2$ 의 매 점  $p$ 와 려관된 기록은  $p$ 의 보로노이근방들과 보로노이세포의 정점들을 포함하고있다. 이 기록을 보로노이기록이라고 부른다.

## 2. VoR-나무에서의 삽입과 삭제

삽입알고리즘은 다음과 같다.

- ① 먼저 대응하는 R-나무에 점  $x$ 를 삽입
- ② 점모임  $S$ 를 빈모임으로 설정한다.
- ③ 근방질문처리를 리용하여 점  $x$ 와 제일 가까운 점  $p$ 를 찾아 모임  $S$ 에 넣는다.
- ④ 선분  $xp$ 의 수직2등분선  $\perp(x, p)$ 와 사귀는 보로노이변들에 대응하는  $p$ 의 보로노이 근방들을 찾아 모임  $S$ 에 추가한다.
- ⑤ 모임  $S$ 의 매 원소들에 대하여 ④의 과정을 원소의 추가가 더는 없을 때까지 반복한다.
- ⑥ 모임  $S$ 의 매 원소  $s$ 에 대하여  $\perp(x, s)$ 에 의해  $V(s)$ 를 잘라내어  $s$ 의 보로노이기록들을 변경시킨다.
- ⑦ 잘라낸 구역들로  $x$ 의 보로노이세포를 구성하고 해당한 정보를  $x$ 의 보로노이기록에 넣는다.

그림 2의 실례에서  $x$ 의 삽입과정을 본다면  $x$ 를 먼저 대응하는 R-나무에 삽입한다. 다음 새로 추가된  $x$ 의 보로노이기록을 만들고 일부 점들에 대하여 보로노이기록을 변경시켜야 한다. 다음  $x$ 의 근방점으로  $p_1$ 을 찾고  $x$ 를  $p_1$ 의 보로노이세포안에 놓는다.

알고리즘의 수행과정에 보로노이기록들이 변경될 점모임  $S$ 에는 점  $p_1, \dots, p_4$ 가 들어있게 된다. 선분  $xp_1$ 의 수직2등분선을 리용하여  $V(p_1)$ 을 자르고  $p_1$ 의 기록에 새 세포를 기억시킨다.  $p_1$ 의 보로노이기록들도 갱신한다. 이 과정을  $p_2, \dots, p_4$ 에 대해서도 적용한다.

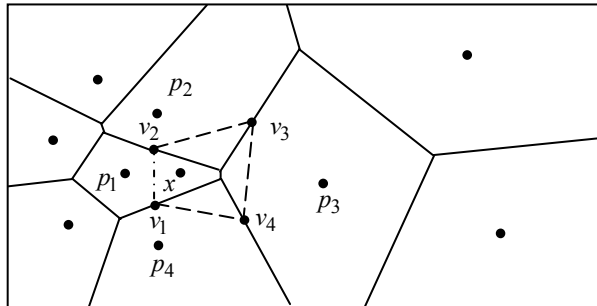


그림 2.  $x$ 의 삽입실례

$x$ 의 보로노이근방들은 갱신된 보로노이세포들의 생성점들의 모임으로 설정된다. 영향 받은 점들인  $p_1, \dots, p_4$ 는 VoR-나무에 기억된 보로노이근방들과 보로노이세포를 리용하여 직접 구할수 있다.

삭제알고리즘은 다음과 같다.

- ① 대응하는 R-나무를 리용하여  $x$ 의 보로노이기록을 찾는다.
- ② 이 기록을 통하여  $x$ 의 보로노이근방들을 호출한다.
- ③  $x$ 의 근방들의 매 쌍의 수직2등분선들의 교차를 리용하여 그것의 보로노이세포들을 갱신한다.

④ 그 점들의 근방들에서  $x$ 를 제거하고 새 근방점을 구성한다.

⑤ 다음 대응하는 R-나무로부터  $x$ 를 삭제한다.

점  $x$ 의 위치변경에 대한 VoR-나무의 갱신조작은 삽입과 삭제연산을 리용하여 진행한다. 삽입과 삭제연산이 수행될 때  $x$ 의 보로노이근방들만이 변경된다. 이 변경은  $x$ 를 리용하여 직접 호출되어 적용된다. 한점의 보로노이근방의 최대개수는 6이므로 이에 대한 탐색시간이 R-나무에서보다 거의 1.5배 줄어든다.

## 맺는 말

R-나무자료구조에 보로노이도를 결합한 VoR-나무구조를 정의하고 그에 기초한 VoR-나무에서의 자료삽입, 삭제에 대한 알고리즘을 제안함으로써 그것의 근방질문처리에서 탐색속도를 1.5배 높였다.

## 참고 문헌

[1] 최미향; 정보과학과 기술, 4, 48, 주체105(2016).

[2] Yannis Manolopoulos et al.; R-Trees: Theory and Applications, Springer, 120~129, 2008.

주체107(2018)년 11월 5일 원고접수

## Research on Inserting Algorism and Deleting Algorism for Neighborhood Query Processing in VoR-tree

*Son Myong Suk*

In this paper, we defined VoR-tree that can promote the efficiency of neighborhood query processing by combining R-tree with voronoi diagram, and proposed delete and insert algorithm in VoR-tree.

Key words: R-tree, voronoi diagram