

## 전자해도 응용 서비스를 위한 SENC 자료구조의 설계

김영진<sup>0</sup>, 권두위\*, 박수현\*

<sup>0</sup>동서대학교 컴퓨터공학과

\*동서대학교 컴퓨터공학과

e-mail: kimnzzang@naver.com<sup>0</sup>, kdoowy@hanmail.net\*, subak@gdsu.dongseo.ac.kr\*

## Design of SENC Datastructure for ENC Application Service

Young-Jin Kim<sup>0</sup>, Doo-Wy Kwon\*, Su-Hyun Park\*

<sup>0</sup>Dept. of Computer Engineering, Dongseo University

\*Dept. of Computer Engineering, Dongseo University

### ● 요약 ●

본 논문에서는 전자해도 응용 서비스를 위한 SENC 자료구조의 설계를 제안한다. SENC란 System Electronic Navigational Chart의 약어이며 전자해도 시스템이라고 부른다. 전자해도 시스템의 배포 형식은 전자해도 데이터를 전자해도 시스템으로 운용하는 시스템 환경에 맞게 재구성한 SENC 데이터로 저장, 관리한다. SENC를 구성하기 위해선 방대한 양의 전자해도 데이터가 필요하고 이 많은 데이터들을 저장, 관리, 검색할 수 있는 알고리즘이 필요하다. 전자해도 시스템을 효율적으로 관리하기 위한 알고리즘들을 설계하고 이 알고리즘을 이용한 간단한 구현화면을 제작하였다.

키워드: SENC, 알고리즘(Algorithm), 전자해도(ENC), B-Tree, 자료구조(Data Structures)

### 1. 서론

ENC(Electronic Navigational Chart)라 불리는 전자해도란 선박의 항해를 위해 제작된 전자 지도를 말한다. 전자해도 기반의 응용 서비스를 제공하기 위해서는 전자해도 데이터를 효과적으로 저장, 관리, 검색할 수 있어야 한다. 전자해도 시스템의 배포 형식은 전자해도 데이터를 전자해도 시스템으로 운용하는 시스템 환경에 맞게 재구성한 SENC 데이터로 저장, 관리한다. 이러한 전자지도 를 구성하기 위해서는 수많은 데이터들이 필요하고 저장되어 있다. 수많은 데이터들을 효율적으로 관리하기 위해서 현재 지도를 서비스하고 있는 많은 업체들이 나름대로의 알고리즘을 이용하여 서비스를 하고 있다. 전자해도 또한 바다 위에 표시되어야 할 많은 데이터들이 있고, 이를 효과적으로 처리하고, 관리할 수 있는 알고리즘의 구현

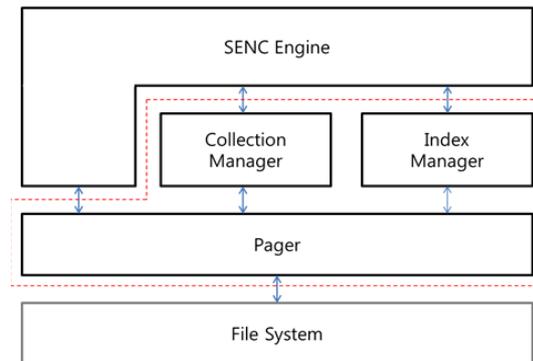


그림 1. 전자해도 자료구조와 엔진 관계도  
Fig. 1. ENC Datastructure and Engine Relationship

이 필요하다. 이러한 알고리즘을 구성함으로써 인해서 빠르게 데이터를 처리하여, 전자해도를 사용하는 사람들에게 보다 만족할 만한 서비스를 제공해 줄 수 있을 것이라고 생각한다. 여러 알고리즘을 이용하여, 알고리즘 커스텀마이징을 통해 전자해도 데이터를 설계하고자 한다.

## II. 본 론

그림 1은 기본 개념 구조도이다. 빨간색 부분에서 Pager를 제외한 Index Manager와 Collection Manger가 핵심 부분이지만 본 논문에서는 Index Manager를 중심으로 다루고 있다.

그림 2의 Index Manager 모듈은 특정 전자해도 데이터를 쉽고 빠르게 찾을 수 있도록 색인(Index)의 구성과 색인 정보의 관리를 담당하는 모듈이다. 전자해도 데이터를 구성하는 데이터 구성요소의 특성에

따라 여러 색인을 생성할 수 있다. 전자해도 데이터의 공간 정보를 위해서는 공간 색인을 구성하고, feature 데이터들의 속성 값을 키로 하는 value index를 생성 할 수 있도록 하였고, 공간색인은 최소경계사각형(MBR : Minimum Bounding Rectangle)을 검색 Key로 해당 범위 내에 존재하는 Feature Object들의 정보를 찾을 수 있도록 한다.

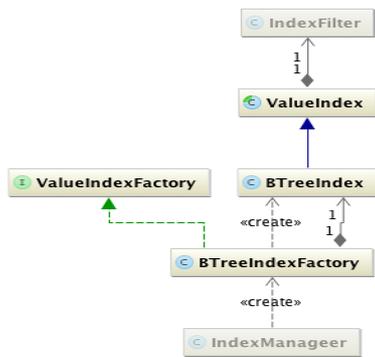


그림 2. 인덱스 관리자  
Fig. 2. Index Manager

인덱스 관리자를 위한 알고리즘을 구현하기 위해 Java를 이용하여 인덱스 알고리즘 중 하나인 B-Tree를 구현하였다.

그림 3은 구현 후 결과 화면인데 화면을 보면 처음 삽입 되었을 때 순차적으로 나열되어있던 숫자들이 B-Tree에 삽입이 되면서 그림 3의 정렬과 같이 트리형태로 자료가 저장이 된다. 본 논문에서 이러한 인덱스 알고리즘으로 자료를 저장하려는 이유는 그림 3에서 45를 찾으려고 하면 B-Tree에 삽입 전에는 11번의 순차검

색을 통해 자료를 찾게 되지만 B-Tree 에 삽입 후에는 단 3번의 인덱스 검색으로도 자료를 쉽게 찾을 수 있기 때문이다. 이러한 인덱스 알고리즘은 균형이 중요하고 균형이 흐트러지면 평균 검색시간이 느려지는 특성이 있다.

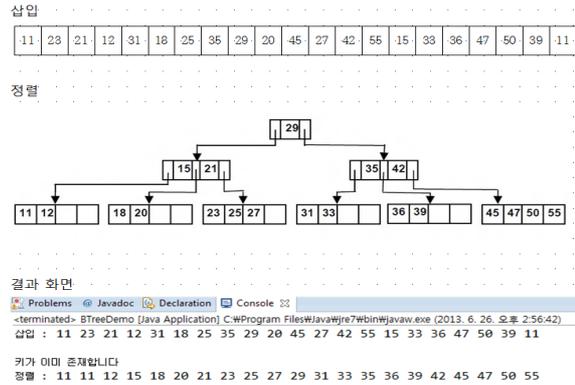


그림 3. B-Tree 구현화면  
Fig. 3. B-Tree Implementing Screen

## III. 결 론

전자해도에는 날씨와 바다의 정보, 배들의 위치 같은 수많은 정보가 저장되어 있다. 이런 수많은 정보가 저장되어 있는 전자해도의 데이터들을 효율적으로 관리하고, 검색을 할 수 있는 여러 가지 알고리즘들이 필요하다. B-Tree와 유사한 B+Tree, R-Tree, R+ Tree같은 알고리즘을 이용하여, 방대한 양의 데이터들을 보다 빠르고, 간단하게 관리하게 될 것이라고 생각한다. 또한, 빠른 속도로 데이터를 관리할 수 있어서 데스크 탑이 아닌 모바일 기기에서도 사용을 할 수 있게 될 것이다. 이를 통해 배의 선장이나 도선사들도 간단하게 모바일 기기만 있으면 배 주변의 정보를 쉽게 얻을 수 있다.

## 참고문헌

- [1] Lee Hee Young, Advanced ENC Standards and International trends ,TTA journal, no.131 pp.89 - 96, 2010.
- [2] MBR R-Tree ,http://blog.daum.net/maxmin93/13380784
- [3] Oh young chul, Data Structure Written by C , pp.234-292, Jan 2008
- [4] Lee Suck Ho, File Processing Theory, pp.228-264, Mar 1985
- [5] B-Tree, http://sourcecode4all.wordpress.com/2012/03/26/b-tree/