

# vp tree에서 효과적인 k-Nearest Neighbor 검색 방법

김민욱<sup>o</sup>, 윤경로  
 건국대학교 컴퓨터공학부  
 {minuk<sup>o</sup>, yoonk}<sup>o</sup>@konkuk.ac.kr

## Effective k-Nearest Neighbor Search method based on vp tree

Min-Uk Kim<sup>o</sup>, Kyoungro Yoon  
 Department of Computer Science and Engineering, Konkuk University

### 요 약

vp tree는 기준점(vantage point)과의 거리를 기준으로 데이터베이스 내의 자료를 색인하는 자료구조이다. 멀티미디어 자료 검색에서 비슷한 정도는 객체간의 거리를 바탕으로 비교하고, vp tree 색인 구조는 이 과정을 내포하고 있기 때문에 최근 멀티미디어 검색 연구에서 많이 사용되고 있다. 검색 방법에는 query와 가장 가까운 대상을 찾는 Nearest Neighbor Search, 또는 query와 가까운 k등까지를 검색하는 k-Nearest Neighbor Search가 있다. 본 논문에서는 Content-based retrieval에서 최근 자주 사용되는 vp tree에서 효과적인 k-NNS 방법을 제안하고, 기존의 전형적인 k-NNS 방법과의 비교 실험 결과를 보인다.

**키워드:** vp tree, content-based retrieval, k-nearest neighbor search

## 1. 서 론

현대 사회는 다양한 멀티미디어 기기(디지털 카메라, 캠코더 등)에서 생산되는 멀티미디어 데이터의 양이 많아지고, 사용자의 질의 요구사항도 복잡해지고 있다. 이렇게 많은 양의 멀티미디어 데이터는 인터넷을 통해 손쉽게 공유되고, 접근이 가능해졌다. 그러면서 기존의 키워드기반 검색의 한계점을 인식하고 내용기반 방식이 1990년대부터 연구되었다[1].

내용기반 검색에서 객체들은 d차원의 vector로 표현될 수 있고, 보다 빠르게 검색하기 위해 클러스터링이나 색인을 사용한다[1].

본 논문에서는 vp tree에서 k등까지 검색하는 방법을 기존방법과 제안하는 방법을 비교 실험한다.

논문의 구성은 다음과 같다. 2장에서는 vp tree 및 관련기술에 대해서 설명하고, 3장에서는 제안하는 k-NN S방법에 대해서 설명한다. 4장에서는 기존 k-NNS 방법과 제안한 방법의 비교 실험 결과를 보이고 5장에서 결론과 향후 과제를 남긴다.

## 2. 관련 연구

### 2.1. vp tree

vp tree는 1990년대에 Yianilos가 만든 자료구조[2]이

※ 본 연구는 서울시 산학연 협력사업(10581)의 지원에 의해 수행되었음.

며, 기존의 k-d tree와 다르게 색인하는 기준이 기준점(vantage point)과 데이터베이스내의 데이터간의 거리이다.

따라서 고차원의 특징소 벡터에서는 서로간의 거리를 통해 유사도를 측정하기 때문에 절대좌표를 통한 나누기 방법보다는 이 방법이 훨씬 직관적이고 실제로 고차원의 특징소 벡터로 객체가 표현되는 내용기반 검색에 적용이 많이 되고 있다.

### 2.2. MPEG-7 Visual Descriptors

ISO/IEC JTC1 SC29 WG11 (MPEG)에서는 멀티미디어 검색과 관련하여 MPEG-7 (Multimedia Description Interface) 표준을 제정하였다[3]. 특히 Part 3 (Visual)은 영상의 시각적 특성에 기반을 둔 다양한 표현 기술들을 제정하였다. 시각 기술자의 경우, 색상(Color), 모양(Shape), 질감(Texture)으로 구분하여 각각 다양한 기술자들을 정의하여 특정한 용도에 맞추어 사용할 수 있게 하였다.

본 논문에서는 MPEG-7의 시각 기술자 중 색상 부분에서 색상 구조 기술자(Color Structure Descriptor)를 바탕으로 실험하였다.

#### 2.2.1. MPEG-7 Color Structure Descriptor

색상 구조 기술자(Color Structure Descriptor)는 영상의 색상 막대그래프(Color Histogram)정보뿐만 아니라, 색상의 구조적인 정보까지 가지고 있는 기술자이다.

색상 구조 기술자의 특징소를 추출하는 단계는 다음과 같다.

1. 영상을 256구역으로 양자화된 HMMD 색공간으로 변환한다.

2. 256개의 저장 공간(BIN)을 가진 누적된 색상 구조 막대그래프를 얻는다.

3. 256개의 저장 공간에 있는 각각의 값이 통계적으로 얼마나 영상에 나타나는지 비선형적으로 양자화 한다

이렇게 양자화된 각각의 값이 색상 구조 기술자를 통해 추출된 특징소이다.

색상 구조 기술자는 256, 128, 64, 32 개의 다양한 길이의 저장 공간을 지원하는데 256개가 아닌 길이의 경우 위의 2번에서 해당되는 개수로 통합된다

본 논문에서는 256개의 bin을 사용했고, 따라서 feature vector의 차원수는 256이다.

### 2.3. Average Normalized Modified Retrieval Rank

ANMRR은 MPEG-7 표준화 과정에서 Visual Descriptor들의 검색 성능을 측정하기 위해 개발된 검색 정확도 측정도구(Retrieval Accuracy Measure)이다[3].

NMRR은 기존의 검색결과를 평가하는 측정도구보다 좀 더 신뢰할 만한 결과를 보여주는데 얼마나 많은 수의 비슷한 영상이 검색이 되는지 뿐만 아니라 얼마나 상위에 유사한 영상이 검색되는지도 측정도구에서 계산하여 알려주기 때문이다

ANMRR을 구하는 식은 다음과 같다

$$ANMRR = \frac{1}{NQ} \sum_{q=1}^{NQ} NMRR(q) \quad (1)$$

(1) 식에서 NQ는 총 query 개수이고, NMRR은 다음과 같이 구한다.

$$NMRR(q) = \frac{\left( \frac{1}{NG(q)} \sum_{k=1}^{NG(q)} Rank(k) \right) - \frac{1 + NG(q)}{2}}{1.25 \times K(q) - \frac{1 + NG(q)}{2}} \quad (2)$$

(2) 식에서 NG(q)는 query q 에 대한 Ground-Truth set에 들어있는 영상의 개수를 말하고 Rank(k)는 기준 집합 안의 k번째 영상이 결과 값에서 몇 등인지 알려주는 함수이다.

$$Rank(k) = \begin{cases} Rank(k) & \text{if } Rank(k) \leq K(q) \\ 1.25 \times K(q) & \text{if } Rank(k) > K(q) \end{cases} \quad (3)$$

$$K(q) = \min\{4 \times NG(q), 2 \times \max[NG(q), \forall q]\}$$

NMRR은 [0, 1] 사이의 값을 갖게 되고 작은 값일수록 검색의 정확도가 높다는 것을 의미한다

본 논문에서 추구하는 결과는 VP Tree를 사용하지 않고 전역 검색한 결과와 동일한 결과를 VP Tree를 사용하여 구현하는 것이므로 Ground-Truth Set은 전역 검색한 결과에서 최상위 100등까지의 이미지로 결정하였다. 따라서 NG(q)는 100이다.

### 3. 제안하는 k-Nearest Neighbor Search 방법

클러스터링된 자료구조에서 k-Nearest Neighbor Search는 Query q에 임계값  $\sigma$ 를 정하고, 그 범위 안에 들어오는 클러스터들을 검색하는 것이 기본적인 방법이다.

따라서, vp tree에도 기존 방법을 적용할 수 있고 그림 1에서 볼 수 있는 것처럼 Query q의 임계값  $\sigma$ 가 기준점 v의 범위 안에 포함된다면, 비록 q가 v의 범위 바깥이지만 안쪽도 탐색해야 한다[4]. 반대의 경우도 비슷한 방법으로 결정된다.

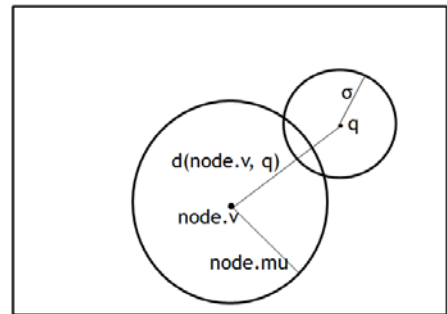


그림 1 기준점 v 와 Query q의 예

하지만, 입력되는 query의 범위  $\sigma$ 를 알기 위해선 예측하거나, 사용자의 입력이 필요하다 따라서 제안하는 vp tree에서 k-NNS 방법은 표 1과 같다.

표 1 제안하는 vp tree에서 k-NNS

```

knns(vptree root, query q, int k) {
    searchTree(root, q, terminalNodes, 0)
    terminalNodes.sort()
    subsetList = subset(terminalNodes)
    subsetList안에 있는 객체들과 거리비교 후 k등까지 반환
}

searchTree(vptree node, query q, List list, double e) {
    dist = d(node.v, q)
    if (leaf) list.add(node.list, e)
    else
        searchTree(node.left, q, list, e+dist-node.mu)
        searchTree(node.right, q, list, e-(dist-node.mu))
}
    
```

위 알고리즘에서 vptree, query, terminalNodes는 각각 vptree node, feature vector로 표현되는 query, vptree 에서 단말노드의 list를 나타낸다. vptree 노드 구성은 그

림 2와 같다.

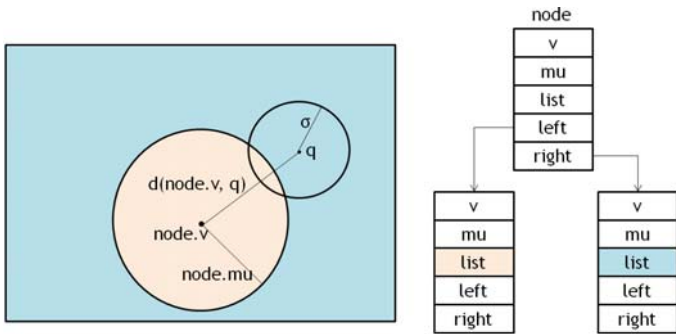


그림 2 vp tree node 구조

vp tree의 노드는 feature vector  $v$ , left node와 right node를 나누기 위해 사용한 임계값  $\mu$ , 자신이 가진 DB list, 그리고  $\mu$ 보다 작은 거리 값을 가진 객체를 가진 left node 포인터,  $\mu$ 보다 크거나 같은 거리 값을 가진 객체를 가진 right node 포인터로 구성된다.

$\text{knn}(n)$ 에서 vp tree의 root를 받아서  $\text{searchTree}()$ 를 호출한다.  $\text{searchTree}()$ 에서는 단말노드가 아니면 vp tree의 전체 노드를 탐색하면서  $e$ 값을 증가 또는 감소시키게 된다. 만약 Query  $q$ 가 어느 노드의 범위  $\mu$ 안에 들어온다면, 왼쪽자식노드를 탐색할 때는  $e$ 값이  $d(\text{node.v}, q)$ 에서  $\text{node.mu}$ 를 뺀 차이만큼 더해져서 계산하게 된다. 따라서 vp tree안의 노드를 계속 탐색해서 단말노드에 도착했을 때, 가장 적합한 노드는 가장 작은  $e$ 값을 갖게 된다. 왜냐하면,  $q$ 가 vp tree의 중간 노드를 거칠 때마다 범위  $\mu$ 와 비교를 하게 되고,  $\mu$ 보다 바깥쪽이거나 안쪽이면  $e$ 값은 음수를 갖게 된다. 따라서 작은  $e$ 값일수록  $q$ 에 많이 속해 있는 노드가 되는 것이다.

마지막으로  $\text{knn}(n)$ 은  $e$ 값으로 vp tree의 단말노드를 오름차순으로 정렬하고 그 중 몇 개를 선택해서 그 범위에서만 객체간의 거리를 계산하게 된다.

본 논문에서는 vp tree Level을 2~12까지 변화를 주면서, 단말노드는  $1 \sim 2^{\text{treeLevel}-2}$ 까지 선택하면서 검색결과를 비교하였다.

#### 4. 실험 결과

본 논문에서는 정지영상을 대상으로 MPEG-7 Color Structure Descriptor를 기반으로 vp tree를 만들었다. MPEG-7 CSD의 feature vector 차원 수는 256이고, vp tree의 기준점은 DB안의 임의의 feature vector를 선택했다. 실험 환경은 표 2와 같다.

실험 데이터는 5개의 서로 다른 랜덤 기준점으로 만들어진 vp tree에서 각각 1000장의 테스트 정지영상의 결과를 구하고, 평균을 구했다. ANMRR을 구하는데 Ground-Truth set은 해당 query의 전체 검색 결과

TOP 100을 사용했다. 5개의 vp tree의 기준점은 DB안에서 랜덤하게 선택된 객체를 선택하는 방법으로 만들었다.

표 2 실험 환경

Feature Extraction	MPEG-7 CSD(256d)
정지영상 수	4,331,983장
테스트 정지영상 수	1,000장
vp tree 개수	5개
vp tree Level	2, 4, 6, 8, 10, 12

그림 3에서 MBM-knns는 제안하는 방법의 결과이고  $\text{knn}(n)$ 은 기존 knns방법의 결과이다 여기서  $n$ 은 실제 query의  $n$ 번째 객체의 거리를 범위를 range search의 범위  $\sigma$ 로 주고 실험했다는 의미이다 따라서  $\text{knn}(100)$ 은 전체 검색결과 TOP 100과 정확히 일치하는 결과를 내준다. 하지만 그림 3에서 볼 수 있는 것처럼, 실제로 2 등까지의 거리를 범위  $\sigma$ 로 줘도 거의 전체범위를 대상으로 검색하는 것을 확인할 수 있다.

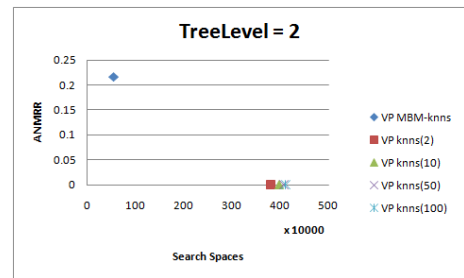


그림 3 vp tree level 2

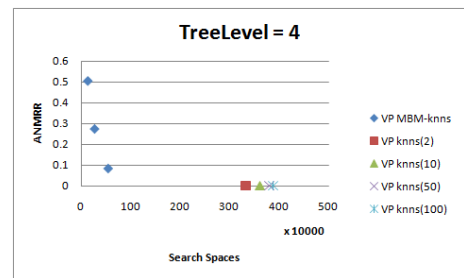


그림 4 vp tree level 4

그림 4에서 MBM-knns는 단말노드 선택을 우선순위로 각각 1개, 2개, 4개를 선택한 결과이다.

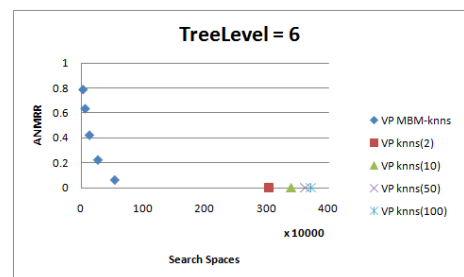


그림 5 vp tree level 6

그림 5에서 MBM-knns는 단말노드를 각각 1개, 2개, 4개, 8개, 16개 선택한 결과이다 여전히 range search 결과는 거의 전체 영역을 대상으로 검색하고 있다

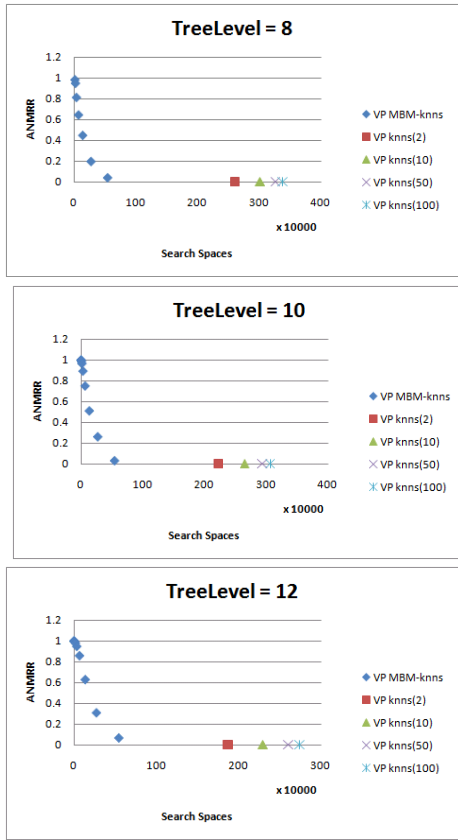


그림 6 vp tree level 8, 10, 12

그림 6는 vp tree의 tree level을 8, 10, 12로 만들고 실험한 결과를 보여준다 tree level이 늘어날수록 제안한 knns방법은 ANMRR이 0에 근접해가고 있다. 기존의 range search 방법은 2등까지의 범위, 즉 범위가 작을수록 검색범위가 작아지고 있다 하지만 여전히 전체DB의  $\frac{1}{2}$  이상을 검색하고 있다

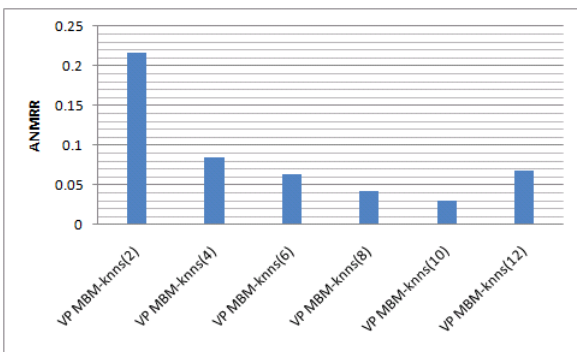


그림 7 제안한 knns의 tree level별 실험 결과

그림 7은 제안한 knns방법을 통해 전체DB의  $\frac{1}{8}$ 에서 검색했을 때의 ANMRR이다. 그림 6에서 볼 수 있는 것처럼,

vp tree를 더욱 더 잘게 나눌수록 결과치가 좋아지다가 일정 tree level을 넘어가면 나빠지는 것을 확인할 수 있다.

### 5. 결론 및 향후 과제

본 논문에서는 vp tree에서 기존 range search 방법을 적용한 k-NNS와 제안한 k-NNS를 비교 실험 하였다 vp tree에서 적합한 k-NNS를 하기 위해 제안한 방법은 전체DB의  $\frac{1}{8}$ , 약 12%만 검색해도 전체검색한 결과 TOP 100과 거의 유사함(ANMRR<0.05)을 실험결과로 보였다 또한 기존의 knns를 하기 위한 범위  $\sigma$ 를 본 논문에서는 미리 검색해서 결과를 구했지만 실제 검색시스템에서는 입력되는 query의 범위를 예측하거나 사용자의 입력을 받을 수밖에 없다.

따라서 본 논문에서 제안하는 방법은 vp tree를 사용하여 대용량 아카이브에서 내용기반 검색을 할 때 보다 적은 노력(전체 검색 공간의 약 12%)으로 정확한 검색 결과를 얻을 때 유용하게 사용될 수 있다 향후에 k-means clustering과 vp tree를 비교 실험할 예정이다

### 참고 문헌

- [1] R. Datta, D. Joshi, J. Li, J.Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. In *ACM Computing Surveys*, Vol. 40, No. 2, pages 1-60, 2008.
- [2] P. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 311-321, 1993.
- [3] B. Manjunath, P. Salembier, T. Sikra. *Introduction to MPEG-7 Multimedia Content Descriptor Interface*. J.Wiley and Sons. 2002.
- [4] A. W.-C. Fu, P. M.-S. Chan, Y.-L. Cheung, Y. S. Moon. Dynamic VP-Tree indexing for N-Nearest Neighbor Search Given Pair-Wise Distances. *VLDB Journal*, volume 9(2), pages 154-173, 2000.