이미지 시퀀스 데이터베이스에서의 유사성 기반 서브시퀀스 검색

김 인 범† · 박 상 현††

요 약

본 논문은 다차원 타임 워핑 거리 함수를 이용하여 유사한 이미지 서브시퀀스를 신속하게 검색할 수 있는 색인 방법을 제안한다. 타임 워핑 거리는 시퀀스들의 길이가 다르거나 샘플링 비율이 다른 많은 응용에서 Lp 거리보다 더욱 적합하다. 우리가 제안한 색인 방법은 디스크 기반의 접미어 트리를 색인 구조체로 채택하고, 유사하지 않은 서브시퀀스를 잘못된 누락 없이 잘 여과하기 위해 하한 거리 함수를 사용한다. 이 방법은 특정 차원의 상대적 가중치를 손쉽게 부여하기 위해 정규화를 적용하고 색인 트리를 압축하기 위해 이산화 과정을 수행한다. 메디컬 이미지와 합성 이미지 시퀀스를 대상으로 한 실험은 본 논문에서 제안한 방법이 naïve한 방법보다 우수한 성능을 보이고 대용량의 이미지 시퀀스 데이터베이스로의 확장이 용이함을 입증한다.

Similarity-Based Subsequence Search in Image Sequence Databases

Inbum Kim† · Sanghyun Park††

ABSTRACT

This paper proposes an indexing technique for fast retrieval of similar image subsequences using the multi-dimensional time warping distance. The time warping distance is a more suitable similarity measure than Lp distance in many applications where sequences may be of different lengths and/or different sampling rates. Our indexing scheme employs a disk-based suffix tree as an index structure and uses a lower-bound distance function to filter out dissimilar subsequences without false dismissals. It applies the normalization for an easier control of relative weighting of feature dimensions and the discretization to compress the index tree. Experiments on medical and synthetic image sequences verify that the proposed method significantly outperforms the naïve method and scales well in a large volume of image sequence databases.

키워드 : 유사 검색(Similarity Search), 이미지 시퀀스 데이터베이스(Image Sequence Database), 타임 워핑(Time Warping), 접미어 트리(Suffix Tree), 색인(Index)

1. 서 론

이미지 시퀀스 데이터베이스는 이미지 시퀀스들의 집합으로 그 원소들은 이미지들의 순차 리스트(ordered list)이다. 이것의 예로는 환자의 폐를 주기적으로 찍은 이미지들과 파노라마 사진기를 이용해 찍은 사진들, 비디오 클립의 연속 프레임 등이 있다.

유사 검색(similarity search)은 주어진 질의 시퀀스와 비슷한 변화 패턴을 가지고 있는 시퀀스나 서브시퀀스를 찾는 연산[2, 3, 18]으로 데이터 마이닝(data mining)과 데이터 웨어하우징(data warehousing), 디지털 이미지/비디오 라이브러리(digital image/video libraries) 등과 같은 많은 응용에서 그 중요성이 점차 강조되고 있다[13, 32]. 특히 메디컬 영역에서, 어떤 환자의 현재 증상이나 특징을 가지고 과거의 유사한 특징을 갖는 의학적 정보를 참고할 수 있는 것은 의사들의 환자 치료에 큰 도움을 준다. 예를 들면 종양 전문 의사들이 어떤 종양 환자를 가장 적절하게 치료하기 위해 그와 유사한 종양 전이(tumor evolution) 패턴을 가진 데이터베이스를 검색하여 치료에 활용할 수 있을 것이다.

유사 검색은 전체 매칭(whole matching)과 서브시퀀스 매칭(subsequence matching)으로 구분된다[2]. 전체 매칭은 데이터 시퀀스와 질의 시퀀스가 같은 길이를 가진다는 가정하에 질의 시퀀스와 유사한 데이터 시퀀스를 찾는 것이다.

† 정 회 원 : 김포대학 컴퓨터계열 교수
†† 종신회원 : 포항공과대학교 컴퓨터공학과 교수
논문접수 : 2002년 11월 29일, 심사완료 : 2003년 3월 4일

서브시퀀스 매칭은 질의 시퀀스와 유사한 서브시퀀스를 데이터 시퀀스 안에서 찾는 것이다. 이 경우 질의 시퀀스의 길이에는 제약이 없다.

naïve한 유사 검색 방법은 각 이미지 시퀀스나 서브시퀀스를 데이터베이스로부터 차례로 읽은 후 질의 이미지 시퀀스와의 거리를 계산한다. 이 방법은 간단하지만 데이터베이스의 크기가 커진다면 심각한 성능 저하가 발생할 수 있다. 그러므로 유사 검색을 확대하기 위해서는 효과적인 색인 방법을 사용하는 것이 필수적이다.

시퀀스의 유사성 판단 기준을 정의하는 것은 쉽지 않다. 비록 정성적(qualitatively)으로 같은 시퀀스라 할지라도 정량적(quantitatively)으로는 같지 않을 가능성이 있기 때문이다. 그러므로 유사성의 판단 기준으로 단순히 유클리드 거리(Euclidean distance)를 사용하면 잘못된 누락(false dismissal)[2, 3]이 발생할 수도 있다. 그러므로 최근의 유사성 검색과 관련된 연구의 흐름은 스케일링(scaling)[3, 14], 쉬프팅(shifting)[3, 14], 정규화(normalization)[17, 20], 타임 워핑(time warping)[6, 23, 28, 29, 42]과 같은 다양한 변형을 지원하는 추세이다.

타임 워핑[6, 31]은 어떠한 시퀀스 원소들이라도 추가 비용 없이 필요한 만큼 자신의 복제를 허용하는 변환이다. 타임 워핑 거리는 타임 워핑에 의해 변형된 두 개의 시퀀스 사이의 최소 거리로 정의된다. 유클리드 거리(Euclidean distance)는 비교되는 두 시퀀스의 길이가 같을 경우에만 사용될 수 있는데 반해, 타임 워핑 거리는 임의의 길이의 어떠한 두 시퀀스에도 적용될 수 있는 장점이 있다. 따라서 타임 워핑 거리는 시퀀스들의 길이가 다르거나 시퀀스 원소들이 서로 다른 시간 간격을 가지는 이미지 시퀀스 데이터베이스에 적합하다.

효율적인 유사 검색을 위해, 대부분의 기존 방법[2, 3, 18]들은 각 시퀀스나 서브시퀀스를 다차원 상의 점으로 변환한 후, 색인 공간에서 유사하지 않은 시퀀스를 여과할 목적으로 다차원 점들 사이의 L_p 거리를 계산한다. 참고문헌[42]에서는 삼각 부등식(triangular inequality)을 가정하는 다차원 색인이 자신의 기본적인 거리 함수가 삼각 부등식을 만족하지 않는다면, 유사 검색시 잘못된 누락의 직접적 혹은 간접적인 원인이 된다고 주장하였다. 잘못된 누락[2, 3]은 실제로는 질의 시퀀스와 유사한 데이터 시퀀스가 최종 질의 결과에 포함되지 않는 것으로 정의된다. 참고문헌[42]에서는 타임 워핑 거리가 삼각 부등식을 만족하지 않음을 증명했다. 그러므로 삼각 부등식을 가정하는 다차원 색인들은 잘못된 누락을 허용하지 않는 응용에서 타임 워핑 거리를 사용할 수 없다.

우리는 이전 연구[28]에서, 각 시퀀스 원소들이 한 개의 수치 값을 가진다는 가정 하에, 타임 워핑 거리를 유사성 판단 기준으로 사용한 효율적인 서브시퀀스 매칭 방법을 제안했다. 이 방법은 색인 구조로 접미어 트리(suffix tree)[38]를 사용했는데, 이것은 삼각 부등식을 가정하지 않는다. 또한 색인 크기를 줄이기 위해 원소 값에 이산화 과정(discretization)을 도입하였다. 이 방법은 유사한 서브시퀀스를 누락 없이 검색하기 위해 접미어 트리를 순회하면서 하한 거리 함수를 여과 함수로 사용하였다.

본 논문은 우리가 이전에 수행한 연구[28]를 다차원 상으로 확장한 것이다. 즉 시퀀스의 원소가 단일 값을 가지는 것이 아니라 다중 값을 가지는 경우를 대상으로 한다. 거리 함수로 다차원 타임 워핑 거리 함수를 사용하며 인덱스 구조로 접미어 트리를 다시 사용한다. 인덱스의 크기를 줄이고 질의 처리 성능을 높이기 위해 이산화(discretization) 기법을 적용한다. 이 확장에서 시도된 주요 내용은 다음 세 가지다. 첫째는 다차원 타임 워핑 거리 함수를 상대적 가중치를 고려해서 정의하는 것이고, 둘째는 다차원 원소를 하나의 심볼로 이산화 하는 것이고, 마지막은 잘못된 누락 없이 유사하지 않은 서브시퀀스들을 여과하기 위해 하한 다차원 타임 워핑 거리 함수를 정의하는 것이다

2장에서는 본 논문에서 사용하는 기호와 다차원 타임 워핑 거리 함수를 정의한다. 3장과 4장에서는 색인을 작성하고 질의를 처리하는 알고리즘들을 기술한다. 5장에서는 제안된 방법의 성능과 확장성에 대한 실험 결과를 기술하며 6장에서는 관련 연구를 소개한다. 마지막으로, 7장에서 논문의 결론을 제시한다.

2. 문제 정의

이 장에서는 우리가 본 논문에서 사용한 기호와 여러 거리 함수를 정의한다. 또한 우리가 해결하려고 하는 문제를 정형적으로 정의한다.

2.1 기 호

n 개의 원소들을 가지고 있는 일차원 시퀀스는 기호 $X = \langle X[1], \cdots, X[n] \rangle$로 표현한다. $X[i]$는 X의 i번째 원소이고 $|X|$는 X의 원소 개수, 또는 X의 길이이다. $X[i:j]$는 i부터 j 위치까지의 원소들을 포함하는 X의 서브시퀀스이다. $X[i:-]$는 i번째 원소로부터 마지막 원소까지의 서브시퀀스이다. 즉 $X[i:-]$는 $X[i:|X|]$와 같다. 또한 $X[i:-]$는 i번째 원소로부터 시작하는 X의 접미어(suffix)이다. $\langle\rangle$는 원소가 하나도 없는 빈(empty) 시퀀스를 표현한다.

원소들이 숫자 값을 갖는 시퀀스들은 이산화(discretization) 작업에 의해 심볼 시퀀스들로 변환될 수 있다. X^c는 X로부터 변환된 심볼 시퀀스이다. $X^c[i]$는 X^c의 i번째 원소, $|X^c|$는 X^c의 원소 개수를 표현한다. $X^c[i:j]$와 $X^c[i:-]$

의 의미는 앞서 기술한 $X[i:j], X[i:-]$와 유사하다.

본 논문에서 사용하는 굵은 글씨체(bold font)는 다차원 시퀀스들의 표현이다. 즉, $\boldsymbol{X} = \langle \boldsymbol{X}[1], \cdots, \boldsymbol{X}[n] \rangle$는 n개의 원소들을 가지고 있는 다차원 시퀀스의 표현이다. $\boldsymbol{X}[i] = \langle \boldsymbol{X}[i][1], \cdots, \boldsymbol{X}[i][k] \rangle$는 k개의 숫자 값을 가지는 $\boldsymbol{X}$의 i번째 원소를 표기한다. 다차원 시퀀스의 모든 원소들은 같은 개수의 숫자 값을 가진다고 가정한다. 이미지 시퀀스는 다차원 시퀀스의 한 예이다.

2.2 거리 함수

2.2.1 Lp 거리

L_p 거리 함수는 임의의 두 시퀀스 X와 Y의 유사성을 결정하는데 광범위하게 사용되고 있다. L_1은 맨해튼 거리(Manhattan distance), L_2는 유클리드 거리(Euclidean distance)이고, L_∞는 임의의 원소 쌍에서의 최대거리이다[35]. L_p 거리 함수에서 비교되는 두 시퀀스들의 길이는 같아야 한다.

$$L_p(X, Y) = \left(\sum_{i=1}^{|X|} |X[i] - Y[i]|^p \right)^{1/p}, \quad p = 1, 2, \cdots, \infty$$

2.2.2 타임 워핑 거리

일반적으로 어떤 시퀀스들의 유사한 정도를 판단하는 것은 쉽지 않다. 비록 정성적으로 같은 시퀀스들이라도 정량적으로 다를 수 있기 때문이다. 시퀀스들은 길이가 서로 다를 수 있기 때문에, 시퀀스들을 색인 공간에 매핑하고 유사 정도의 결정을 위해 유클리드 거리를 계산하는 것은 어렵거나 불가능하다. 또한 시퀀스들의 샘플링 비율이 다른 경우에는 교차-상관관계(cross-correlation)와 같은 유사성 계산 기법을 사용할 수 없다. 음성 인식과 같은 영역[31]에서는 이와 같은 문제를 타임 워핑 거리를 사용해서 해결하고 있다[6, 31].

타임 워핑은 이산 값의 시퀀스를 연속 값의 시퀀스와 비교하기 위한 전통적인 알고리즘을 일반화한 것이다[31]. 두 시퀀스간의 최소 차이를 얻기 위해, 타임 워핑은 한 시퀀스의 각 원소들이 다른 시퀀스의 하나 이상의 이웃된 원소들과 매핑되는 것을 허용한다.

정의 1 : 임의의 두 개 시퀀스 X, Y에 대해, 타임 워핑 거리 D_{tw}는 다음과 같이 재귀적으로 정의된다[31].

$$\begin{aligned} &D_{tw}(\langle\rangle, \langle\rangle) = 0 \\ &D_{tw}(X, \langle\rangle) = D_{tw}(\langle\rangle, Y) = \infty \\ &D_{tw}(X, Y) = |X[1] - Y[1]| \\ &\qquad + \min \begin{cases} D_{tw}(X, Y[2:-]) \\ D_{tw}(X[2:-], Y) \\ D_{tw}(X[2:-], Y[2:-]) \end{cases} \end{aligned}$$

$D_{tw}(X, Y)$는 기본적으로 순환 관계(recurrence relation)를 이용한 동적 프로그래밍(dynamic programming) 기법[6]을 사용해서 계산될 수 있다. 동적 프로그래밍 알고리즘[6]은 그 실행 결과로 누적 거리 테이블(cumulative distances table) T를 생성한다. 최종 누적 거리 $T[|X|, |Y|]$는 X와 Y의 최소 거리이고 원소들의 매핑은 최소 누적 거리를 가지는 이전 셀을 선택함으로써 그 테이블 내에서 역으로 추적될 수 있다. 이 거리 계산의 복잡도는 $O(|X||Y|)$이다. <표 1>은 두 시퀀스 X = <4, 5, 6, 7, 6, 6>과 Y = <3, 4, 3>에 대한 누적 거리를 계산한 결과이다. 여기서 $T[|X|, |Y|] = T[6, 3] = 12$ 이므로 $D_{tw}(X, Y) = 12$이다.

<표 1> X = <4, 5, 6, 7, 6, 6> 과 Y = <3, 4, 3>에 대한 누적 거리 계산 테이블

6행	**6**	16	11	12
5행	**6**	13	9	10
4행	**7**	10	7	8
3행	**6**	6	4	5
2행	**5**	3	2	3
1행	**4**	1	1	2
	X/Y	**3**	**4**	**3**
		1열	2열	3열

2.2.3 다차원 타임 워핑 거리

다차원 타임 워핑 거리 함수를 제안하기 전에, k개 특성을 가지는 임의의 두 다차원 원소 $\boldsymbol{X}[i]$와 $Y[i]$에 대한 가중 거리 함수 D_{mbase}를 살펴보자.

$$D_{mbase}(\boldsymbol{X}[i], \boldsymbol{Y}[j]) = \sum_{h=1}^{k} W_h * |\boldsymbol{X}[i][h] - \boldsymbol{Y}[j][h]|$$

위에서 W_h는 h번째 특성의 가중치이다. 두 원소의 거리 함수로 D_{mbase}를 이용해서, 임의의 두 다차원 시퀀스 $\boldsymbol{X}$와 $\boldsymbol{Y}$의 거리를 아래와 같이 정의할 수 있다.

정의 2 : 임의의 다차원 시퀀스 $\boldsymbol{X}$와 $\boldsymbol{Y}$에 대해서, 이들 간의 다차원 타임 워핑 거리는 다음과 같이 정의된다.

$$\begin{aligned} &D_{mtw}(\langle\rangle, \langle\rangle) = 0 \\ &D_{mtw}(\boldsymbol{X}, \langle\rangle) = D_{mtw}(\langle\rangle, \boldsymbol{Y}) = \infty \\ &D_{mtw}(\boldsymbol{X}, \boldsymbol{Y}) = D_{mbase}(\boldsymbol{X}[1], \boldsymbol{Y}[1]) \\ &\qquad + \min \begin{cases} D_{mtw}(\boldsymbol{X}, \boldsymbol{Y}[2:-]) \\ D_{mtw}(\boldsymbol{X}[2:-], \boldsymbol{Y}) \\ D_{mtw}(\boldsymbol{X}[2:-], \boldsymbol{Y}[2:-]) \end{cases} \end{aligned}$$

여기서 $D_{mtw}(\boldsymbol{X}, \boldsymbol{Y})$는 순환 관계 기반의 동적 프로그래밍 기법[6]을 사용하여 $O(k|\boldsymbol{X}||\boldsymbol{Y}|)$의 복잡도로 계산될 수 있다.

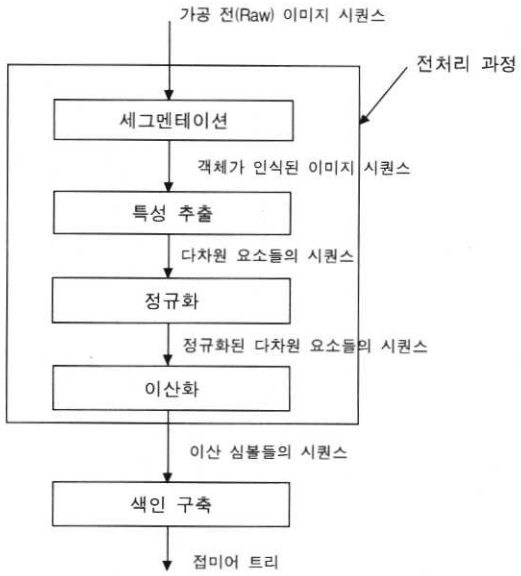
2.3 문제 정의

본 논문의 목표는 이미지 시퀀스 데이터베이스로부터 질의 시퀀스와 유사한 서브 시퀀스를 효과적으로 검색할 수 있는 색인 방법을 제안하는 것이다. 이 문제를 정형적으로 정의하면 아래와 같다.

문제 정의 : 이미지 질의 시퀀스 q와 거리 허용 오차(distance tolerance) ε를 가지고, 임의의 길이 이미지 데이터 시퀀스들의 집합에서 $D_{mtw}(x, q) \leq \varepsilon$를 만족하는 모든 데이터 서브시퀀스 x를 찾는다.

더 고려할 수 있는 질의의 유형에는 질의 시퀀스에 가장 근접한 k개의 이웃을 찾는 k-nearest-neighbor 질의, 서로 간의 거리가 주어진 거리 허용 오차 이내인 모든 시퀀스의 짝을 찾는 all-pairs 질의가 있다. 이와 같은 두 가지 유형의 질의들은 우리가 제안한 방법에 spatial join 알고리즘[9]과 branch-and-bound 알고리즘[19]을 적용하여 해결할 수 있다.

3. 색인 생성



(그림 1) 가공 전 이미지 시퀀스 집합으로부터 색인을 생성하는 과정

색인 단계는 이미지 시퀀스들로부터 색인을 생성한다. (그림 1)과 같이, 색인 단계는 전처리(pre-processing) 과정과 색인 구축 과정으로 크게 구분할 수 있다. 전처리 과정은 본래의(raw) 이미지 시퀀스들의 집합을 세그멘테이션(segmentation), 특성 추출(feature extraction), 정규화(normalization), 이산화(discretization) 과정을 거쳐 심볼 시퀀스의 집합으로 변환시킨다. 색인 구축 과정은 일련의 접미어 트리를 이진 병합함으로써, 심볼 시퀀스의 집합으로부터 디스크 기반의 접미어 트리(disk-based suffix tree)를 생성한다.

3.1 세그멘테이션과 특성 추출

세그멘테이션은 이미지의 배경으로부터 객체들의 경계를 검출하는 것이며, 특성 추출은 인식된 객체로부터 특성 벡터를 계산하는 것이다. 예를 들어, 종양을 가지고 있는 뇌의 이미지로부터 종양의 위치(location), 크기(size), 둘레값(perimeter)을 추출할 수 있다. 이 경우 이미지 시퀀스 **X**는 <(location[1], size[1], perimeter[1]), …, (location[n], size[n], perimeter[n])>로 표기할 수 있다. 여기서 n은 **X**를 구성하는 이미지의 수이다. 세그멘테이션과 특성 추출에 대한 자세한 내용은 본 논문의 범위를 벗어나므로 필요시 참고 문헌[10, 11, 12, 36]을 참조하면 된다.

3.2 정규화

정규화의 목적은 사용자가 특정한 차원의 상대적 가중치를 할당하거나 조절하는 것을 쉽게 하기 위해서이다. 정규화 과정이 없으면, 높은 평균값을 갖는 특성 차원은 낮은 평균값을 갖는 다른 특성 차원에 비해 시퀀스들의 유사성을 판단하는데 더 많은 영향을 미치게 된다. 그러므로 모든 특성 차원이 유사성 결정에 고르게 영향을 미칠 수 있도록 정규화 과정을 통해 데이터 분포가 정규 분포(normal distribution)를 따르도록 만들어 준다.

3.3 이산화

이산화 과정은 색인 구조를 압축하여 저장 공간을 줄이고 질의 처리 성능을 향상시키다. 이산화를 위해 먼저 정규화된 다차원 원소들을 여러 개의 카테고리 혹은 그룹으로 분리한 후 각각의 그룹에 유일한 심볼을 할당한다. 널리 알려진 다차원 클러스터링 방법 중에서 우리는 다중 속성 타입 추상 계층(multiple-attribute type abstraction hierarchy, MTAH)[16] 분류법을 사용한다. MTAH는 확장 오류(relaxation error)를 최소화 하도록 다차원 그룹의 경계 값을 결정하는 분류법으로 다음과 같은 장점이 있다[16]. 첫째, 데이터의 값과 발생 빈도 분포를 동시에 고려하므로 등간격 분류법(equal-length categorization) 보다 더 정확한 결과를 생성할 수 있다. 둘째, 최대 엔트로피 분류법(maximum entropy categorization)에 비해 알고리즘이 간단하므로 구현이 수월하다. 그러나 이산화 과정이 특정 분류법에 의존하지 않는다는 것에 유의해야 한다. 즉 MTAH 대신에 등간격 분류법이나 최대 엔트로피 분류법을 사용해도 다른 과정에는 전혀 영향을 미치지 않는다. k차원을 갖는 원소들로부터 생성된 카테고리는 $C = ([C[1].min, C[1].max], [C[2].min, C[2].max], \cdots, [C[k].min, C[k].max])$로 표현된다. 카테고리 집

합을 생성한 후에, 각각의 원소들을 해당하는 카테고리의 심볼로 변환한다. 기호 X^C는 X로부터 이산화 과정을 통해 변환된 심볼 시퀀스를 나타낸다.

3.4 색인 구축

우리는 효과적인 서브시퀀스 매칭을 위해 접미어 트리를 색인 구조로 사용한다. 접미어 트리는 다음과 같은 장점이 있다. 첫째, 주어진 시퀀스의 모든 접미어를 트리 안에 저장하기 때문에 서브시퀀스 매칭에 아주 적합한 구조이다. 둘째, 트리의 구조가 삼각 부등식을 가정하고 있지 않기 때문에 타임 워핑 거리 함수를 이용해 유사도를 측정하는 경우에도 잘못된 누락을 발생시키지 않는다.

접미어 트리를 좀 더 자세히 살펴보자. 트라이(trie)는 키워드 집합에서 원하는 키워드를 신속하게 찾을 수 있는 색인 구조이다. 접미어 트라이(suffix trie)는 키워드 집합이 시퀀스의 접미어로 구성된 것이다[38]. 접미어 트라이를 구성하는 노드 중에서 하나의 자식만을 가지는 것들을 병합하면 접미어 트리가 얻어진다.

각 시퀀스의 접미어는 단말 노드(leaf node)로 표현된다. 좀 더 명확하게 말하자면, $X[i:-]$는 $(ID(X), i)$를 저장하는 단말 노드에 의해 표현된다. 여기서 $ID(X)$는 X의 식별자이고 i는 시퀀스 상에서 접미어가 시작되는 위치를 나타낸다. 루트 노드와 단말 노드를 연결하는 경로 상의 모든 값들을 연결하면 그 단말 노드가 표현하는 접미어를 얻게 된다. 루트 노드와 내부 노드 N을 연결하는 경로 상의 모든 값을 연결하면 N아래의 단말 노드들이 나타내는 접미어들의 가장 긴 공통 접두어(prefix)가 된다. 우리는 노드 N_1과 N_2를 연결하는 경로 상의 모든 값들을 연결한 것을 $label(N_1, N_2)$로 나타낸다.

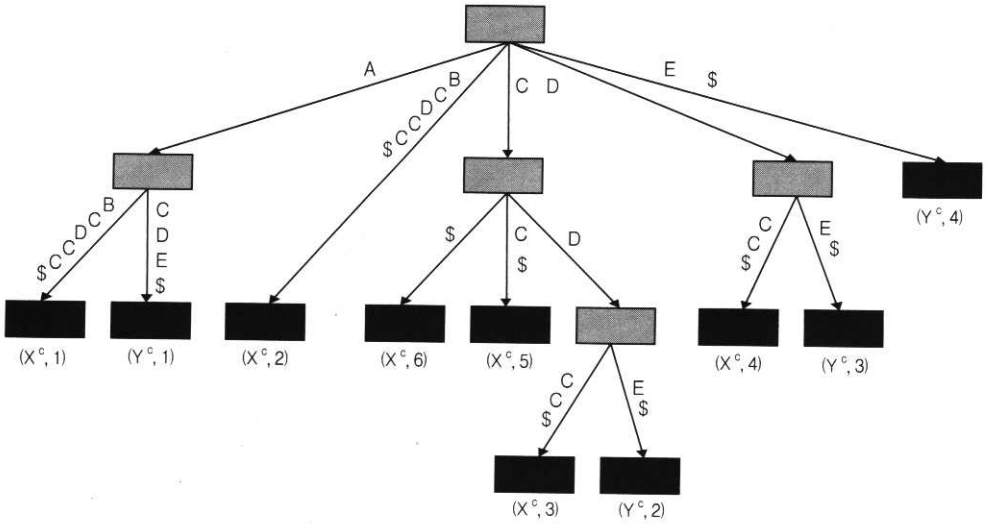
다중 시퀀스로부터 접미어 트리를 생성하기 위해서, 우리는 디스크 기반의 점진적 접미어 트리 생성(disk-based incremental suffix tree construction) 기법[7]을 사용한다. 즉, 서로 다른 시퀀스 집합으로부터 생성된 두 개의 접미어 트리를 병합하기 위하여 두 개의 트리를 동시에 선위 회하면서 공통된 서브시퀀스에 해당하는 경로를 결합한다. 평균 길이가 $\overline{L}$인 m개의 데이터 시퀀스로부터 접미어 트리를 구성하기 위한 알고리즘의 복잡도는 $O(m\overline{L})$이다.

(그림 2)는 두 개의 심볼 시퀀스, $X^C = \langle A, B, C, D, C, C \rangle$, $Y^C = \langle A, C, D, E \rangle$로부터 구성된 접미어 트리를 보여준다. $ 기호는 접미어의 마지막을 나타내는 표시자이다.

4. 질의 처리

제출된 질의 이미지 시퀀스는 세그멘테이션과 특성 추출에 의해서 다차원 시퀀스 q로 변환된다. 다음 단계로 질의 시퀀스와의 하한 거리(lower-bound distance)가 허용 오차 ε 이내인 후보 서브시퀀스들을 검색하기 위해 접미어 트리를 루트 노드부터 순회한다.

하한 타임 워핑 거리가 필터링에 사용되므로 실제 타임 워핑 거리가 ε보다 큰 서브시퀀스들이 후보 집합에 포함될 가능성이 있는데, 이러한 서브시퀀스들을 잘못된 경보(false alarms)라고 한다[2, 3]. 잘못된 경보를 제거하기 위해 인덱스 검색 후, 후처리(post-processing) 작업을 수행한다. 즉, 데이터베이스에서 해당 서브시퀀스를 추출한 후 D_{tw}를 사



(그림 2) X^C = ⟨A, B, C, D, C, C⟩, Y^C = ⟨A, C, D, E⟩로부터 생성된 접미어 트리. X^C로부터의 (⟨A, B, C, D, C, C⟩, ⟨B, C, D, C, C⟩, ⟨C, D, C, C⟩, ⟨D, C, C⟩, ⟨C, C⟩, ⟨C⟩) 6개 접미어와 Y^C로부터의 (⟨A, C, D, E⟩, ⟨C, D, E⟩, ⟨D, E⟩, ⟨E⟩) 4개의 접미어가 추출되어 접미어 트리에 삽입됨. $는 접미어의 끝을 표시하는 기호이다.

용해서 그들의 타임 워핑 거리를 계산하고, 마지막으로 실제 타임 워핑 거리가 ε보다 크지 않은 서브시퀀스들을 최종 결과로 반환하게 된다.

4.1 색인 순회 알고리즘

본 논문에서 제안하는 색인 순회 과정이 (알고리즘 1)에 있다. 이 알고리즘은 루트(root)부터 검색을 시작하여 각 노드들을 깊이 우선(depth-first)으로 방문한다. CN_i를 노드 N의 i번째 자식 노드라고 하자. 알고리즘이 노드 N을 방문한다면, 새로운 후보를 찾기 위해 각각의 자식 노드 CN_i를 조사한다. 즉, 루트로부터 CN_i를 연결하는 경로가 새로운 후보들을 생성할 수 있는지를 조사하기 위해, 알고리즘은 *label*(*root*, CN_i)와 $\boldsymbol{q}$사이의 하한 타임 워핑 거리 $D_{mtw\text{-}lb}$을 구한다. $D_{mtw\text{-}lb}$는 다음 절에서 정의된다.

알고리즘을 쉽게 설명하기 위해서, 모든 에지가 하나의 값만 저장하고 있다고 가정하자. *label*(*root*, CN_i)와 $\boldsymbol{q}$사이의 하한 타임 워핑 거리를 계산하기 위해서 동적 프로그래밍 기법을 적용한다. 즉, X축 상에는 $\boldsymbol{q}$를, Y축 상에는 *label*(*root*, CN_i)를 위치시킨 상태에서 누적 거리 테이블을 작성한다. N이 루트라면, 누적 거리 테이블은 최하층(bottom)부터 만들어진다. 그 외의 경우에는, 함수 AddRow(T, $\boldsymbol{q}$, *label*(N, CN_i), $D_{mtw\text{-}lb}$)를 호출하여, 루트부터 N까지 축적되어온 기존의 테이블 T위에 *label*(N, CN_i)에 해당하는 새로운 행(row)을 추가함으로써 누적 거리 테이블을 구축한다. 새로 추가되는 행의 가장 오른쪽 열의 값이 허용 거리 오차 ε이하이면 *label*(*root*, CN_i)을 후보 집합에 추가시킨다.

알고리즘은 재귀적(recursive)으로 호출되므로 CN_i가 조사된 후에는 CN_i의 자식 노드들이 방문된다. 하지만 탐색 공간을 줄이기 위해서는 방문의 필요성이 입증된 자식 노드만을 방문해야 한다. 이를 위해 누적 거리 테이블의 마지막 행을 점검한다. 마지막 행에 저장된 값 중에서 ε보다 작거나 같은 것이 하나라도 있으면 CN_i의 자식 노드들을 방문한다. 그렇지 않으면 N의 다음 자식인 CN_{i+1}을 방문한다. 이 가지치기 방법(branch-pruning method)은 다음 정리 1을 기반으로 한다.

정리 1 : 누적 거리 테이블의 최상위 행의 모든 열들이 거리 허용 오차 ε보다 큰 값을 가진다면, 이 테이블에 새로운 행들을 추가하더라도 ε이하의 새로운 값을 생성할 수 없다.

증명 : 참고문헌[27] 참조

Input : node N, query sequence $\boldsymbol{q}$, distance tolerance ε, cumulative distance table $\boldsymbol{T}$
Output : candidateSet

```
candidateSet ← { } ;
CN ← GetChildren (N) ;
for i ← 1 to |CN| do
    CT_i ← AddRow(T, q, label (N_i, CN_i), D_mtw-lb ) ;
    Let dist be the value in the rightmost column of newly
      added row ;
    Lit minDist be the minimum value in the newly added
      row ;
    if dist ≤ ε then
      candidateSet ← candidateSet ∪ {label (root, CN_i )} ;
    if minDist ≤ ε then
      candidateSet ← candidateSet ∪ IndexTraveral (CN_i, q, ε,
          CT_i ) ;
loop
return candidateSet
```

(알고리즘 1) IndexTraversal Algorithm

4.2 하한 타임 워핑 거리 함수

접미어 트리의 모든 에지들은 이산화 과정을 통해서 얻어진 심볼들을 저장하고 있기 때문에, 접미어 트리 안에 포함된 시퀀스와 질의 시퀀스 간의 정확한 다차원 타임 워핑 거리는 계산할 수 없다. 그러므로 우리는 D_{mtw}의 하한 거리를 반환하는 새로운 거리 함수 $D_{mtw\text{-}lb}$를 도입한다.

정의 3 : 주어진 두 개의 다차원 서브시퀀스 $\boldsymbol{x}$와 $\boldsymbol{y}$에서, $D_{mtw}(\boldsymbol{x}, \boldsymbol{y})$의 하한 거리를 계산하는 $D_{mtw\text{-}lb}(\boldsymbol{x}^C, \boldsymbol{y})$를 다음과 같이 정의한다.

$$
\begin{aligned}
D_{mtw-lb}(\langle\rangle, \langle\rangle) &= 0 \\
D_{mtw-lb}(\boldsymbol{x}^c, \langle\rangle) &= D_{mtw-lb}(\langle\rangle, \boldsymbol{y}) = \infty \\
D_{mtw-lb}(\boldsymbol{x}, \boldsymbol{y}) &= D_{mbase-lb}(\boldsymbol{x}^C[1], \boldsymbol{y}[1]) \\
&\quad + \min \begin{cases} D_{mtw-lb}(\boldsymbol{x}^c, \boldsymbol{y}[2:-]) \\ D_{mtw-lb}(\boldsymbol{x}^c[2:-], \boldsymbol{y}) \\ D_{mtw-lb}(\boldsymbol{x}^c[2:-], \boldsymbol{y}[2:-]) \end{cases}
\end{aligned}
$$

$$
D_{mbase-lb}(C, \boldsymbol{y}[1]) = \sum_{h=1}^{k} W_h * D_{base-lb}(C[h], \boldsymbol{y}[1][h])
$$

$$
D_{base-lb}(C[h], \boldsymbol{y}[1][h])
$$

$$
= \begin{cases} 0 & \text{if } C[h].\min \le \boldsymbol{y}[1][h] \le C[h].\max \\ C[h].\min - \boldsymbol{y}[1][h] & \text{if } \boldsymbol{y}[1][h] < C[h].\min \\ \boldsymbol{y}[1][h] - C[h].\max & \text{if } \boldsymbol{y}[1][h] > C[h].\max \end{cases}
$$

여기서 $\boldsymbol{x}^C$는 이산화 과정을 통해 $\boldsymbol{x}$로부터 얻은 심볼 시퀀스이고, C는 $\boldsymbol{X}^C$의 첫 번째 원소, 즉 $\boldsymbol{X}^C[1]$이며, $C[h]$는 심볼 C로 표현되는 다차원 원소들이 h번째 특성 차원으로 가지는 값의 범위, 즉 최대값과 최소값을 나타낸다.

임의의 두 다차원 원소들에 대해 언제나 $D_{mbase\text{-}lb}$가 D_{mbase}보다 작거나 같은 값을 반환하는 것은 명백하므로, 임의의 두 다차원 원소 시퀀스들에 대해 언제나 $D_{mtw\text{-}lb}$가 D_{mtw}보

다 크지 않는 값을 반환하는 것은 당연하다. 본 논문에서 제안하는 질의 처리 알고리즘은 유사하지 않는 서브시퀀스를 여과하기 위해 $D_{mtw\text{-}lb}$를 이용한다. 다차원 데이터 시퀀스와 질의 시퀀스간의 다차원 타임 워핑 거리 D_{mtw}가 ε보다 크지 않을 때, 그들의 하한 타임 워핑 함수 거리 $D_{mtw\text{-}lb}$는 확실히 ε이하이다. 이것은 질의 시퀀스로부터 ε이내의 모든 데이터 시퀀스들이 후보 집합에 확실히 포함됨을 의미한다.

4.3 알고리즘 분석

naïve한 방법은 각각의 이미지 시퀀스를 데이터베이스로부터 읽고 시퀀스에 포함된 접미어의 수만큼 누적 거리 테이블을 만든다. 각각의 이미지로부터 추출한 특성의 수를 k라고 할 때, 질의 이미지 시퀀스 $\boldsymbol{q}$와 길이가 L인 접미어에 대한 누적 거리 테이블을 생성하는 알고리즘의 복잡도는 $O(kL|q|)$이다. 평균 길이가 $\overline{L}$인 m개의 데이터 시퀀스 내에는 $m\overline{L}$개의 접미어가 존재하고 그들의 평균 길이는 $\frac{\overline{L}+1}{2}$이다. 그러므로 naïve한 방법의 복잡도는 $O(km\overline{L}^2|q|)$로 표현될 수 있다.

논문에서 제안된 알고리즘은 naïve한 방법보다 적은 계산량을 가진다. 그 이유는 ① 제안된 가지치기(branch-pruning) 방법이 탐색 공간을 줄이고, ② 공통 접두어(prefix)를 가진 접미어들이 색인 순회 과정에서 누적 거리 테이블을 공유하기 때문이다. $R_p(\geq 1)$는 가지치기로부터 인해 얻어지는 감소 계수(reduction factor), $R_d(\geq 1)$를 누적 거리 테이블의 공유로 인한 감소 계수로 표현하면, 제안된 알고리즘은 $O(\frac{km\overline{L^2}|\boldsymbol{q}|}{R_d R_p}+kn\overline{L}\boldsymbol{q})$의 복잡도를 가지게 된다. 여기서 n은 후처리를 요구하는 서브시퀀스의 수를 나타낸다. 그러므로 복잡도 계산식의 왼쪽 항은 색인 순회 비용, 오른쪽 항은 후처리 비용을 의미하게 된다.

접미어 트리에 저장된 공통 에지의 수와 길이가 증가함에 따라 감소 계수 R_d는 점차 커지게 된다. 처음 t개의 원소가 같은 h개의 접미어들 $s_1, \cdots s_h$에서, h개의 누적 거리 테이블을 구성하려면 $|s_1||q|+\cdots+|s_h||q|$개의 셀(cell) 계산을 필요로 한다. 그러나 $\boldsymbol{q}$와 길이 t인 공통 접두어로부터 구축된 누적 거리 테이블을 h개의 접미어가 공유한다면 이것은 $t|\boldsymbol{q}|+(|s_1|-t)|\boldsymbol{q}|+\cdots+(|s_h|-t)|\boldsymbol{q}|$로 감소될 수 있다. 이 경우 R_d는 다음과 같이 표현될 수 있다.

$$R_d=\frac{|s_1|+\cdots+|s_h|}{(|s_1|+\cdots+|s_h|)-(h-1)t}$$

R_d가 시퀀스에 포함된 원소값의 분포에 의해 결정되는 데 비해서, R_p는 사용자가 요청하는 거리 허용 오차 ε에 의해 결정된다. ε이 매우 작아서 단지 한 개나 두 개의 서브시퀀스가 유사 시퀀스로 판명될 수 있다면, 질의 처리 과정에서 색인의 최상위 부분만을 방문할 것이다. 반대로 ε이 너무 커서 모든 서브시퀀스가 유사 시퀀스가 될 수 있다면 색인의 모든 노드들이 방문되어 $R_p=1$이 된다.

5. 실험 결과

본 논문에서 제안한 색인 방법을 C++ 프로그래밍으로 구현하여 실제 데이터 집합과 합성 데이터 집합에 적용한 결과를 성능(performance)과 확장성(scalability) 측면에서 분석한다.

5.1 데이터 집합(Data Set)

20명의 환자에 대하여 각기 다른 시간에 찍은 3개의 폐 이미지로 이미지 시퀀스들의 집합을 얻는다. 이것을 기반으로 확장된 데이터베이스를 얻기 위해 각 폐 이미지를 96개의 서브 영역으로 나누어 각 환자 당 길이 3의 96개의 서브 시퀀스를 만든다. 즉 20명 환자의 이미지 시퀀스들은 각 시퀀스 당 3개의 이미지를 가지는 1920개의 시퀀스로 변환된다. 변환된 시퀀스의 각 이미지로부터 다음 7개의 특성을 추출한다. ① percentage of voxels ② mean of gray level ③ standard deviation of gray level ④ median of gray level ⑤ tenth centile ⑥ first measure of correlation ⑦ horizontal edge

확장성 테스트를 위해서 대용량 합성 이미지 시퀀스의 집합을 생성한다. 각 특성 차원의 값의 생성식은 랜덤 워크(random walk)를 채택하였다. 즉 j번째 특성 차원의 값은 식 $X[i][j]=X[i-1][j]+Z_i$인데, 여기서 $Z_i(i=1,2,\cdots)$는 [1, 100] 범위의 독립적 동일 분포의 랜덤 변수이다. 합성 이미지 시퀀스들의 개수와 평균 길이는 각 확장성 테스트의 목적에 따라 결정하였다.

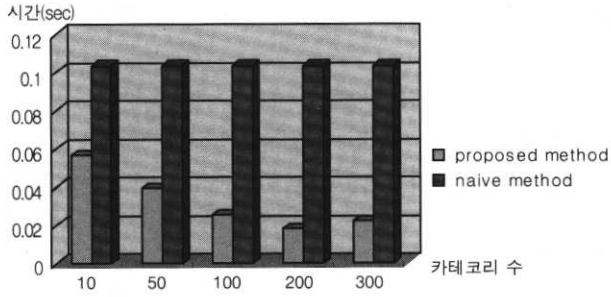
5.2 시스템 환경

본 연구의 실험을 위한 하드웨어 플랫폼은 512KB 캐쉬, 512MB RAM과 RPM이 7200이고, 평균 탐색 시간이 9ms인 80GB 하드디스크를 장착한 Pentium Ⅳ 2.0GHz 시스템이고, 소프트웨어 플랫폼은 Windows XP professional이다.

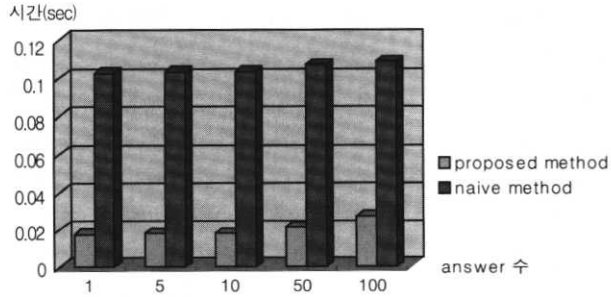
5.3 성능 테스트

본 논문에서 제안한 방법과 naïve한 방법의 성능을 여러 조건에서 비교해 보았다. naïve한 방법은 데이터베이스로부터 각 이미지 시퀀스들을 순차적으로 읽은 후, 동적 프로그래밍 기법을 이용해서 질의 시퀀스와의 거리를 계산한다.

첫 번째 실험은 이산화 과정에 사용되는 카테고리의 수를 증가시키면서 100개의 질의에 대한 평균 질의 처리 시간을 측정하는 것이다. 거리 허용 오차는 검색 결과가 5개의 서브시퀀스를 포함하도록 조정하였다. (그림 3)은 첫 번째 실험에 대한 결과를 보여준다. naïve한 방법은 이산화 과정과 무관하므로 일정한 성능을 나타내었다. 그러나 본 논문에서 제안한 방법은 카테고리의 수가 증가함에 따라 전반적으로 더 나은 성능을 보였다. 이것은 카테고리의 수가 증가함에 따라 하한 거리 함수가 원래 거리 함수에 근접하게 되어 잘못된 경고의 수가 감소하기 때문이다. 그런데 카테고리의 수가 너무 많아지면 색인의 크기가 매우 커져서 성능이 저하되는 것을 관찰할 수 있다.



(그림 3) 이산화 과정에 사용되는 카테고리의 수에 따른 평균 질의 처리 시간. 1920개의 이미지 시퀀스를 사용했고, 각각은 7개의 특성을 가지는 3개의 원소로 구성. 질의 시퀀스의 평균 길이는 3이고 거리 허용 오차는 결과 집합이 5개의 서브시퀀스를 포함하도록 조정



(그림 4) 결과 집합의 크기 증가에 다른 평균 질의 처리 시간. 1920개의 이미지 데이터 시퀀스를 사용했고, 그 각각은 7개의 특징을 가지는 3개의 원소로 구성. 질의 시퀀스의 평균 길이는 3이고 카테고리의 수는 200개

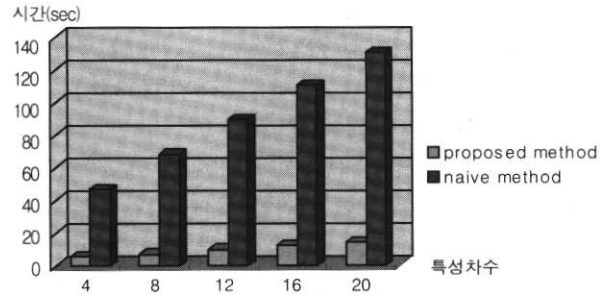
두 번째 실험에서는 결과 집합의 크기를 증가시키면서 두 방법의 성능을 비교하였다. 논문에서 제안된 방법은 200개의 카테고리를 사용하여 이산화를 수행하였다. (그림 4)의 결과처럼, 결과 집합의 수가 증가함에 따라 두 방법 모두 약간의 성능 저하가 발생한다. 결과 집합에 많은 서브시퀀스를 포함하기 위해서는 거리 허용 오차를 증가시켜야 하는데, 이것은 곧 탐색 공간의 확장을 초래하여 성능상의 이점을 감소시킨다. 그러나 보편적으로 사용자들은 적은 개수의 높은 순위를 갖는 결과에 관심이 있기 때문에 이것은 큰 문제가 아니다.

5.4 확장성 실험

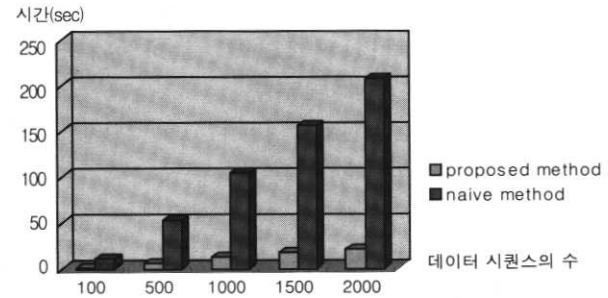
확장성 실험을 위해 합성 데이터 집합을 사용했다. 이 확장성 테스트에 이용되는 파라미터는 ① k(특성 차원의 개수) ② m(데이터 이미지 시퀀스의 개수) ③ $\overline{L}$(데이터 이미지 시퀀스의 평균 길이) ④ $|\boldsymbol{q}|$(질의 시퀀스의 평균 길이)이다. 본 논문에서 행한 모든 확장성 실험에서, 카테고리의 수는 100, 거리 허용 오차는 총 데이터 서브시퀀스의 10^{-3}%개가 결과 집합에 포함되도록 설정하였다.

첫 번째 확장성 실험은 특성 차원의 개수 k를 4에서 20까지 증가시키며 평균 질의 처리시간을 비교하는 것이다. (그림 5)의 결과와 같이, 두 방법 모두 질의 처리시간이 특성 차원의 수가 증가함에 따라 선형적으로 증가한다. 그러나 제안된 방법이 보다 낮은 시간 증가율을 보이는 것에 주목할 필요가 있다.

두 번째 확장성 실험은 데이터 이미지 시퀀스의 수를 100에서 2000까지 증가시키며 두 방법의 성능을 비교하는 것이다. (그림 6)을 보면, 두 방법 모두 질의 처리시간이 데이터 시퀀스 수의 증가에 따라 선형적으로 증가하지만 성능 개선 비율은 거의 일정하게 유지됨을 알 수 있다.

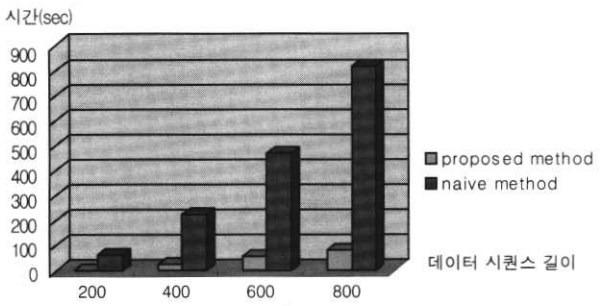


(그림 5) 특성 차수의 변화에 따른 평균 질의 처리 시간 (m = 500, $\overline{L}$ = 200, $|\boldsymbol{q}|$ = 20)

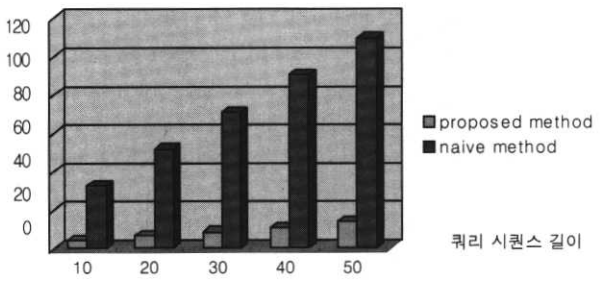


(그림 6) 데이터 시퀀스의 개수 변화에 따른 평균 질의 처리 시간(k = 5, $\overline{L}$ = 200, $|q|$ = 20)

세 번째 확장성 실험은 데이터 이미지 시퀀스의 길이를 200에서 800까지 증가시키면서 두 방법의 성능을 비교하는 것이다. (그림 7)과 같이, 두 방법의 질의 처리 속도는 모두 데이터 시퀀스의 길이에 따라 2차 함수적으로 증가한다. 그러나 성능 개선 비율은 거의 일정함을 보인다.



(그림 7) 데이터 시퀀스의 평균 길이 변화에 따른 평균 질의 처리 시간(k = 5, m = 500, $|q|$ = 20)



(그림 8) 질의 시퀀스의 평균 길이의 변화에 따른 평균 질의 처리 시간(k = 5, m = 500, $\overline{L}$ = 200)

마지막 확장성 실험은 질의 이미지 시퀀스의 길이를 10에서 50까지 증가시키면서 두 방법을 비교하는 것이다. (그림 8)과 같이, 질의 시퀀스들의 길이에 따라 두 방법 모두 선형적으로 증가하지만 성능 개선비율은 거의 일정함을 보인다.

6. 관련 연구

6.1 일차원 시퀀스에 대한 연구

그동안 일차원 시퀀스의 유사 검색에 대해 많은 연구가 있었다. 대표적인 예로 전체 시퀀스 매칭을 수행하는 F-Index[2]가 있다. 이것은 먼저 시퀀스들을 이산 푸리에 변환(Discrete Fourier Transform)을 이용해 다차원 상의 점들로 변환한 후 R^*-tree에 저장한다. 질의 시퀀스가 주어지면 이것을 다차원 상의 점으로 변환한 후 이 점으로부터 일정한 거리 이내에 위치하는 점들을 찾아서 후보 집합에 포함시킨다. 참고문헌[18]에서는 F-Index를 서브 시퀀스 매칭을 위해 사용하였다. 이러한 방법들은 유클리드 거리를 유사성 기준으로 사용하기 때문에 길이가 일치하지 않거나 샘플링 비율이 가변적인 시퀀스에는 적합하지 않다.

모양을 기반으로 유사 검색을 수행하기 위해 정규형(normal form)을 이용하는 연구가 참고문헌[20]에서 소개되었다. 이 연구는 스케일링이나 쉬프팅과 같은 변환이 수행되어도 시퀀스의 정규형이 변경되지 않는다는 성질을 이용하였다. 그러나 시퀀스가 시간 축으로 압축되거나 팽창되는 경우에는 정규형을 적용할 수 없다. 시퀀스 변형을 질의 언어에 포함시키는 것을 허용하는 연구가 참고문헌[32]에 소개되었다. 이 연구는 시퀀스 변형 하에서도 한 번 구축된 R-tree 인덱스를 변경 없이 사용할 수 있는 방안을 제안하였다. 제안된 방법은 이동 평균이나 전역 타임 스케일링(global time scaling)에는 잘 적용되지만 타임 워핑에는 적용될 수 없는 단점이 있다. 참고문헌[4]에서는 모양 정의 언어(shape definition language, SDL)와, SDL 질의의 실행 속도를 높일 수 있는 색인 구조가 제안되었고, 참고문헌[34]에서는 데이터의 일반적인 모양을 지정할 수 있는 일반화된 근사 질의(generalized approximate queries)의 개념이 제안되었다.

최근의 연구들은 길이가 다른 시퀀스의 매칭을 허용한다. 원소의 상당 부분이 매치되면 두 시퀀스가 유사하다고 판단하는 수정된 버전의 에디트 거리(edit distance)가 참고문헌[8]에서 제안되었고, FastMap 색인 필터와 하한 거리 필터(lower-bound distance filter)를 동시에 사용하는 기법이 참고문헌[42]에서 제시되었다. 이 두 방법의 근본적인 색인 구조들은 삼각 부등식을 기본으로 한다. 또 다른 연구로는 데이터베이스로부터 순차적으로 데이터 시퀀스를 읽고, 그것을 선형 세그먼트의 순차 리스트로 변환하여, 수정된 타임 워핑 거리의 측정에 적용하는 것이 있다[22].

타임 워핑 거리를 지원하는 색인 방법에 관한 연구가 최근에 활발하게 행해졌다. 길이가 긴 시퀀스를 위한 세그먼트 기반 서브시퀀스 검색 방법이 참고문헌[30]에서 제안되었고, 삼각 부등식을 만족하는 L_∞ 기반의 하한 타임 워핑 거리가 참고문헌[23]에서 제시되었다. 제시된 하한 타임 워핑 거리는 참고문헌[23]에서는 전체 시퀀스 매칭에, 참고문헌[29]에서는 서브시퀀스 매칭에 이용되었다.

생물학적 시퀀스(biological sequences)의 매칭에 대한 몇 개의 연구가 있었는데, 참고문헌[7]에서는 시퀀스 정렬 문제(sequence alignment problem)를 해결하기 위해 디스크 기반의 접미어 트리가 이용하였고, 참고문헌[40]에서는 스트링 에디트 거리(string edit distance)을 사용하여 단백질의 유사성을 측정하였다.

6.2 다차원 시퀀스에 대한 연구

수정된 에디트 거리(edit distance)를 이용하여 다차원 시퀀스의 유사성을 측정한 방법이 참고문헌[41]에 소개되었다. 이 방법은 전체 시퀀스 매칭에 관심을 집중하였고, 삼각 부등식에 기반을둔 색인 구조를 이용하였다. 따라서 유사성 기준으로 타임 워핑 거리를 사용하면 잘못된 누락이 발생할 가능성이 높다.

비디오 데이터베이스의 유사성 검색에 대한 연구가 수행되었는데, 비디오는 프레임들의 연속이므로 다차원 시퀀스의 특별한 예로 간주될 수 있다. 대부분의 비디오 색인 기법들은 naïve한 방법에 의존하여 데이터베이스에서 유사한 비디오를 검색하였다. 참고문헌[26]에서는 동작 유사성(ac-

tion similarity)을, 참고문헌[39]에서는 이미지 내용과 움직임을 결합하여 유사 검색을 수행하였다. 참고문헌[24, 33]에서는 색 결합 벡터와 색 히스토그램을 이용하여 상업용 비디오 클립의 유사 검색을 수행하였다. 비디오 시퀀스들을 위해 *vstring* 표현을 도입하고 유사성 기준으로 *vstring* 에디트 거리를 제안한 연구[1], 한 개의 MPEG 모션 벡터를 16×16 크기의 서브 이미지로부터 추출한 연구[5], 카메라의 줌과 팬(pan)과 같은 낮은 수준의 모션 특성을 사용한 연구[25] 등이 행해졌다. 위에서 언급한 모든 비디오 색인 기법들은 naïve한 방법을 이용하기 때문에, 데이터베이스가 커지면 성능이 저하되는 단점이 있다.

역 파일(inverted file) 기법을 이미지 검색에 사용한 연구[37]와 미디어 추적(tracking)에 활용한 연구[21]도 행해졌다. 두 방법 모두 질의를 처리하는 동안 역 파일을 주 기억장치에 보관해야 하므로 대 용량 비디오 데이터베이스에는 적당하지 않다.

7. 결 론

이미지 시퀀스 데이터베이스는 이미지 시퀀스들의 집합으로 그 원소들은 이미지들의 순차 리스트이다. 유사 검색은 주어진 질의 시퀀스와 비슷한 변화 패턴을 가지고 있는 시퀀스나 서브시퀀스를 찾는 연산으로 데이터 마이닝과 데이터 웨어하우징, 디지털 라이브러리 등과 같은 많은 응용에서 그 중요성이 점차 강조되고 있다. 특히 메디컬 영역에서, 어떤 환자의 현재 증상이나 특징을 가지고 과거의 유사한 특징을 갖는 의학적 정보를 참고할 수 있는 것은 의사들의 환자 치료에 큰 도움을 준다.

본 논문에서는 주어진 이미지 시퀀스와 유사한 서브시퀀스를 데이터베이스로부터 잘못된 누락 없이 효과적으로 검색하기 위해 디스크-기반 접미어 트리를 색인으로 사용하는 것을 제안했다. 이미지 시퀀스들은 길이가 다르거나 샘플링 비율이 다를 수 있기 때문에, 본 논문에서 제안한 방법은 다차원 타임 워핑 거리를 유사성 기준으로 사용했다. 다차원 타임 워핑은 다차원 시퀀스들이 시간 축으로 확장되는 것을 허용함으로써 그들 간의 거리를 최소화 시킨다. 색인의 크기를 줄이고 질의 처리 속도를 높이기 위해 다차원 이산화 과정을 도입하였으며 유사하지 않은 서브 시퀀스를 잘못된 누락 없이 여과하기 위해 하한 거리 함수를 사용하였다. 메디컬 이미지와 합성 이미지의 시퀀스들에 대한 실험을 통해 본 논문에서 제안된 방법이 naïve한 방법보다 성능이 우수하며 대용량 이미지 시퀀스 데이터베이스에서도 잘 적용됨을 보였다.

논문에서 제안된 방법은 다양한 프레임 비율을 가진 비디오 데이터베이스에 쉽게 적용될 수 있다. 예로서 불법으로 복제된 비디오를 인터넷상에서 찾는 응용을 생각해보자. 타임 워핑을 고려하지 않고 비디오간의 유사성을 측정하려고 한다면 고의로 프레임 비율을 변경시킨 불법 복제 비디오를 검출하기 어려울 것이다. 또한 인터넷상에는 수많은 비디오 파일이 존재하기 때문에 순차적인 검색 방법을 사용한다면 너무 오랜 시간이 소요될 것이다. 그러나 논문에서 제안된 거리 함수와 색인 방법을 사용한다면 유사한 비디오를 정확하고 신속하게 검색할 수 있다.

참 고 문 헌

[1] D. A. Adjeroh, M. C. Lee and I. King, A distance measure for video sequence similarity matching, In Proc. Int'l Work hop on Multi-Media Database Management Systems, 1998.

[2] R. Agrawal, C. Faloutsos and A. Swami, Efficient similarity search in sequence databases, In Proc. Int'l Conf. on Foundations of Data Organization and Algorithms (FODO), pp.69-84, 1993.

[3] R. Agrawal, K. Lin, H. S. Sawhney and K. Shim, Fast similarity search in the presence of noise, scaling, and translation in time-series databases, In Proc. Int'l Conf. on Very Large Data Bases (VLDB), pp.490-501, 1995.

[4] R. Agrawal, G. Psaila, E. L. Wimmers and M. Zait, Querying shapes of histories, In Proc. Int'l Conf. on Very Large Data Bases (VLDB), pp.502-514, 1995.

[5] E. Ardizzone, M. L. Cascia, A. Avanzato, and A. Bruna, Video indexing using MPEG motion compensation vectors. In Proc. IEEE Int'l Conf. on Multimedia Computing System, pp.490-501, 1999.

[6] D. J. Berndt and J. Cliford, Finding patterns in time series : A dynamic programming approach, In Advances in Knowledge Discovery and Data Mining, AAAI/MIT, pp.229-248, 1996.

[7] P. Bieganski, J. Riedl and J. V. Carlis, Generalized suffix trees for biological sequence data : Applications and implementation, In Proc. Hawaii Int'l Conf. on System Sciences, 1994.

[8] T. Bozkaya, N. Yazdani and M. Ozsoyoglu, Matching and indexing sequences of different lengths, In Proc. ACM Int'l Conf. on Information and Knowledge Management (CIKM), pp.128-135, 1997.

[9] T. Brinkho, H.-P. Kriegel, R. Schneider and B. Seeger, Multi-step processing of spatial joins, In Proc. ACM Int'l Conf. on Management of Data (SIGMOD), pp.237-246, 1994.

[10] M. S. Brown, J. G. Goldin, M. F. McNitt-Gray, L. E.Greaser, A. Sapra, K. T. Li, J. W. Sayre, M. Martin and D. R. Aberle Knowledge-based segmentation of thoracic CT images for assessment of split lung function, In Proc. Medical Physics, 2000.

[11] M. S. Brown, M. F. McNitt-Gray, J. G. Goldin, L. E. Greaser, U. M. Hayward, J. W. Sayre, M. K. Arid and D. R. Abe-

rle, Automated measurement of single and total lung volume from CT, Computer Assisted Tomography, pp.632-640, 1999.

[12] M. S. Brown, L. S. Wilson, B. D. Doust, R. W. Gill and C. Sun, Knowledge-based method for segmentation and analysis of lung boundaries in chest X-ray images, Computerized Medical Imaging and Graphics, pp.463-477, 1999.

[13] M. S. Chen, J. Han and P. S. Yu, Data mining : An overview from database perspective, IEEE Transactions on Knowledge and Data Engineering (TKDE), pp.866-883, 1996.

[14] K. W. Chu and M. H. Wong, Fast time-series searching with scaling and shifting, In Proc. ACM Symposium on Principles of Database Systems (PODS), pp.237-248, 1999.

[15] W. W. Chu, A. F. Cardenas and R. K. Taira, KMeD : a knowledge-based multimedia medical distributed database system. Information Systems, pp.75-96, 1995.

[16] W. W. Chu and K. Chiang, Abstraction of high level concepts from numerical values in databases, In Proc. AAAI Workshop on Knowledge Discovery in Databases, pp.133-144, 1994.

[17] G. Das, D. Gunopulos and H. Mannila, Finding similar time series, In Proc. Principles and Practice of Knowledge Discovery in Databases (PKDD), pp.88-100, 1997.

[18] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, Fast subsequence matching in time-series databases, In Proc. ACM Int'l Conf. on Management of Data (SIGMOD), pp. 419-429, 1994.

[19] K. Fukunaga and P. M. Narendra, A branch and bound algorithms for computing k-nearest neighbors, IEEE Transactions on Computers, Vol.C-24, No.7, pp.750-753, 1975.

[20] D. Q. Goldin and P. C. Kanellakis, On similarity queries for time-series data : Constraint specification and implementation, In Proc. Constraint Programming, pp.137-153, 1995.

[21] A. Hampapur and R. Bolle, Feature based indexing for media tracking, In Proc. IEEE Int'l Conf. on Multimedia and Expo (ICME), 2000.

[22] E. J. Keogh and M. J. Pazzani, Scaling up dynamic time warping to massive datasets, In Proc. Principles and Practice of Knowledge Discovery in Databases (PKDD), 1999.

[23] S. W. Kim, S. Park and W. W. Chu, An index-based approach for similarity search supporting time warping in large sequence databases, In Proc. IEEE Int'l Conf. on Data Engineering (ICDE), pp.607-614, 2001.

[24] R. Lienhart, C. Kuhmunch and W. Effelsberg, On the detection and recognition of television commercials, In Proc. IEEE Int'l Conf. on Multimedia Computing and Systems, 1997.

[25] J. Meng and S.-F. Chang, CVEPS-A compressed video editing and parsing system, In Proc. ACM Multimedia, 1996.

[26] R. Mohan. Video sequence matching, In Proc. Int'l Conf. on Acoustics Speech and Signal Processing (ICASSP), 1998.

[27] S. Park, W. W. Chu, J. Yoon and C. Hsu, A suffix tree for fast similarity searches of time-warped subsequences in sequence databases. Technical Report UCLA-CS-TR-990005, UCLA, 1999.

[28] S. Park, W. W. Chu, J. Yoon and C. Hsu, Efficient searches for similar subsequences of different lengths in sequence databases, In Proc. IEEE Int'l Conf. on Data Engineering (ICDE), pp.23-32, 2000.

[29] S. Park, S. W. Kim, J. S. Cho and S. Padmanabhan, Prefix-querying : An approach for effective subsequence matching under time warping in sequence databases, In Proc. ACM Int'l Conf. on Information and Knowledge Management (CIKM), pp.255-262, 2001.

[30] S. Park, D. Lee and W. W. Chu, Fast retrieval of similar subsequences in long sequence databases, In Proc. IEEE Knowledge and Data Engineering Exchange Workshop (KDEX), pp.60-67, 1999.

[31] L. Rabinar and B.-H. Juang. Fundamentals of Speech Recognition. Prentice Hall, 1993.

[32] D. Ra ei and A. Mendelzon, Similarity-based queries for time series data, In Proc. ACM Int'l Conf. on Management of Data (SIGMOD), pp.13-24, 1997.

[33] J. M. Sanchez, X. Binefa, J. Vitria and P. Radeva, Local color analysis for scene break detection applied to TV commercials recognition, In Proc. Visual 99, 1999.

[34] H. Shatkay and S. B. Zdonik, Approximate queries and representations for large data sequences, In Proc. IEEE Int'l Conf. on Data Engineering (ICDE), pp.536-545, 1994.

[35] K. Shim, R. Srikant and R. Agrawal, High-dimensional similarity joins, In Proc. IEEE Int'l Conf. on Data Engineering (ICDE), pp.301-311, 1997.

[36] M. Sonka, V. Hlavac and R. Boyle Image Processing, Analysis, and Machine Vision, Chapman Hall, 1993.

[37] D. M. Squire, H. Muller and W. Muller, Improving response time by search pruning in content based image retrieval system, using inverted file techniques, In Proc. IEEE Work shop on Content Based Image and Video Libraries, 1990.

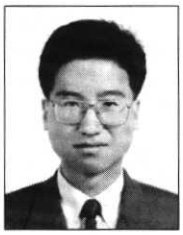
[38] G. A. Stephen, String Searching Algorithms, World Scientific Publishing, 1994.

[39] A. Vailaya, W. Xiong and A. K. Jain, Query by video clip. In Proc. Int'l Conf. on Pattern Recognition, 1998.

[40] J. T. Wang, G. Chirn, T. G. Marr, B. Shapiro, D. Shasha, and K. Zhang Combinatorial pattern discovery for scientific data : Some preliminary results, In Proc. ACM Int'l Conf. on Management of Data (SIGMOD), pp.115-125, 1994.

[41] N. Yazdani and M. Ozsoyoglu, Sequence matching of images, In Proc. Int'l Conf. on Statistical and Scientific Database Management (SSDBM), pp.53-62, 1996.

[42] B.-K. Yi, H. V. Jagadish and C. Faloutsos, Efficient retrieval of similar time sequences under time warping, In Proc. IEEE Int'l Conf. on Data Engineering (ICDE), pp. 201-208, 1998.



김 인 범

e-mail : ibkim@kimpo.ac.kr

1989년 서울대학교 컴퓨터공학과(학사)

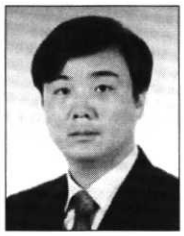
1991년 서울대학교 컴퓨터공학과(석사)

1991년~1994년 대우통신 연구원

1995년~1996년 오라클 코리아 연구개발실 근무

1996년~현재 김포대학 컴퓨터계열 교수

관심분야 : 시공간 데이터베이스, 데이터베이스 시스템, 컴퓨터 이론 및 알고리즘, 컴퓨터 보안, 소프트웨어 취약성 분석



박 상 현

e-mail : sanghyun@postech.ac.kr

1989년 서울대학교 컴퓨터공학과(학사)

1991년 서울대학교 컴퓨터공학과(석사)

2001년 UCLA대학교 전산학과(박사)

1991년~1996년 대우통신 연구원

2001년~2002년 IBM T. J. Watson Research Center Post-Doctoral Fellow

2002년~현재 포항공과대학교 컴퓨터공학과 교수

관심분야 : 시공간 데이터베이스, 멀티미디어 데이터베이스, 데이터마이닝, 데이터웨어하우징, XML, 분산 데이터베이스