

분기 한정적인 동적 타임 워핑 기반의 시계열 패턴인식

(Time Series Pattern Recognition based on
Branch and Bound Dynamic Time Warping)

장 석 우 [†] 박 영 재 ^{**} 김 계 영 ^{***}
(Seok-Woo Jang) (Young-Jae Park) (Gye-Young Kim)

요 약 시계열 패턴 인식에 일반적으로 많이 사용되는 동적인 타임 워핑 알고리즘은 대부분의 연산시간을 상관표를 작성하는데 소비한다. 그리고 이 연산시간을 줄이고자 전역 경로 제약조건을 설정하여 연산 범위에 제한을 두는데, 이것은 패턴의 내용을 고려하지 않은 시간축에 의한 제한이다. 따라서 본 논문에서는 패턴의 형태에 따라 적응적으로 전역 경로 제약조건을 설정하여 보다 효율적으로 패턴인식을 수행하는 분기 한정적인 동적 타임 워핑 알고리즘을 제안한다. 제안된 방법의 성능 평가를 위한 실험에서는 분기 한정적인 동적 타임 워핑 알고리즘이 기존의 동적 타임 워핑 방법과 경로 거리는 유사하면서 연산 시간이 보다 개선되었음을 확인할 수 있었다.

키워드 : 분기한정, 동적 타임 워핑, 시계열 데이터, 전역 경로 제약조건

Abstract The dynamic time warping algorithm generally used in time series pattern recognition spends most of the time in generating the correlation table, and it establishes the global path constraint to reduce the corresponding time complexity. However, the constraint restrains just in terms of the time axis, not considering the contents of input patterns. In this paper, we therefore propose an efficient branch and bound dynamic time warping algorithm which sets the global constraints by adaptively reflecting the patterns. The experimental results show that the proposed method outperforms conventional methods in terms of the speed and accuracy.

Key words : Branch and Bound, Dynamic Time Warping, Time Series Data, Constraints

1. 서 론

인식의 대상이 되는 패턴(pattern)은 크게 정적 패턴

과 동적 패턴으로 구분된다. 정적 패턴은 지문, 숫자, 정맥 등과 같이 고정된 영상의 경우에 해당하며, 동적 패턴은 음성, 제스처, 로봇의 이동궤적 등과 같이 시간에 따라 변화하는 패턴을 의미한다. 다시 말해, 동적 패턴은 시간이라는 변수가 추가되어 정적 패턴의 연속적인 궤적이 하나의 패턴을 형성한다. 따라서 일반적인 패턴 인식 방법으로는 이들 동적 패턴을 인식하기가 어려우므로 동적 패턴의 시변성을 고려한 패턴 인식 알고리즘이 요구된다.

그리고 두 패턴의 길이가 동일한 경우에는 일대일로 대응된 각 성분의 거리를 누적한 총 누적 거리를 비교해서 유사도를 평가할 수 있으나, 길이가 서로 다를 경우에는 매핑(mapping) 함수를 사용해야 한다. 즉, 어느 한 패턴의 길이를 축소하거나 신장하여 두 패턴의 길이를 동일하게 만든 후에 비교를 해야 한다. 그러나 패턴을 축소할 경우에는 정보의 손실이 발생하며, 신장할 경우에는 부가적인 정보가 추가되어 올바른 패턴 정합이 어려

· 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단-신기술융합형 성장동력사업의 지원을 받아 수행된 연구임(No. 2009-0093669)

† 정 회 원 : 안양대학교 디지털미디어학과 교수

swjang@anyang.ac.kr

** 학생회원 : 숭실대학교 컴퓨터학부

webmaster@kuly.org

*** 종신회원 : 숭실대학교 컴퓨터학부 교수

gykim11@ssu.ac.kr

(Corresponding author)

논문접수 : 2010년 3월 4일

심사완료 : 2010년 5월 4일

Copyright©2010 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 소프트웨어 및 응용 제37권 제7호(2010.7)

울 수 있다. 이런 경우 비선형 매핑 함수가 필요한데, 동적 타임 워핑(DTW: dynamic time warping) 알고리즘을 이용하면 비선형 매핑 함수를 최적으로 찾을 수 있다.

DTW 알고리즘은 시간에 따라 변화하는 길이가 서로 다른 두 동적 패턴 사이의 유사도를 판별하기 위한 알고리즘으로 음성 인식, 제스처 인식, 서명 인식 등의 시계열(time series) 패턴을 분석하는 많은 분야에서 사용되고 있다. 일반적으로, DTW 알고리즘은 대부분의 연산시간을 상관표(correlation table)를 작성하는데 소비한다. 그리고 이 연산시간을 줄이고자 전역 경로 제약조건(global path constraint)을 설정하여 연산 범위에 제한을 두는데, 이것은 패턴의 내용을 고려하지 않은 시간 축에 의한 제한이다. 따라서 본 논문에서는 패턴의 형태에 따라 적용적으로 전역 경로 제약조건을 설정하여 효율적으로 패턴인식을 수행하는 분기 한정적인(BnB: brand and bound) DTW 알고리즘을 제안한다. 본 논문에서 제안된 DTW 알고리즘은 기존의 DTW 알고리즘과 경로 거리가 유사하면서 연산 시간이 보다 빠른 패턴 정합을 수행한다.

본 논문은 총 5장으로 구성된다. 1장에서는 본 연구를 수행하게 된 전체적인 개요를 기술하였다. 2장에서는 DTW와 관련된 기존의 연구에 대해 소개하고, 3장에서는 제안된 분기 한정적인 동적 타임 워핑 알고리즘에 대해 설명한다. 그리고 4장에서는 본 논문에서 수행한 실험결과를 보이며, 5장에서는 결론 및 향후연구 방향을 제시한다.

2. 관련 연구

DTW 알고리즘은 크게 전형적인 DTW 알고리즘[1]과 1차 미분을 적용한 DDTW(derivative DTW) 알고리즘[2]의 두 가지 방법으로 분류된다.

2.1 DTW

DTW는 일반적으로 상관표 D 를 사용한 동적 프로그래밍(dynamic programming)을 통해 길이가 서로 다른 두 개의 패턴 $F(f_1, f_2, f_3, \dots, f_n)$ 와 $S(s_1, s_2, s_3, \dots, s_m)$ 의 정합을 수행한다. 먼저, 상관표의 왼쪽 상단에 해당하는 $D_{(1,1)}$ 부터 식 (1)의 방식으로 누적 거리(cumulative distance) $D_{(i,j)}$ 를 정의하는데, 이는 현재 정합하고자 하는 두 패턴 사이의 거리 $d(f_i, s_j)$, 그리고 상관표 $D_{(i,j)}$ 와 인접한 세 요소($\delta_{(1,1)}, \delta_{(1,0)}, \delta_{(0,1)}$)의 누적 거리 중에서 최소값을 더하여 할당한다. 이런 방식으로 상관표의 값을 모두 할당한 후에 $D_{(n,m)}$ 부터 $D_{(1,1)}$ 까지 역추적을 통해 두 패턴 F 와 S 의 최적의 정합경로를 구한다.

$$D_{(i,j)} = d(f_i, s_j) + \min(\delta_{(1,1)}, \delta_{(1,0)}, \delta_{(0,1)}) \quad (1)$$

$$\delta_{(x,y)} = D_{(i-x, j-y)}$$

DTW는 많은 패턴인식의 분야에서 유용하게 사용되지만 대응 경로를 계산할 때 한 패턴의 특징점 하나에 다른 패턴의 여러 특징점들이 매핑되는 현상인 특이점(singularities)이 발생하는 문제가 있다. 이런 특이점 문제를 해결하기 위해 워핑 윈도우, 기울기 가중치, 기울기 제약조건 등의 많은 방법[3]들이 제안되었으나 이 방법들은 본질적으로 허용되는 워핑을 제한하는 기법을 이용하므로, 부정확한 워핑의 가능성이 항상 내재되어 있다는 단점을 가진다. 또한, DTW는 연산시간의 상당부분을 상관표를 작성하는데 소요하므로 전체 연산시간을 줄이기 위해 전역 경로 제약조건(global path constraint)을 사용하여 상관표의 작성 범위를 제한하는 방법[4]들이 소개되었다.

2.2 DDTW

DTW는 입력되는 패턴이 가진 특징점의 y 축 좌표만을 고려하여 두 패턴의 매핑을 시도하므로 특이점이 발생한다. 예를 들어, 동일한 y 값을 가지는 두 개의 특징점 f_i 와 s_j 가 있다고 가정할 때 f_i 가 상승점이고, s_j 가 하강점일 경우에도 DTW는 두 점의 y 값이 동일하므로 서로 매핑된다고 오판한다. DDTW는 이런 특이점 문제를 해결하기 위해 제안된 방법으로 입력되는 패턴을 그대로 사용하는 대신, 식 (2) 및 식 (3)과 같이 입력 패턴의 1차 미분값을 이용하여 두 패턴의 대응경로를 계산한다.

$$f_i = \frac{(f_i - f_{i-1}) + ((f_{i+1} - f_{i-1})/2)}{2} \quad (2)$$

$$s_j = \frac{(s_j - s_{j-1}) + ((s_{j+1} - s_{j-1})/2)}{2} \quad (3)$$

where $1 < i < n, 1 < j < m$

DDTW는 입력 패턴을 그대로 사용하는 DTW 보다 1차 미분된 형태의 패턴을 이용하므로 아웃라이어(outlier)에 보다 강건하며, 시간 복잡도도 $O(mn)$ 으로 DTW와 동일하다는 장점이 있다. 그러나 DDTW는 지수 평탄화(exponential smoothing)를 사용하므로 초기 예측치가 반드시 필요하고, 경로 탐색 시 패턴의 최근 자료에 높은 가중치를 부여함으로써 반복되거나 불규칙한 잡음이 발생한다는 단점이 있다.

3. 분기 한정적인 DTW

시계열 패턴에 적용되는 DTW는 일반적으로 5가지의 제약조건을 가진다. 이 중에서 전역 경로 제약조건은 그림 1과 같이 두 패턴 사이의 전 구간에 걸친 허용 가능한 탐색 영역을 제한하는 조건이다. 예를 들어, 동일한 패턴일 경우 지속시간이 다르지만 대략 시간의 차이는 0.5배에서 2배는 초과하지 않는다는 조건[5]을 설정함으로써 탐색공간을 축소하는 역할을 한다. 그러나 이 조건

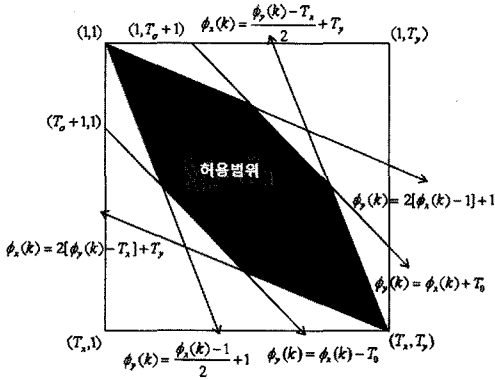


그림 1 DTW의 전역 제약조건

은 두 패턴의 내용을 고려하지 않은 시간축에 의한 단순 제한이다.

따라서 본 논문에서는 최적화 문제에서 사용되는 분기 한정법[6]을 활용하여 두 패턴의 형태에 따라 적용적으로 변화하는 전역 경로 제약조건을 적용함으로써 보다 효율적으로 탐색공간을 축소시킨다. 일반적으로, 분기 한정법은 모든 후보 해를 체계적으로 늘어놓으면서 최적화할 수치의 상한과 하한을 추정하고, 가망 없다는 판정이 나는 해를 제거해 나간다. 제거하는 해에서 파생되는 해는 살펴보지 않기 때문에 불필요한 시간 소모를 줄이게 된다.

제안된 방법에서는 경로 탐색 시 현재까지의 거리, 앞으로 예상되는 거리, 그리고 전체 예상되는 거리를 통해 유효한 경로의 값만을 이용하여 상관표를 작성한다. 그림 2는 분기 한정법을 이용하여 유효한 경로를 선택한 예를 보여준다. 그림에서 검정색 부분은 경로 계산을 수행하지 않은 영역이며, 흰색 부분은 유효한 경로로 판단하여 경로를 계산한 영역이다. 제안된 방법은 이와 같이 선택된 탐색영역 내에서 기존의 방법과 마찬가지로 역추적을 통해 정합 경로를 구하므로 탐색시간을 보다 단축할 수 있다.

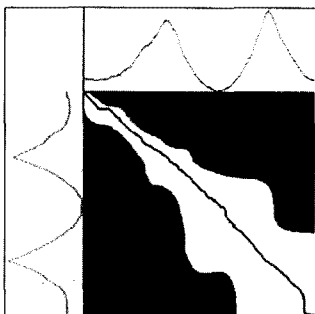


그림 2 제안 방법의 경로제약

제안된 방법에서는 먼저 예상되는 두 패턴 사이의 정합 거리를 구하기 위해 1 대 1 정합을 수행한다. 그러나 두 패턴의 길이가 다를 경우에는 1 대 1 정합이 불가능하므로, 패턴 F와 S 중 길이가 짧은 패턴 S를 F와 길이가 같도록 선형 보간하여 S'을 생성한다. 그런 다음, 패턴 F와 S'을 1대 1 매칭하여 예상되는 전체 경로 거리를 나타내는 Bound 값을 식 (4)를 사용해 산출한다. 그런데 S'이 너무 정확하게 보간되어 Bound 값이 실제 두 패턴 사이의 거리보다 짧게 나올 경우 유효한 경로를 찾을 수가 없는 경우가 발생할 수 있다. 따라서 α 값을 이용하여 Bound 값을 조정한다.

$$Bound = \alpha \sum_{i=1}^n |F_i - S'_i| \quad (4)$$

그런 다음, 전역 경로 제약조건을 고려하여 상관표의 왼쪽상단 $D_{(1,1)}$ 부터 식 (1)을 이용하여 누적 거리 $D_{(i,j)}$ 를 정의한다. 본 논문에서는 상관표의 값을 구할 때 Bound 값과 경로 탐색 시 남은 두 패턴의 예상 정합 거리를 의미하는 Profit 값을 이용하여 전역 경로를 제약한다. 그리고 Profit 값은 상관표를 작성할 때 유효 경로를 찾기 위해 빈번하게 이용되므로 식 (5)와 같이 Profit 테이블을 생성하여 사용한다.

$$Profit_{(0)} = Bound \quad (5)$$

$$Profit_{(i)} = Profit_{(i-1)} - |(F_i - S'_i)|$$

Profit의 초기값은 두 패턴을 정합하였을 때 예상되는 전체 거리이기 때문에 Bound 값으로 초기화한다. 제안된 분기 한정적인 DTW는 기존의 DTW와 달리 연산량을 줄이기 위해 식 (6) 및 식 (7)과 같이 앞서 계산한 Profit 값과 Bound 값을 이용해 계산할 값이 유효한지의 여부를 판단해 전역 경로를 제약한다.

$$\min(\delta_{(1,1)}, \delta_{(1,0)}, \delta_{(0,1)}) \leq Bound \quad (6)$$

$$(D_{(i,j)} + Profit_{(k)}) \leq Bound \quad (7)$$

식 (7)에서 k는 Profit 테이블에서 현재 위치에 대한 Profit 값을 참조하기 위한 인덱스이며, 식 (8)과 같이 계산한다.

$$k = \min(i, \frac{N \times j}{M}) \quad (8)$$

위의 과정을 통해 상관표를 작성한 후에는 $D_{(n,m)}$ 부터 $D_{(1,1)}$ 까지 역추적을 통해 최적의 정합경로를 구한다. 그림 3은 제안된 분기 한정적인 DTW 알고리즘의 의사코드(pseudo-code)를 보여준다.

4. 실험 결과

본 논문에서는 실험을 위해 인텔 펜티엄 Core 2 Duo 2.66GHz의 CPU와 4GB의 메모리가 장착된 컴퓨터를 이용하였고, 운영체제는 마이크로 소프트의 윈도우 7을

```

function BnB-DTW(Array F[], Array S[])
{
    Array S1[];
    int Length1 = Length of Array F[];
    int Length2 = Length of Array S[];

    for i = 1 to Length1 - 1 // 선형 보간
    {
        dx = ceiling(i * (Length2 / Length1)) - (i * (Length2 / Length1));
        S1[i] = S[i] * dx + S[i + 1] * (1 - dx);
    }

    double Bound = 0;
    for i = 1 to Length1 // 두 패턴 사이의 예상되는 한계값 산출
        Bound = Bound + (F[i] - S1[i]);

    Bound = Bound * alpha; // 한계값 조정

    int ProfitTable[];
    ProfitTable[0] = Bound;

    for i = 1 to Length1 // Profit Table 작성
        ProfitTable[i] = ProfitTable[i-1] - (F[i] - S1[i]);

    double D[Length1][Length2];
    // 상관표 초기화
    for i = 0 to Length1
        D[i][0] = Infinity;

    for i = 0 to Length2
        D[0][i] = Infinity;

    D[0][0] = 0;

    for j = 1 to Length2
    {
        for i = 1 to Length1
        {
            k = min(i, ((Length1) * j) / Length2); // Profit Table의 위치 계산

            CurrentCost = min(D[i - 1][j], D[i][j - 1], D[i - 1][j - 1]);

            if (CurrentCost > Bound) // 현재 거리와 한계값에 따른 전역경로 제약
                continue;

            CurrentCost = CurrentCost + (F[i] - S1[i]);

            if (CurrentCost + ProfitTable[k] <= Bound) // 현재 거리, 예상 거리, 한계값에 따른 전역경로 제약
                D[i][j] = CurrentCost;
            else
                D[i][j] = Infinity;
        }
    }

    OptimalPathSearchwithBacktracking(); // 기존의 DTW와 같이 역추적을 통해 최적 경로 탐색
}
    
```

그림 3 제안된 알고리즘의 의사코드

사용하였으며, 컴파일러는 비주얼 C++를 사용하여 프로그램 개발하였다. 제안된 방법의 성능을 측정하기 위한 실험 데이터로는 원형 프로젝션(circular projection) 특징을 이용한 필적 패턴 200개와 VoiceBox라는 MATLAB에 공개된 12차원의 특징(7)을 이용한 음성 패턴 100개의 2가지 시계열 패턴을 사용하였다. 그리고 제안하는 방법과 성능을 비교 평가하기 위해 본 논문에서는 기존의 방법 중 DTW 알고리즘과 DDTW 알고리즘도 구현하였다.

본 논문에서는 정확한 평가를 위해 다음과 같은 3가지 방법으로 성능을 비교 평가한다. 첫째, 제안된 방법으로 설정한 전역 경로가 전역 경로를 제약하지 않았을 때의 최적 거리를 포함하고 있는지의 여부를 평가한다. 둘째, 기존의 전역 경로 제약조건과 비교하였을 때 제안된 BnB-DTW가 경로 제약이 효율적으로 진행되었는지

의 여부를 평가한다. 셋째, 최적 거리의 유용성과 연산 시간을 평가한다.

그림 4는 음성 패턴을 이용하여 전역 경로를 제약하고, 기존의 방법과 제안된 방법의 최적 경로를 비교한 결과를 보여준다. 제안된 방법 이외의 DTW와 DDTW의 경우에는 기존의 전역 경로 제약조건을 사용하였다. 그림에서 검은색 영역은 상관표를 작성할 때 계산하지 않았으며, 흰색 영역은 유효한 경로라고 판단하여 계산한 영역을 나타낸다. 그림 5에서 좌측의 그림은 동일한 필적 패턴을 비교한 결과를 나타내며, 우측의 그림은 서로 다른 필적 패턴을 비교한 결과를 각각 나타낸다. 제안된 방법의 동일 패턴을 비교하는 경우에는 상관표를 작성하기 이전에 계산한 Profit 값이 비교적 정확하여 탐색 영역이 급격하게 좁아지는 것을 확인할 수 있으며, 다른 패턴의 비교에서는 예상한 결과 값과 실제 값의



(a) DTW (b) DDTW (c) BnB-DTW

그림 4 전역경로 제약(음성 패턴)

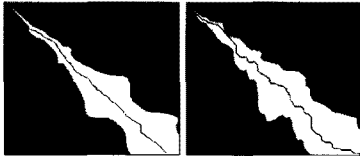


그림 5 전역경로 제약(필적 패턴)

차이가 커서 탐색 영역이 넓어지는 결과를 확인할 수 있다. 그리고 동일 패턴의 비교에서는 탐색 영역이 급격하게 좁아졌음에도 불구하고 기존의 전역 경로 제약조건을 사용해서 산출된 최적 경로를 전역 경로에 대부분 포함하고 있다.

표 1과 표 2는 최소 거리, 연산 시간, 그리고 식 (9)에 의해 계산되는 특이점의 발생빈도 S에 대한 실험 결과를 보여준다. 식 (9)에서 N은 실험에 사용한 동적 패턴의 개수, length는 경로의 길이, m_F 와 m_S 는 동적 패턴 F와 S에서 한 개의 점에 2개 이상의 점이 대응되는 특이점의 개수를 나타낸다.

$$S = \frac{1}{N} \times \sum_1^N \frac{(m_Q + m_C)}{length} \quad (9)$$

제안된 방법은 DTW와 비슷한 최소 거리를 나타내고 있으며, 연산 시간은 DTW 보다 빠른 결과를 얻을 수 있었다. DDTW의 최소 거리는 입력되는 패턴을 수정하므로 다른 두 개의 DTW 최소 거리와는 다소 차이가 발생했다. 그리고 특이점 발생빈도의 경우에는 제안된 방법이 DDTW보다는 높은 빈도를 보였으며, DTW와는 비슷한 수치를 보였다.

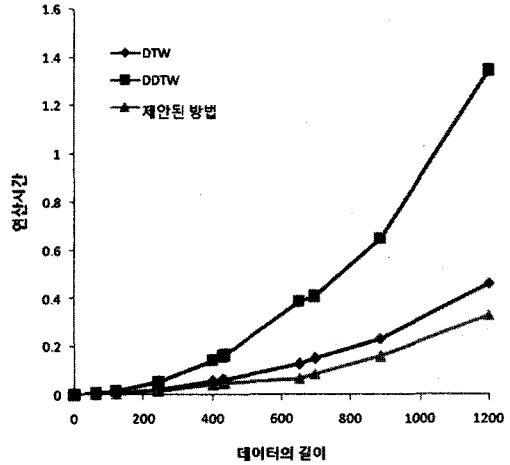


그림 6 패턴 길이에 따른 연산시간 변화량

연산 시간을 비교해 볼 때 제안된 방법은 다른 방법에 비해 속도가 향상되었음을 확인할 수 있다. 본 실험에서 사용된 필적 패턴의 특징은 1차원 데이터로 길이는 100 내외이고, 음성 특징은 12차원 데이터로 길이는 400 내외이다. 실험 결과를 보면 연산 시간이 필적 패턴보다 음성 패턴의 경우 차원이 늘어난 만큼 연산 시간이 증가했음을 확인할 수 있는데, DTW의 경우에는 약 20배, DDTW의 경우에는 약 25배, 제안한 방법의 경우에는 약 10배 정도 증가하였다. 입력된 패턴의 내용에 따라 적용적으로 전역 경로를 제약한 제안한 방법의 경우 패턴의 길이나 특징의 길이가 늘어날 경우에 보다 분명하게 연산 시간이 감소했음을 확인할 수 있었다.

그림 6과 그림 7은 패턴의 길이 및 특징의 길이가 증가함에 따라 변화하는 연산시간을 그래프로 보여준다. DDTW는 입력 데이터를 직접 변환하기 때문에 데이터의 수가 늘어날수록 기하급수적으로 연산시간이 오래 걸리며, 제안된 방법은 패턴에 적용적으로 전역 경로를 제한하였기 때문에 DTW보다 연산시간이 감소하였다.

표 1 최소거리와 연산시간 비교(필적패턴)

입력 패턴	DTW			DDTW			BnBDTW		
	거리	S	시간	거리	S	시간	거리	S	시간
동일	3.6855	0.6438	0.1110	0.8836	0.5936	0.2717	3.5866	0.6317	0.0905
상이	31.473	0.5953	0.1533	2.2911	0.5643	0.3154	36.551	0.5926	0.1218

표 2 최소거리와 연산시간 비교(음성패턴)

입력 패턴	DTW			DDTW			BnBDTW		
	거리	S	시간	거리	S	시간	거리	S	시간
동일	357.09	0.6685	1.9220	252.58	0.6110	5.8670	356.95	0.6793	1.0860
상이	551.40	0.6138	2.7681	254.76	0.5682	6.4165	594.67	0.6216	1.5991

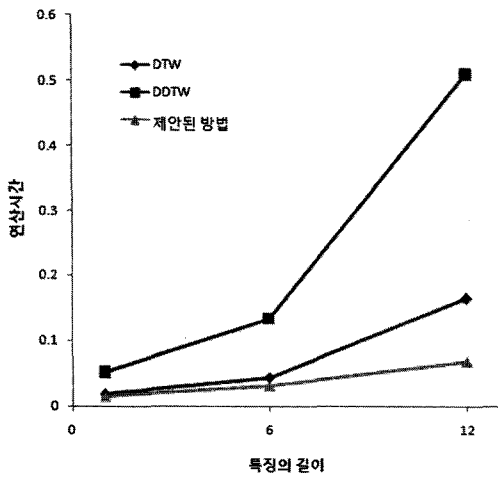


그림 7 특징 길이에 따른 연산시간 변화량

5. 결론 및 향후연구

본 논문에서는 시계열 데이터를 인식하기 위해 기존의 DTW 알고리즘에 분기 한정법을 적용하여 패턴에 적응적으로 변화하는 전역 경로 제약조건을 생성하는 분기 한정적인 DTW 알고리즘을 새롭게 제안하였다. 기존의 DTW는 패턴을 고려하지 않은 시간축에 의한 전역 경로 제약조건을 사용하였으나 제안된 알고리즘에서는 분기 한정법을 이용함으로써 전역 경로 제약조건을 패턴에 따라 유동적으로 적용할 수 있었다. 그 결과 탐색 범위가 축소되었으며, 연산 시간도 입력패턴에 따라 1.5배에서 2배 정도 향상되었음을 확인하였다. 특히, 패턴 특징의 길이가 증가할수록 보다 좋은 속도 향상의 효과를 보였다.

향후에는 보다 다양한 실험 데이터를 가지고 제안된 분기 한정적인 DTW 알고리즘을 비교 평가할 예정이며, 제안된 방법을 시계열 패턴을 사용하는 다양한 실제 응용 분야에 적용하여 그 우수성을 검증할 예정이다.

참고 문헌

[1] H. Sakoe, S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol.26, no.1, pp.43-49, February 1978.

[2] E. J. Keogh, M. J. Pazzani, "Derivative Dynamic Time Warping," In *Proceedings of the First SIAM International Conference on Data Mining*, pp.1-11, April 2001.

[3] Y. Mizuhara, A. Hayashi, N. Suematsu, "Embedding of Time Series Data by Using Dynamic Time Warping Distances," *Systems and Computers in*

Japan, vol.37, no.3, pp.1-9, February 2006.

[4] G. Tomasi, F. V. Berg, C. Andersson, "Correlation Optimized Warping and Dynamic Time Warping as Preprocessing Methods for Chromatographic Data," *Journal of Chemometrics*, vol.18, no.5, pp. 231-241, May 2004.

[5] L. Rabiner, B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[6] Richard Neapolitan, Kumarss Naimipour, *Foundations of Algorithms Using C++ Pseudocode*, Third Edition, Jones & Bartlett Publishers, 2003.

[7] M. Faundez-Zanuy, M. Hagmüller, G. Kubin, "Speaker Identification Security Improvement by Means of Speech Watermarking," *Pattern Recognition*, vol.40, no.11, pp.3027-3034, November 2007.



장 석 우

1995년 숭실대학교 전자계산학과 졸업(학사). 1997년 숭실대학교 대학원 컴퓨터학과 졸업(석사). 2000년 숭실대학교 대학원 컴퓨터학과 졸업(박사). 2009년~현재 안양대학교 디지털미디어학과 교수. 관심분야는 로봇비전, 증강현실, 게임, 비디오 색인 및 검색 등



박 영 재

2005년 청운대학교 컴퓨터과학과 졸업(학사). 2008년 숭실대학교 컴퓨터학과 졸업(석사). 2008년~현재 숭실대학교 컴퓨터학과 박사과정. 관심분야는 영상분석, 패턴인식, 컴퓨터 비전 등



김 계 영

1990년 숭실대학교 전자계산학과 졸업(학사). 1992년 숭실대학교 대학원 컴퓨터학과 졸업(석사). 1996년 숭실대학교 대학원 컴퓨터학과 졸업(박사). 2001년~현재 숭실대학교 컴퓨터학부 교수. 관심분야는 컴퓨터 비전, 형태인식, 생체인식,

증강현실, 영상처리 등