지역화된 템플릿기반 동적 시간정합을 이용한 모바일 제스처인식

(Mobile Gesture Recognition using Dynamic Time Warping with Localized Template)

최 봉 환 † 민 준 기 †
(BongWhan Choe) (Jun-Ki Min)

조 성 배 ††
(Sung-Bae Cho)

요 약 최근 모바일기기에 탑재된 가속도 센서가 제스처기반 모바일 사용자 인터페이스에 활용됨에 따라 동적 시간정합(Dynamic Time Warping, DTW)기반 인식기에 대한 연구가 활발하다. DTW는 학습샘플을 매칭 템플릿으로 사용하기 때문에 별도의 학습과정이 없다. 하지만 인식 시 입력 데이터를 모든 템플릿과 비교해야하기 때문에 계산복잡도로 인하여 모바일환경에 적용하기 어렵다. 본 논문에서는 이러한 문제를 해결하기 위해 지역화된 소수의 템플릿을 사용하는 DTW기반 제스처 인식기를 제안한다. 지역화된 템플릿은 k-평균 클러스터링(k-means clustering) 알고리즘을 사용하여 학습 제스처 셋의 유사한 패턴들을 k개의 그룹으로 묶고, 각 그룹의 중심(centroid)에 가까운 패턴을 DTW인식기의 템플릿으로 선택한다. 이러한 방법으로 템플릿수를 줄여 인식속도를 향상하고, 템플릿의 다양성을 유지하여 인식성능저하를 최소화한다. 실험 결과 제안하는 방법이 학습 템플릿을 전부 사용하는 DTW보다 약 5배 빠른 인식속도를 보였으며, 템플릿을 임의로 선택한 경우보다 안정적인 성능을 보임을 확인했다.

키워드 : k-Means클러스터링, 지역화된 템플릿, 제스처 인식, 동적 시간 정합

Abstract Recently, gesture recognition methods based on dynamic time warping (DTW) have been actively investigated as more mobile devices have equipped the accelerometer. DTW has no additional training step since it uses given samples as the matching templates. However, it is difficult to apply the DTW on mobile environments because of its computational complexity of matching step where the input pattern has to be compared with every templates. In order to address the problem, this paper proposes a gesture recognition method based on DTW that uses localized subset of templates. Here, the k-means clustering algorithm is used to divide each class into subclasses in which the most centered sample in each subclass is employed as the localized template. It increases the recognition speed by reducing the number of matches while it minimizes the errors by preserving the diversities of the training patterns. Experimental results showed that the proposed method was about five times faster than the DTW with all training samples, and more stable than the randomly selected templates.

Key words : k-Means Clustering, Localized Template, Gesture Recognition, Dynamic Time Warping

1. 서 론

최근 Micro-Electro-Mechanical Systems (MEMS) 칩 형태의 저가 가속도 센서를 탑재한 모바일 장치가 크게 늘어나고 있다. 이들 모바일 폰의 다수는 PDA의 형태를 띄고 있으며, 이와 같은 변화에는 마이크로소프트社의 Windows Mobile™과 애플컴퓨터社의 iPhone 보급이 큰 역할을 하였다. 기존의 모바일 폰에서는 멀티미디어 재생 중 혹은 어플리케이션 사용 중에 사용자의 자세에 따른 화면 방향 전환을 지원하는 등 중력 방향을 감지를 위해 가속도 센서를 가장 많이 사용했다. 하지만 일부 모바일 폰에서는 주사위 놀이나, 낚시, 볼링 등 게임의 형태로 가속도 센서가 사용되고 있으며, Wii와 같은 게임기에서는 사용자 제스처 입력을 지원하여 게임의 재미와 몰입감을 증대시키고 있다.

현재 상용화된 가속도 센서기반 UI는 상당수가 방향 전환의 감지, 가속도의 평균 크기, 가속도의 반복 주파수, 가속 타이밍 등에 대한 규칙 기반 시스템을 사용하고 있다. 그러나 이러한 방식은 다양한 동작을 인식하는데 있어 한계가 있다. 기존 방식의 한계를 극복하기 위

· 본 연구는 지식경제부 및 한국산업기술평가관리원의 산업원천기술개발사업의 일환으로 수행하였음(10033807, 다중센서 및 협업을 위한 자율 학습 기반 상황인지 기술)
· 이 논문은 제36회 추계학술발표회에서 '지역화된 템플릿기반 동적 시간정합을 이용한 모바일 제스처인식'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 연세대학교 컴퓨터과학과
bitbyte@sclab.yonsei.ac.kr
loomlike@sclab.yonsei.ac.kr
†† 종신회원 : 연세대학교 컴퓨터과학과 교수
sbcho@yonsei.ac.kr

논문접수 : 2009년 12월 23일
심사완료 : 2010년 1월 21일

정보과학회논문지 : 컴퓨팅의 실제 및 레터 제16권 제4호(2010.4)

한 다양한 방법이 시도되고 있는데 그 중 한 가지가 동적 시간 정합(Dynamic time warping, DTW) 인식기처럼 기록되어 있는 패턴과 비교하는 방법이다. DTW 인식기 방법은 N개의 템플릿에 대한 거리를 비교하려면 입력 데이터의 길이가 n, i번째 패턴의 길이가 mi, 일 때, 계산 복잡도가 $O(n\sum_{i=1}^{N} m_i)$로 매우 높다. 따라서 작은 컴퓨팅 파워를 가진 모바일 UI에 적용하기 위해서는 최소한의 패턴템플릿으로 제스처의 다양성을 최대한 모델링하도록 하는 방법이 필요하다. 이러한 문제를 해결하기 위해 k-평균 클러스터링(k-means Clustering) 알고리즘을 사용해 DTW의 비교 패턴의 수를 효과적으로 줄여주는 방법을 제안한다. 또한, 클러스터링 알고리즘을 적용한 경우와 적용하지 않은 경우의 성능 차이를 실험을 통해 검증한다.

2. 관련연구

2.1 제스처 인식

최근 이뤄지는 제스처 인터페이스 연구는 은닉 마르코프 모델(Hidden Markov models, HMMs), 지지벡터 기계(Support vector machine, SVM), 동적 베이지안 네트워크(Dynamic Bayesian network, DBN), 선형 분석법, 주성분 분석법, 상관함수 등과 같은 다양한 방법이 적용되고 있다. 또한 인식 장비로서 영상센서와 가속도 센서, 터치스크린 등이 이용되어 사용자의 직관적이고, 풍부한 사용자 인터페이스를 목적으로 연구가 진행되고 있다. 이중에서 I.B. Ozer 등은 영상 정보를 입력받아 HMMs으로 인식하는 방법을 제안했고[1], R. Solera-Urena등은 SVM에 HMMs을 커널로 사용하여 데이터 글로브 기반 수화 인식기를 제안했다[2]. J. Rett와 J. Dias는 영상으로 입력 받은 인체 동작을 DBNs으로 모델링하였다[3]. 최근 J. Liu 등이 제안한 uWave는 사용자에 따른 제스처의 개인화를 고려해 DTW를 인식기로서 사용했다[4]. 최근 제스처 인식에서 활발히 연구되고 있는 DTW인식기는 적은 학습 데이터에 효과적인 알고리즘으로 사용자의 특성에 따른 제스처 인터페이스의 개발 및 개인화에 적합하다.

2.2 동적 시간 정합

DTW는 1970년대 음성 인식 분야에서 적은 수의 패턴으로 짧은 패턴을 인식하기 위해 제안 됐다[5]. 이후 HMMs에 음성인식의 주요 기술의 자리를 내주었으나, 아직 DTW는 적은 숫자의 샘플로도 동작 가능한 인식기 개발이 가능하다는 점에서 여전히 강점을 가지고 있다. 1990년대 말부터 가속도 센서 및 모바일 장치의 보급과 성능 향상에 기반해 DTW는 제스처 및 필기 인식 분야에서 다시 활발히 연구되었으며, 2000년대에서는 이러한 DTW를 실시간 인식기를 필요로 하는 시스템에 적용하는 연구가 진행 중이다.

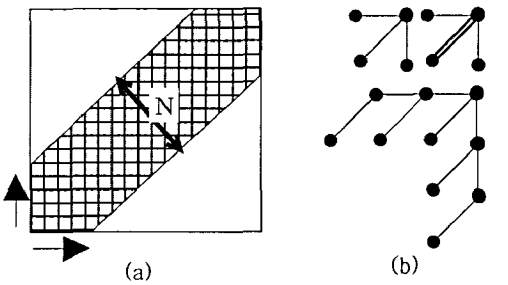

그림 1 (a) Sakoe-Chiba Band의 전역제약과 (b) DTW의 도약 방법들

```
Function DTW(Data[] row, Data[] col) : Distance
  Distance D[row.length+1, col.length+1];
Begin
  Foreach ( Distance d in D )
    d := max;
  D[0,0] = 0;
  For r := 1 to row.Length
    For c := 1 to col.Length
      Distance cost = d(row[r], col[c] );
      D[r, c] := cost + min( D[r-1, c-1],
                             D[r,   c-1],
                             D[r-1, c] )
  Return D[row.Length, col.Length]
End
```

그림 2 DTW 알고리즘

DTW는 두 개의 순차 데이터의 시간길이를 왜곡함으로써 두 패턴의 최적의 정합(matching)을 구하고, 해당 정합에서의 두 데이터 사이의 거리를 계산하는 알고리즘이다. 일반적으로 1차원 계열 데이터 사이의 정합을 구하며, 이때, DTW는 동적 프로그래밍(Dynamic programming)을 통해 구현된다. DTW의 속도 향상을 위 Fast DTW[6], Stream DTW[7]와 같은 변형이 제안 되었으나, 정합을 위한 지역 제약(Local constraint 또는 Warping Method)과 계산 범위를 한정하는 전역 제약(Global Constraint)은 초기에 제안된 Sakoe와 Chiba가 사용했던 방법이 여전히 많이 사용되고 있다(그림 1(a)).

Sakoe등은 그림 1(b)와 같은 다양한 형태의 도약 방법도 제안 했으나 그 중 많이 사용되는 형태는 왼쪽 위의 가장 간단한 1차 대칭형 도약(1'st order symmetric warping) 방법이다. 1차 대칭형 도약 방법은 $D(i,j)$와 같이 표현될 때, 그림 2와 같은 형태로 구현 된다. 그림 2에서 비용함수 d()는 두 입력 데이터 row와 col의 각

요소사이의 거리를 측정하는 함수로 유클리드거리(Euclidean distance)와 같은 간단한 비교 함수가 사용된다. 두 패턴 i, j간의 DTW거리 $D(i,j)$는 다음과 같이 계산된다.

$$D(i,j) = \min\begin{bmatrix} D(i,j-1) \\ D(i-1,j) \\ D(i-1,j-1) \end{bmatrix} + d(i,j) \quad (1)$$
$$D(0,0) = 0, D(i,0) = D(0,j) = \max$$

2.3 k-평균 클러스터링 알고리즘

k-평균 클러스터링 알고리즘(k-Means Clustering 혹은 c-Means Clustering)은 입력 데이터에 대해서 k개의 클러스터를 자동으로 결정하는 알고리즘의 한가지이다. 이는 샘플과 클러스터 중심 간의 거리의 총합 I를 최소화하는 k개의 클러스터 반복적으로 중심을 탐색한다[8]. 데이터 셋의 샘플 수를 n이라 하고, k번째 클러스터의 중심을 Z_k라고 할 때, I는 다음과 같이 계산된다.

$$I = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{j,i} \|x_i - Z_j\|^2 \quad (2)$$

식 (2)에서 $u_{j,i}$는 분할 행렬(Partition matrix)의 (j, i)번째 원소를 나타내는 것으로, 샘플 x_i가 클러스터 j에 속한 경우 1, 그 밖의 경우는 0의 값을 갖는다. k-평균 클러스터링 알고리즘은 최초의 중심을 임의로 또는 발견적인 방법으로 정하기 때문에 초기 중심에 따른 클러스터링 성능 차이가 있지만, 일반적으로 빠른 수렴을 하기 때문에 샘플의 숫자가 많고, 고차원 특징을 갖는 시계열 데이터의 클러스터링에 적합하다고 볼 수 있다.

3. 클러스터링 기반 템플릿 구성

본 논문에서는 DTW의 템플릿을 선택하기 위해 k-평균 클러스터링 알고리즘을 사용했다. k-평균 클러스터링 알고리즘에 의해 선택된 각 클러스터의 중심 샘플 데이터를 DTW의 패턴 템플릿으로 선택하는 것으로 다음과 같은 3가지 효과가 기대 된다.

1) 비교 대상 제한을 통한 계산 시간 감소
2) 아웃라이어(outlier)의 제거로 인식 정확도 보장
3) 패턴-데이터간의 거리를 평준화를 통한 인식기 파라메터 설정용이

패턴 데이터의 개수를 클러스터의 개수로 고정함으로써 매칭 시간이 줄어들다. 아웃라이어는 클러스터의 중심보다는 클러스터의 소속으로 선택되어 사실상 제거되므로 DTW의 인식 오류를 줄일 수 있다. 마지막으로 실시간 입력 데이터에 대해서 제스처 인식을 위해서는 인식 대상이 해당 제스처에 속하지는 판별하기 위해 임계치(Threshold)를 둘 필요가 있다. 인식 과정에서는 입력 데이터와 템플릿 간의 거리가 임계치보다 클 경우 거부해야 한다. 이 때, 평균 거리가 일정하도록 선출된

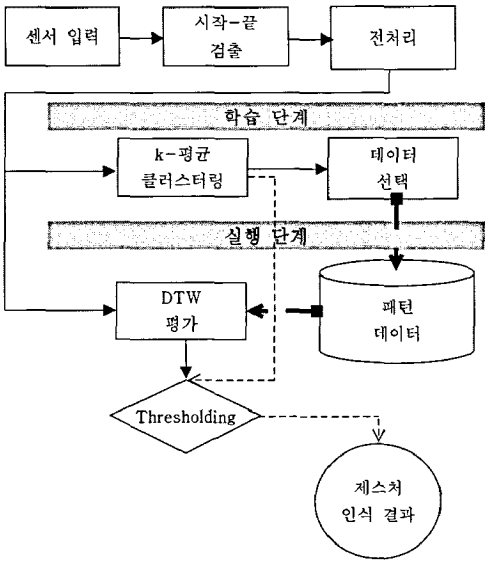

그림 3 k-평균 클러스터링을 사용한 동적시간 정합 템플릿 선택

클러스터 중앙을 템플릿 데이터로 사용하면 클러스터의 평균 거리를 기준으로 손쉽게 임계치의 값을 결정할 수 있다. 그림 3은 제안하는 방법의 전체 흐름을 보여준다.

3.1 제스처 데이터의 클러스터링 및 템플릿 선택

k-평균 클러스터링의 계산을 위해서는 클러스터 중심과 각 샘플간의 거리를 계산할 필요가 있다. 본 논문에서 사용한 방법은 유클리드 거리의 합을 사용했다. 즉, 길이가 N인 두 가속도 데이터 X와 Y가 있고, $X \ni x_i$, $Y \ni y_i$ ($i = 1$ .... N) 일 때, X와 Y사이의 거리 $d_E(X,Y)$는 다음과 같다.

$$d_E(X,Y) = \sum_{i=1}^{N} d(x_i, y_i), \quad (3)$$

$$d(i,p) = \sqrt{(i_x - p_x)^2 + (i_y - p_y)^2 + (i_z - p_z)^2} \quad (4)$$

유클리드 거리를 통해 두 시계열 데이터를 비교하는 방법은 빠른 속도와 좋은 성능을 보여 준다. 다만, 이 방법을 사용할 경우 입력 데이터의 모두 N으로 동일해야 한다. 따라서 클러스터링을 하기 전에 입력 데이터의 길이를 N으로 리샘플링(resampling)한다.

3.2 동적 시간 정합 인식기 구성

본 논문에서 사용한 DTW인식기의 비용함수는 입력 데이터가 i, 패턴 데이터가 p일 때, 가속도의 x, y, z축을 유클리드거리로 계산하는 식 (3)과 같은 함수를 사용했다. 또한 전역 제약으로 가장 단순한 Sakoe-Chiba 밴드 방식으로 구성 하였다. 동적 시간 정합은 두 시계열 데이터의 최적의 정합을 찾고, 그 거리를 계산하기 위한 알고리즘이다. 따라서 동적 시간 정합 인식기로 여

러 형태를 인식하기 위해서는 결합 모델이 필요하다. 본 논문에서는 결합 모델로 승자 독식(Winner takes it all)모델을 선택했다. 승자 독식 모델은 모든 인식기에서 각각 산출된 결과 중 가장 우수한 한 개가 채택 되는 방식이다. DTW 인식기로 산출된 결과는 단순값(Scalar value)이기 때문에 우위를 결정하기 쉽다. 따라서 승자 독식모델을 사용할 경우 간단하게 인식기를 구성할 수 있다. 다만 이러한 구성 방법은 인식과 거부를 결정할 수 없는 단점이 있다. 이러한 단점을 극복하기 위해 승자 독식 이후에 임계치를 두어 DTW의 결과가 임계체 보다 작은 경우에만 승자 독식에 반영되도록 했다.

4. 실험 및 결과

4.1 데이터 셋 및 실험 설정

실험 데이터는 3축 가속도 센서를 이용하여 수집했다. 수집은 50Hz로 수행했으며, 20대 남자 4명에게서 2일간 수집했다. 제스처는 표 1과 같은 7종류 25가지 동작으로 구성했으며, 수집은 1회 수집 당 동작별로 5회씩, 총 2일간 10회의 데이터를 수집했다. 25가지 동작에 대해서 사람별 실험(4 fold)과 날짜별 실험(2 fold)으로 구성했다. 수집된 데이터는 길이가 0.4초에서 1.4초로의 분포로 확인됐다. 4인분의 데이터를 모아 실험에 사용 했으므로 총 1000개의 데이터를 사용했다. Snap은 장치를 잡은 손목을 꺾었다가 복원 하는 동작, Bounce는 장치를 잡은 속목을 이동하는 동작, Rotate와 Tilt는 장치를 잡은 손을 꺽고 멈추는 동작, Tip과 Tap은 장치를 치는 동작과 장치로 다른 장치를 치는 동작, Shake는 장치를 흔드는 동작으로 설정했다. DTW인식기의 Sakoe-Chiba Band 크기는 20으로 설정해 최소한의 제약으로 수행했다. 클러스터의 개수는 3으로 했고. 클러스터링을 위한 리샘플링 크기는 샘플 중 가장 긴 데이터의 길이에 맞췄다.

표 1 실험에서 사용한 제스처 구성

종류	방향	기호	방향	기호
Snap	Left	NL	Up	NU
	Right	NR	Down	ND
Bounce	Forward	BF	Backward	BB
Rotate	Horizontal	RH	Vertical	RV
Tilt	Left	LL	Up	LU
	Right	LR	Down	LD
Tip	Left	TL	Down	TD
	Right	TR	Up	TU
	Back	TB		
Tap	Left	AL	Up	AU
	Right	AR	Down	AD
	Front	AF	Back	AB
Shake	Left-Right	SLR	Forward-Backward	SFB

실험은 크게 3가지로 수행 되었는데 1) 모든 패턴 데이터를 사용한 경우, 2) 패턴 데이터를 임의로 3가지 고른 경우 3) 패턴 데이터를 클러스터에서 3가지 고른 경우로 수행 되었다. 특히 실험 2)와 실험 3)은 반복 수행 했는데, 2)와 3)은 클러스터링 및 데이터 선택에 따른 편차가 3)의 경우가 적음을 보이기 위해서 반복 수행 했다. 1)과 3)은 인식률의 편차를 보이고, 동시에 수행 시간의 차이를 확인할 수 있는 대조군으로 추가했다.

4.2 제스처 인식 결과

4.2.1 전체 데이터를 사용한 경우와 비교

표 2의 실험 결과에서 4회 평균한 인식률이 전체 데이터를 사용한 것에 비해 나쁨을 알 수 있다. 표 2의 (a)는 비교 데이터로 사람별의 경우 동작 당 30개, 날짜별의 경우 동작 당 20개의 데이터를 사용한 경우로 (b)의 3-평균 클러스터에 비해 대량의 처리를 하기 때문에 수행 시간 측면에서는 약 10배의 차이를 보임을 알 수 있다.

4.2.2 임의 선택과 클러스터링의 성능 비교

표 3의 실험 결과 에서 알 수 있듯이, 임의 선택의 경우 인식률은 최고 70%에서 최저 23%까지 매우 편차가 크고 불안정함을 알 수 있다. 실제로 임의 선택은 반복된 횟수에 따라서도 많은 차이를 보이고 있다. 반면 k-Means로 클러스터링을 하는 경우 초기 중심을 임의로 설정해도 45%내외에서 유사한 성능을 보임을 알 수

표 2 전체 사용과 지역화된 템플릿 비교실험 결과

(a) 전체 데이터를 사용한 경우

	사람별				날짜별	
	F0	F1	F2	F3	F0	F1
인식(%)	82.5	81.5	85.5	78.5	71.8	66%
시간(s)	207	208	236	204	278	198

(b) 3-평균 클러스터를 사용한 경우(4회 평균)

평균	사람별				날짜별	
	F0	F1	F2	F3	F0	F1
인식(%)	49.5	45.5	49	40	45.5	47.3
시간(s)	23	22	39	25	61	39

표 3 임의 선택과 지역화된 템플릿 비교실험 결과

(a) 임의로 선택한 3개의 데이터를 사용한 경우

횟수	사람별				날짜별	
	F0	F1	F2	F3	F0	F1
1회	53.5	40	66.5	46.5	37	33
2회	47.5	46.5	69.5	57.5	45.8	38
3회	61	51.5	70	47	50.3	41.5
4회	69.5	50	51.5	45.5	46.5	23.8

(b) 3-평균 클러스터를 사용한 경우

횟수	사람별				날짜별	
	F0	F1	F2	F3	F0	F1
1회	49.5	45.5	49	40	45.5	47.3
2회	47	43.5	45	42.5	52.8	38.5

있다. 따라서 안정적인 인식기 개발을 목적으로 한다면 클러스터링을 통해 인식률을 안정화하면서 속도를 향상할 수 있다는 것을 알 수 있다.

5. 결론 및 향후 연구

본 논문에서는 모바일 기기에서 제스쳐 UI를 위한 DTW인식기를 개발하기 위해 DTW의 패턴 데이터를 k-평균 클러스터링을 사용해 구성하는 방법을 제안했다. DTW는 비록 인식 성능은 좋지만, 그 수행 과정이 느려 최신의 모바일 장치에서도 좋은 성능을 내기 힘들다는 단점이 있다. 따라서 본 논문에서 k-평균 클러스터링 방법이 다양한 동작을 모바일 장치에서 사용하기 위해서는 패턴 데이터의 숫자를 최소화해서 비교 시간을 줄이는데 유효할 수 있음으로 보였다.

k-평균 클러스터링 방법은 초기 중심점의 설정에 따른 성능 편차가 있으며, 클러스터링을 위해 리샘플링과정을 거치게 되어 정보 손실이 발생할 수 있다. 향후 이러한 단점을 보완하여 시계열 데이터에 대한 고성능, 고정확도를 유지하는 클러스터링 방법을 적용할 계획이다. 또한, 추가적인 패턴에 대해 강화 학습 형태의 클러스터링 방법을 개발하게 된다면 사용자 적응적인 인식기를 개발할 수 있을 것으로 생각된다. 마지막으로 현재는 임의로 작은 값인 3을 클러스터의 개수로 사용했으나, 향후 최적의 k값에 대한 연구도 필요하다.

참 고 문 헌

[1] I. B. Ozer, T. Lu, and W. Wolf, "Design of a real-time gesture recognition system: high performance through algorithms and software," *IEEE, Signal Processing Magazine*, vol.22, no.3, pp. 57-64, 2005.

[2] R. S.-Urena, D. M.-Iglesias, A. G.-Antolin, C. P.-Moreno, and F. D.-de-Maria, "Robust ASR using support vector machines," *Speech Communication*, vol.49, no.4, pp.253-267, 2007.

[3] J. Rett and J.Dias, "Gesture recognition using a marionette model and dynamic bayesian networks (DBNs)," *ICIAR 2006*. LNCS, vol.4141, pp.69-80. 2006.

[4] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer -based personalized gesture recognition and its applications," in *Pervasive and Mobile Computing (PerCom)*, vol.5, no.6, pp.657-675, 2009.

[5] H. Sakoe and 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, 1978.

[6] S. Salvador and P. Chan, "Fast DTW: Toward Accurate Dynamic Time Warping in Linear Time and Space," *Intelligent Data Analysis*, vol.11, no.5, pp.561-580, 2007.

[7] P. Capitani and P. Ciaccia, "Warping the time on data streams," *Data & Knowledge Engineering*, vol.62, no.3, pp.438-458, 2007.

[8] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1988.