

Feature-Strengthened Gesture Recognition Model based on Dynamic Time Warping

Hyuck Tae Kwon[†] · Suk Kyoon Lee^{**}

ABSTRACT

As smart devices get popular, research on gesture recognition using their embedded-accelerometer draw attention. As Dynamic Time Warping(DTW), recently, has been used to perform gesture recognition on data sequence from accelerometer, in this paper we propose Feature-Strengthened Gesture Recognition(FsGr) Model which can improve the recognition success rate when DTW is used.

FsGr model defines feature-strengthened parts of data sequences to similar gestures which might produce unsuccessful recognition, and performs additional DTW on them to improve the recognition rate. In training phase, FsGr model identifies sets of similar gestures, and analyze features of gestures per each set. During recognition phase, it makes additional recognition attempt based on the result of feature analysis to improve the recognition success rate, when the result of first recognition attempt belongs to a set of similar gestures. We present the performance result of FsGr model, by experimenting the recognition of lower case alphabets.

Keywords : Gesture Recognition, Dynamic Time Warping(DTW), Machine Learning

Dynamic Time Warping 기반의 특징 강조형 제스처 인식 모델

권혁태[†] · 이석균^{**}

요 약

스마트 디바이스가 보편화되면서 이에 내장된 가속도 센서를 사용한 제스처의 인식에 관한 연구가 주목받고 있다. 최근 가속도 센서 데이터 시퀀스를 통한 제스처 인식에 Dynamic Time Warping(DTW) 기법이 사용되는데, 본 논문에서는 DTW 사용 시 제스처의 인식률을 높이기 위한 특징 강조형 제스처 인식(FsGr) 모델을 제안한다.

FsGr 모델은 잘못 인식될 가능성이 높은 유사 제스처들의 집합에 대해 특징이 강조되는 데이터 시퀀스의 부분들을 정의하고 이들에 대해 추가적인 DTW를 실행하여 인식률을 높인다. FsGr 모델의 훈련 과정에서는 유사 제스처들의 집합들을 정의하고 유사 제스처들의 특징들을 분석한다. 인식 과정에서는 DTW를 사용한 1차 인식 시도의 결과 제스처가 유사 제스처 집합에 속한 경우, 특징 분석 결과를 기반으로 한 추가적인 인식을 시도하여 인식률을 높인다. 알파벳 소문자에 대한 인식 실험을 통해 FsGr 모델의 성능 평가 결과를 보인다.

키워드 : 제스처 인식, Dynamic Time Warping(DTW), 기계학습

1. 서 론

IT 기술의 발전으로 스마트폰과 같은 다양한 스마트 기가들이 보급되고 있으며, 이들은 내장된 센서들을 활용해 헬스케어, 인포테인먼트, 군사 및 산업 등 다양한 분야로 사용 범위를 넓혀가고 있다. 최근 스마트 기기의 가속도 센서를 통한 사용자 제스처 인식에 대한 관심이 높아지고 있다 [6, 9]. 가속도 센서 기반의 제스처 인식은 기계학습, 패턴인식

분야에서 연구되었으며 인식 알고리즘으로는 DTW (dynamic time warping), SVM, HMM(hidden Markov Model) 등이 사용되었다[3, 6, 8]. 본 논문에서는 가속도 센서 기반의 제스처의 인식에서 DTW의 인식률을 높이는 특징 강조형 인식 FsGr(Feature-strengthened Gesture Recognition) 모델을 제안한다.

1957년 Bellman에 의해 처음 소개된 DTW[1]는 길이가 서로 다른 두 시계열 데이터 시퀀스를 대응(alignment)시켜 유사도를 계산하는 알고리즘으로 음성인식, 데이터 마이닝 등의 분야에 널리 사용되어왔다[4, 7].

최근 가속도 센서를 사용한 DTW 기반의 제스처 인식 연구가 진행되고 있다. Liu는 가속도 센서 기반의 개인화된

[†] 정 회 원 : 단국대학교 컴퓨터과학과 박사수료

^{**} 중신회원 : 단국대학교 소프트웨어학과 교수

Manuscript Received : November 28, 2014

First Revision : February 2, 2015

Accepted : February 13, 2015

* Corresponding Author : Suk Kyoon Lee(sklee@dankook.ac.kr)

제스처 인식 알고리즘 uWave를 제안하였는데[6], 이는 여덟 개의 제스처들이 4000개 이상의 데이터들에 대해 95% 수준의 인식 정확도를 보였다. 또한 uWave는 DTW가 적은 수의 샘플로도 훈련이 가능하다는 점에 착안하여 한 번의 훈련으로 사용 가능한 개인 맞춤형(user-dependent) 제스처 인식 개념을 제안하고 이는 개인 인증 등에 사용될 수 있음을 제시했다.

그리고 다차원 데이터 시퀀스들에 대해 DTW를 적용하여 제스처를 인식하는 연구가 시도되었다[2, 5]. [5]에서는 다중 센서(multi-sensor)로부터 발생하는 일정 기간의 다차원의 데이터에 대해 데이터의 시간적 융합(temporal fusion)을 시도하고 이에 DTW를 적용하여 사용자의 행위를 추정하는 연구 결과를 제시하였다. 데이터의 융합을 위해 센서 간의 차이를 고려한 정규화(normalization) 개념과, 연속된 데이터의 흐름으로부터 데이터 시퀀스를 처리하기 위해 시작점과 끝나는 점이 명확하지 않은 경우를 위한 확장된 DTW 알고리즘을 제안하였고, 두 개의 가속도 센서를 사용한 경우와, 벤치마크 데이터 셋을 실험한 결과 HMM 기법보다 뛰어난 성능을 보였다. [2]에서는 다차원 DTW 알고리즘을 사용하여 음악적 제스처의 인식을 시도하였다.

최근 국내에서도 가속도 센서 데이터에 대한 DTW 기반의 제스처 연구가 발표되었다. [9]에서는 훈련 방법으로 DTW를, 인식 방법으로는 DTW와 k-최근접 이웃 알고리즘(k-Nearest Neighbors)을 사용하여 스마트폰을 통해 로봇제어를 수행하였다.

DTW 기반의 제스처 인식 연구가 많은 주목을 받고 있지만 기존의 연구들은 비교적 단순한 제스처들에 대한 인식만을 시도하고 있다. DTW를 통한 제스처 인식이 활성화되기 위해서는 현실적으로 의미를 갖는 다양한 제스처들에 대한 인식이 필요하다. 본 연구에서는 다양한 제스처에 대한 인식 정확도를 높이기 위한 시도로 DTW 기반의 특징 강화형 제스처 인식(FsGr: Feature-strengthened Gesture Recognition) 모델을 제안한다. FsGr 모델에서는 인식 과정에서 잘못 인식될 수 있는 유사한 제스처들에 대해 유사 제스처 집합을 정의하고 유사 제스처 집합의 제스처들에 대해서는 제스처들의 특징들을 추출한 후 이를 기반으로 한 추가적인 제스처 인식 작업을 수행한다.

본 논문의 구성은 다음과 같다. 2절에서는 DTW 알고리즘의 개요를 소개하고 3절에서는 FsGr 모델의 기본 개념들을 정의하고 인식 알고리즘을 제시한다. 4절에서는 FsGr 모델의 훈련 알고리즘을 설명하고 5절에서는 실험 및 성능 평가를 제시하고 6절에서는 결론과 미래 연구를 언급한다.

2. DTW

DTW(Dynamic time warping) 시간이나 속도의 차이로

길이와 동일하지 않은 시계열 데이터 시퀀스들 사이의 유사도를 측정하는 알고리즘으로 음성인식, 데이터 마이닝, 제스처 인식 등의 시계열 데이터의 패턴 인식에 사용된다[1, 5-7]. DTW는 두 시계열 데이터 시퀀스 $\mathbf{P} = p_1, p_2, \dots, p_m$, $\mathbf{Q} = q_1, q_2, \dots, q_n$ 에 대해 비선형 대응(nonlinear alignment)을 시키는데, 임의의 p_i 와 q_j 의 대응 (i, j) 를 w_k 라 할 때 두 시퀀스의 대응은 정합 경로(warping path) $\mathbf{W} = w_1, w_2, \dots, w_z$ 로 표현된다. 정합 경로 \mathbf{W} 는 다음의 특징을 갖는다. m, n, z 가 각각 $\mathbf{P}, \mathbf{Q}, \mathbf{W}$ 의 길이라 할 때

- $w_1 = (1, 1), w_z = (m, n)$
- $w_k = (i, j)$ 이면 $w_{k+1} = (i, j), \text{ or } (i+1, j) \text{ or } (i, j+1)$
- $\max(m, n) \leq z < m+n$

이다. 위의 조건을 만족하는 정합 경로의 수는 데이터 시퀀스의 길이에 대해 기하급수적으로 증가한다.

\mathbf{P} 와 \mathbf{Q} 의 임의의 데이터 쌍 (p_a, q_b) 의 거리 비용 계산 함수를 $d(p_a, q_b)$ 라 할 때, \mathbf{P} 와 \mathbf{Q} 에 대한 임의의 대응 $w_k = (i, j)$ 의 비용 $c(w_k) = d(p_i, q_j)$ 로 표현된다. 정합 경로 \mathbf{W} 의 비용 $C(\mathbf{W})$ 와 $DTW(\mathbf{P}, \mathbf{Q})$ 는 다음과 같이 정의된다.

$$C(\mathbf{W}) = \sum_{k=1}^z c(w_k)$$

$$DTW(\mathbf{P}, \mathbf{Q}) = \min \{ C(\mathbf{W})/z \mid \mathbf{W} \in \{ \mathbf{P}, \mathbf{Q} \text{에 대한 정합 경로들의 집합} \} \}$$

$DTW(\mathbf{P}, \mathbf{Q})$ 는 모든 정합 경로의 비용들 중 최소 비용을 계산하므로 그 비용이 상당히 크다. 다음의 동적계획법(dynamic programming)에 기초한 계산 방법이 보다 현실적이다.

$$D(i, j) = d(p_i, q_j) + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\}$$

이 경우 두 데이터 시퀀스 \mathbf{P} 와 \mathbf{Q} 의 유사도는 $DTW(\mathbf{P}, \mathbf{Q}) = D(m, n)$ 이며 이때 알고리즘 복잡도(time complexity)는 $O(mn)$ 로 평가된다. 본 논문에서는 가속도 센서 데이터가 사용되므로 비용 함수 d 는 다음과 같이 정의된다. p_x, p_y, p_z 는 각각 p 의 x 축, y 축, z 축 값을 의미한다.

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2}$$

3. FsGr 모델

3.1 필요성 및 정의

DTW 기반의 제스처 인식 연구들은 대부분 차별성이 강한 단순한 동작들을 식별 대상으로 한다. 그러나 DTW 기반의

제스처 인식이 산업 현장에서 활용되기 위해서는 사용자들이 자주 사용하는 의미 있는 동작들, 가령 영어의 알파벳, 문자, 숫자, 수신호, 음악의 지휘 등에 대한 인식이 필요하다. 그러나 이들은 동작이 다양하고 복잡할 뿐 아니라 유사한 동작들이 존재해서 인식 과정에 종종 잘못된 결과를 제시한다. 예를 들어 영어의 c와 e, h와 n 등은 동작이 유사하여 잘못 인식되곤 한다. 본 논문에서는 유사한 동작들에 대해 인식의 정확도를 높이기 위한 **특징 강조형 제스처 인식(FsGr: Feature-Strengthened Gesture Recognition) 모델**을 제안한다.

FsGr 모델은 DTW 기반의 실시간 제스처 인식 모델로 제스처들의 특징 분석을 통해 제스처의 차별화되는 부분에 대해 추가적인 인식 과정을 통해 인식률을 높인다. **FsGr** 모델의 주요 개념들은 다음과 같다. **G**는 식별하고자 하는 제스처들의 집합, 그리고 **T**는 제스처 인식에 사용되는 대표 데이터 시컨스(exemplar)들의 집합이고, 본 논문에서의 데이터 시컨스는 가속도 센서가 발생하는 3축 데이터 시컨스를 의미한다. **T**에 속한 모든 대표 데이터 시컨스는 훈련 과정을 통해 생성되며 간단히 대표 시컨스로 호칭한다. 제스처 g_i ($1 \leq i \leq n$)의 인식에는 대표 시컨스 t_i 가 사용되는데 편의상 같은 인덱스 첨자가 사용한다.

$$\mathbf{G} = \{ g_1, g_2, \dots, g_n \}, \quad \mathbf{T} = \{ t_1, t_2, \dots, t_n \}$$

DTW 기반의 인식 방법들[2, 5, 6, 9]에서는 임의의 데이터 시컨스 t 의 제스처를, t 와 **T**의 모든 대표 시컨스들에 대해 $\text{ArgMin}_{x \in T} \text{DTW}(t, x)$ 이 나타내는 대표 시컨스의 제스처로 추정한다. 즉 가장 작은 **DTW** 비용 함수 값을 보이는 대표 시컨스의 제스처를 t 의 제스처로 결정한다.

FsGr 모델에서 잘못 인식될 가능성이 높은 유사 제스처들을 **유사 제스처 집합**이라 하며 이들은 훈련 및 테스트 과정에서 생성된다. 유사 제스처 집합은 다음과 같이 정의된다. 임의의 데이터 시컨스 t 와 대표 시컨스 집합 **T**에 DTW를 적용했을 때, t 의 제스처를 g_i 로 판정했으나 실제로는 g_k 인 경우, g_k 가 g_i 의 유사 제스처 집합 \mathbf{sG}_i 에 속한다. 가령 t 가 나타내는 실제 제스처는 e 였으나 DTW 적용 시 t 를 c 로 판정하면 e 는 c 의 유사 제스처 집합에 포함된다. **FsGr** 모델에서는 공집합이 아닌 유사 제스처 집합의 제스처들에 대한 추가적인 인식을 위해 유사 제스처 집합에 대해 **FsGr** 서브모델을 정의한다.

유사 제스처 집합 \mathbf{sG}_i 에 대한 **FsGr** 서브모델 \mathbf{sM}_j 은 $\langle g_i, \mathbf{sG}_i, \mathbf{sT}_i, \text{part_bits} \rangle$ 의 튜플로 정의되는데 \mathbf{sT}_i 는 g_i 와 \mathbf{sG}_i 에 속한 제스처들의 인식률을 높이도록 구성된 대표 시컨스들의 집합이고 **part_bits**은 \mathbf{sT}_i 의 대표 시컨스들의 구성 방법을 나타내는 비트 시컨스다.

$$\text{part_bits} = b_1 b_2 \dots b_p \quad b_m \in \{0, 1\} \quad 1 \leq m \leq p$$

part_bits의 길이는 $|\text{part_bits}| = p$ 이고 **part_bits**의 각 비트는 p 등분된 대표 시컨스의 각 부분에 대한 포함 여부를 나타낸다. 즉 $b_i = 1$ 이면 p 등분한 대표 시컨스의 i 번째 부분을 포함한다. 예를 들어 **part_bits** = 0001이면, 이의 **part_bits**의 길이, 즉 $|0001| = 4$ 이고 대표 시컨스는 사등분된 후 이의 마지막 부분이, **part_bits** = 0011이면 사등분 후 마지막 두 부분이 새로운 대표 시컨스로 사용된다. 함수 **build_seq**는 대표 시컨스 t 에 대해 **part_bits**를 적용하여 새로운 대표 시컨스를 반환하는 함수다. 이때 \mathbf{sG}_i 에 대한 대표 시컨스들의 집합 \mathbf{sT}_i 는 다음과 같이 정의된다.

$$\begin{aligned} \mathbf{sT}_i &= \{ \text{build_seq}(t, \text{part_bits}) \mid t \in \mathbf{T} \wedge \\ &\quad \text{gesture of } t \in \mathbf{sG}_i \cup \{g_i\} \} \\ \mathbf{SM} &= \{ \mathbf{sM}_1, \mathbf{sM}_2, \dots, \mathbf{sM}_k \} \end{aligned}$$

SM은 **G**에 관련된 **FsGr** 서브모델들의 집합이다.

5절의 Table 1에서는 알파벳 소문자 인식을 위한 **FsGr** 서브모델들의 예제가 제시되는데 편의상 \mathbf{sT}_i 는 생략했다. 첫 번째 행은 제스처 c (즉 $g_i = c$)에 대한 **FsGr** 서브모델을 표현하는데, c 의 유사 제스처 집합 \mathbf{sG}_i 는 $\{e\}$ 이며 \mathbf{sT}_i 는 c 와 e 의 대표 시컨스들의 집합으로 구성된다. 제스처 c 의 **part_bits**는 011000이며, 제스처 c 와 이의 유사 제스처 집합에 속한 제스처들 간에 차별성을 높일 수 있는 부분은 6등분 시 두 번째와 세 번째 부분임을 의미한다.

이제 **FsGr** 모델을 정의한다. **FsGr** 모델 **M**은 $\langle \mathbf{G}, \mathbf{T}, \mathbf{SM}, \mathbf{d} \rangle$ 튜플로 표현되며 \mathbf{d} 는 서브모델에 속할 대표 시컨스 생성 시 원래의 대표 시컨스를 몇 등분할지를 나타내는 것으로 이는 서브모델들의 **part_bits**의 길이를 결정한다. 가령 $\mathbf{d} = 6$ 이면 서브모델의 $|\text{part_bits}| = 6$ 의 의미로 대표 시컨스들을 6등분하여 분석한다.

3.2 FsGr 모델의 인식 알고리즘

인식 대상이 되는 임의의 데이터 시컨스 t 에 대해 **FsGr** 모델의 제스처 인식 알고리즘은 1차 DTW와 2차 DTW의 두 단계로 진행된다.

1차 DTW : $\text{ArgMin}_{x \in T} \text{DTW}(t, x)$ 이 반환하는 대표 시컨스의 제스처 g_i 에 대한 \mathbf{sG}_i 가 존재하면

2차 DTW로 진행하고 아니면 g_i 를 t 의 제스처로 정한다.

2차 DTW : \mathbf{sG}_i 의 **part_bits**이 a 라 하면,

$\text{ArgMin}_{x \in \mathbf{sT}_i} [\text{DTW}(\text{build}(t, a), x)]$ 이 반환하는 대표 시컨스의 제스처를 t 의 제스처로 정한다.

1차 DTW는 데이터 시컨스 t 에 대해 **T**에 속한 대표 시컨스들을 적용하는 기존 DTW의 기법과 동일하다. 그러나 결과 제스처에 유사 제스처 집합이 존재하면 **T**에 **part_bits**

를 적용한 데이터 시퀀스에 대해 유사 제스처 집합에 대한 대표 시퀀스들을 적용하는 DTW를 실행하게 된다. 이를 2차 DTW라 한다.

4. FsGr 모델의 훈련 알고리즘

본 절에서는 FsGr 모델의 훈련 개요와 part_bits 생성 알고리즘에 대해 설명한다.

4.1 FsGr 모델의 인식 알고리즘

FsGr 모델에서의 훈련은 G에 대한 대표 시퀀스의 집합 T를 구하는 부분과 FsGr 서브모델들의 집합 SM을 구하는 부분으로 구성된다. DTW에서는 대표 시퀀스들, 즉 T를 구하는 다양한 훈련 방법들[5]이 있는데, 본 논문에서는 그중 최소선택(minimum selection)[5]을 사용했다. 이는 동일 제스처에 대한 샘플 데이터 시퀀스들에 대해 각 데이터 시퀀스와 이를 제외한 모든 데이터 시퀀스들 간의 DTW 비용의 합들을 구한 후, 비용의 합 중 최솟값의 데이터 시퀀스를 대표 시퀀스로 정하는 방법이다.

SM의 계산은 각 유사 제스처 집합들에 대해 FsGr 서브모델들을 생성하는 것으로, 이 작업의 핵심은 DTW의 훈련을 통해 얻은 대표 시퀀스들의 집합 T를 기반으로 각 유사 제스처 집합에 대해 차별성 있는 데이터 시퀀스들을 생성하도록 part_bits를 구성하는 데 있다. part_bits에 대한 내용은 4.2 절에서 설명하고 우선 FsGr 모델의 훈련 개요를 설명한다.

FsGr 모델의 훈련 개요

1. DTW 훈련 알고리즘을 통해 각 제스처에 대해 대표 시퀀스들을 생성한다.
2. 모든 대표 시퀀스들에 대해 테스트를 거쳐 각 대표 시퀀스들의 인식률을 계산한다. 이때 테스트는 훈련 데이터나 다른 데이터를 사용할 수 있다.
3. 위에서 계산된 인식률이 목표 인식률 a보다 낮은 제스처들에 대해서는 유사 제스처 집합들을 구성한다.
4. 모든 유사 제스처 집합에 대해 part_bits 생성 알고리즘을 통해 part_bits를 계산하고 대표 시퀀스들을 계산하여 서브모델들을 생성한다.

목표 인식률 a는 1 이하의 실수값으로 애플리케이션의 필요에 따라 결정되는데 이의 값이 너무 작으면 유사 제스처 집합들이 생성되지 않을 것이고 너무 크면 대부분의 제스처에 대해 유사 제스처 집합이 생성될 수 있다. 구성된 제스처들이 단순하고 차별성이 크면 목표 인식률 a는 1에 근접하게 정할 수 있다.

4.2 part_bits의 생성 알고리즘

유사 제스처 집합의 제스처들의 대표 시퀀스에 적용되는 part_bits는 특징 부분을 표현하는 것으로 인식률 향상에 직접적인 영향을 준다. 임의의 서브모델의 sG_i의 part_bits를 간단히 part_bits_i로 표현하자. part_bits_i는 g_i에 대해 sG_i의 제스처들을 차별화할 수 있도록 데이터 시퀀스들의 부분들을 표현하는 개념으로 이는 다음의 두 데이터 시퀀스 간의 part_bits_{x,y} 개념을 통해 정의된다.

제스처 g_x의 데이터 시퀀스 t_x와 제스처 g_y의 데이터 시퀀스 t_y에 대해 part_bits_{x,y}는 차별성이 높은 부분들을 나타내는 비트 시퀀스로 가정하면, sG_i의 part_bits_i는 제스처 g_i와 sG_i에 속한 각각의 제스처들에 대해 다음과 같이 정의된다. ||는 비트 논리합(bitwise or)을, 그리고 n은 sG_i의 제스처의 수를 의미한다.

$$\text{part_bits}_i = \text{part_bits}_{i,p1} \parallel \text{part_bits}_{i,p2} \parallel \dots \parallel \text{part_bits}_{i,pn}$$

단, $g_{p_j} \in \text{sG}_i \wedge 1 \leq j \leq n$

part_bits_{ij}는 제스처 g_i와 제스처 g_j의 데이터 시퀀스들을 각각 d등분된 데이터 시퀀스 중 어떤 부분을 포함시킬지를 나타내는 길이가 d인 비트 시퀀스다. 우선 part_bits_{ij}의 정의에 필요한 개념을 소개한다. 함수 ToListOfsublists(t_i, d)는 데이터 시퀀스 t_i를 d등분한 서브 리스트들의 리스트를 반환한다. 즉 t_i의 k번째 부분을 t_{ik}라 할 때, ToListOfsublists(t_i, d) = < t_{i1}, t_{i2}, ..., t_{id-1}, t_{id} >로 표현된다. 이를 통해 part_bits_{ij}의 개념이 정의되는데, part_bits_{ij}의 k번째 비트값은 (DTW(t_i, t_j)/d)*β ≤ DTW(t_{ik}, t_{jk})이면 1로, 아니면 0으로 설정된다. 단, 1 이상의 값으로 응용 분야의 특성에 따라 결정된다.

β = 1, 즉 DTW(t_i, t_j)/d ≤ DTW(t_{ik}, t_{jk})는 k번째 서브 시퀀스들 t_{ik}와 t_{jk}가 평균적인 차별성을 보일 때 새로운 시퀀스에 이들을 포함시킨다는 의미다. β는 1 이상의 값을 가지며 실험을 통해 결정된다. 이제 집합 sG_i의 part_bits_i의 계산 알고리즘을 다음에 제시한다.

sG_i의 part_bits_i 생성 알고리즘

가정 :

- S_Val은 가속도 센서의 데이터의 타입으로 x, y, z축의 값을 표현하는 구조체를 의미
- 제스처 g_i의 데이터 시퀀스를 t, sG_i의 데이터 시퀀스들의 집합을 T로 표현
- setXero(d)는 길이 d의 비트형 리스트를 생성하고 모든 비트를 false로 설정
- ToListOfsublists(t, d)는 리스트 t를 d등분한 서브 리스트들의 리스트를 반환
- setOne(Bits, j)는 Bits의 j번째 비트값을 1로 설정
- Or(Bits1, Bits2)은 Bits1과 Bits2에 대해 비트 Or를 수행 후 결과를 반환

```

List<Bit> Gen_Part_Bits (List<S_Val> t,
                      List<List<S_Val>> T, int d, double beta)
{
    List<Bit> PartBits = setZero(d);
    List<Bit> tempBits = setZero(d);

    List<List<S_Val>> conv_t = ToListOfsublists(t, d);

    foreach ( List<S_Val> x in T ) {
        double dtw_temp = DTW(t, x)/d;
        List<List<S_Val>> conv_x = ToListOfsublists(x, d);

        for ( int j = 0; j < d; j++) {
            if ((DTW(conv_t[j], conv_x[j])/d)*beta >= dtw_temp)
                setOne(tempBits, j);
        }
        PartBits = Or(PartBits, tempBits);
    }
    return PartBits;
}

```

위의 알고리즘에서 데이터 시퀀스의 d 등분된 부분의 포함 여부에는 β 의 값이 중요한 역할을 한다. 이 값이 크면 상당히 차별성이 큰 부분만을 포함하게 되므로 적절한 값의 설정이 중요하다. $part_bits_i$ 의 모든 비트가 다 1로 설정되는 경우는 서브모델로서의 의미가 없으므로 이러한 서브모델은 삭제한다.

5. 실험 및 성능 평가

본 절에서는 **FsGr** 모델에 대한 성능 평가를 위해 알파벳 필기체 소문자에 대한 인식을 시도하였다. 소문자 알파벳에는 유사한 제스처들이 포함되어있어 **FsGr** 모델의 평가에 적합하고 다양한 응용에 활용이 가능하기 때문이다.

LG G2 스마트폰의 가속도 센서를 실험에 사용하였으며 **FsGr** 모델의 a , β , d 는 각각 1, 1.2, 6으로 설정하였다. $d=6$ 이므로 모든 $part_bits$ 의 길이는 6으로 결정된다. 적절한 β 와 d 를 찾기 위해 다양한 상황의 실험을 수행했는데 그 결과 d , 즉 $part_bits$ 의 길이는 6 또는 7일 경우가 가장 인식률이 뛰어나고 8 이상의 경우 오히려 감소하는 경향을 보였다. 한편 β 값은 1.2 근처에서 가장 인식률이 높고 오히려 커질수록 인식률이 하락하는 경향을 보이는데 이는 일정 수준 이상으로 차별성이 큰 부분만을 포함하면 오히려 전체적인 차별화를 떨어뜨림을 나타낸다.

각 제스처에 대한 학습으로 생성된 **FsGr** 서브모델들은 Table 1에 제시되어있는데, 제스처 g_i 에 대한 유사 제스처 집합 sG_i 와 이에 적용될 $part_bits_i$ 를 보이고 있다. $part_bits_i$ 의 내역을 보면 유사한 동작들 간의 차이 부분에 해당하는

비트가 설정되어있음을 알 수 있다. Table 1에서 제시한 알파벳 필기체 소문자의 경우 원소가 두 개 이상인 유사 제스처 집합이 존재하지 않으나 이는 학습시키는 사람의 제스처의 특징을 반영한 것으로 사람마다 달라질 수 있다.

Table 1. FsGr sub model for recognition of lower case alphabet

| g_i | sG_i | $part_bits_i$ |
|-------|--------|----------------|
| c | {e} | 011000 |
| i | {v} | 000101 |
| n | {h} | 010000 |
| r | {f} | 001100 |
| v | {x} | 000110 |

Table 2는 Table 1에서 제시한 **FsGr** 서브모델들을 기반으로 한 제스처 인식 실험 결과를 제시한다. 이때 제스처는 학습자의 제스처들을 사용하였다. Table 2A에서는 각 제스처에 대해 1차 DTW의 실험 결과를 보이고, 제스처에 서브모델이 존재하는 경우는 Table 2B에서 2차 DTW 실험 결과를 포함시키고 있다. 1차 DTW 실험 결과에서는 ‘문자’와 ‘횃수’는 1차 DTW로 인식된 문자와 문자의 인식 횃수를, ‘실제 문자/횃수’는 인식된 문자의 실제 문자와 횃수를, 그리고 ‘성공(오인) 횃수’는 성공한(실패한) 인식 횃수를 의미한다. 예를 들면 세 번째 행의 경우, c로 인식된 횃수가 11회이고 그중 실제 제스처가 c였던 경우는 10회, 그리고 한 번은 e가 c로 오인되었음을 의미한다. 2차 DTW 실험 결과에서 ‘문자/횃수’는 추가 인식 작업에서 인식된 문자와 횃수를, 성공 횃수는 최종 성공 횃수를, 그리고 증감은 2차 DTW에서 증가된 성공 인식 수를 의미한다. 세 번째 행의 경우 2차 DTW에서 ‘문자/횃수’가 c로 인식된 경우가 10회, e로 인식된 경우가 1회로 1차 DTW에서 c로 오인되었던 경우가 2차 DTW에서 e로 제대로 인식했다.

전체적으로 1차 DTW 실험 결과를 보면, **FsGr** 서브모델이 존재하지 않는 제스처는 오인된 경우가 발생하지 않았고 서브모델이 존재하는 다섯 제스처들의 경우 1차 DTW 결과 다섯 건이 오인되었으며 이들 중 2차 DTW 시도에서 네 건이 제대로 인식되어 80%의 개선 효과가 있음을 보였다. 1차 DTW 실험 결과는 일반적인 DTW를 적용한 경우로 98.08%의 인식률을 보이는데 FsGr 모델을 사용하면 인식률이 99.62%로 증가한다.

Table 2. Recognition result for trainer's gestures

A. Result of 1st DTW experiment

| 문자 | 횟수 | 실제문자/횟수 | 성공(오인) 횟수 | 서브모델 존재여부 |
|----|-----|-----------|-----------|-----------|
| a | 10 | a/10 | 10 | |
| b | 10 | b/10 | 10 | |
| c | 11 | c/10, e/1 | 10 (1) | O |
| d | 10 | d/10 | 10 | |
| e | 9 | e/9 | 9 | |
| f | 9 | f/9 | 9 | |
| g | 10 | g/10 | 10 | |
| h | 8 | h/8 | 8 | |
| i | 11 | i/10, v/1 | 10 (1) | O |
| j | 10 | j/10 | 10 | |
| k | 10 | k/10 | 10 | |
| l | 10 | l/10 | 10 | |
| m | 10 | m/10 | 10 | |
| n | 12 | n/10, h/2 | 10 (2) | O |
| o | 10 | o/10 | 10 | |
| p | 10 | p/10 | 10 | |
| q | 10 | q/10 | 10 | |
| r | 11 | r/10, f/1 | 10 (1) | O |
| s | 10 | s/10 | 10 | |
| t | 10 | t/10 | 10 | |
| u | 10 | u/10 | 10 | |
| v | 9 | v/9 | 9 | O |
| w | 10 | w/10 | 10 | |
| x | 10 | x/10 | 10 | |
| y | 10 | y/10 | 10 | |
| z | 10 | z/10 | 10 | |
| 합계 | 260 | | 255 (5) | |

B. Result of 2nd DTW experiment

| 1차 DTW 결과 | | | | 2차 DTW 결과 | |
|-----------|----|-----------|-----------|-----------|-----------|
| 문자 | 횟수 | 실제 문자/횟수 | 성공(오인) 횟수 | 문자/횟수 | 성공(증감) 횟수 |
| c | 11 | c/10, e/1 | 10 (1) | c/10, e/1 | 11 (1) |
| i | 11 | i/10, v/1 | 10 (1) | i/11 | 10 (0) |
| n | 12 | n/10, h/2 | 10 (2) | n/10, h/2 | 12 (2) |
| r | 11 | r/10, f/1 | 10 (1) | r/10, f/1 | 11 (1) |
| v | 9 | v/9 | 9 (0) | v/9 | 9 (0) |
| 합계 | 54 | | 49 (5) | | 53 (4) |

Table 3. Recognition result for non-trainer's gestures

A. Result of 1st DTW experiment

| 문자 | 횟수 | 실제 문자/횟수 | 성공(오인) 횟수 | 서브모델 존재여부 |
|----|-----|---------------------|-----------|-----------|
| a | 11 | a/10, c/1 | 10 (1) | |
| b | 7 | b/3, k/3, v/1 | 3 (4) | |
| c | 8 | c/4, e/4 | 4 (4) | O |
| d | 11 | d/6, k/1, l/1, t/3 | 6 (5) | |
| e | 11 | e/6, c/5 | 6 (5) | |
| f | 10 | f/10 | 10 | |
| g | 10 | g/10 | 10 | |
| h | 1 | h/1 | 1 | |
| i | 10 | i/10 | 10 | O |
| j | 8 | j/8 | 8 | |
| k | 4 | k/4 | 4 | |
| l | 5 | l/5 | 5 | |
| m | 11 | m/10, x/1 | 10 (1) | |
| n | 19 | n/10, h/9 | 10 (9) | O |
| o | 10 | o/10 | 10 | |
| p | 21 | p/10, b/7, k/2, r/2 | 10 (11) | |
| q | 10 | q/10 | 10 | |
| r | 10 | r/8, j/2 | 8 (2) | O |
| s | 10 | s/10 | 10 | |
| t | 7 | t/7 | 7 | |
| u | 14 | u/10, l/4 | 10 (4) | |
| v | 9 | v/9 | 9 | O |
| w | 17 | w/10, d/4, x/2, y/1 | 10 (7) | |
| x | 6 | x/6 | 6 | |
| y | 10 | y/9, x/1 | 9 (1) | |
| z | 10 | z/10 | 10 | |
| 합계 | 260 | | 206 (54) | |

B. Result of 2nd DTW experiment

| 1차 DTW 실험 결과 | | | | 2차 DTW 실험 결과 | |
|--------------|----|-----------|-----------|--------------|-----------|
| 문자 | 횟수 | 실제 문자/횟수 | 성공(오인) 횟수 | 문자/횟수 | 성공(증감) 횟수 |
| c | 8 | c/4, e/4 | 4 (4) | c/4, e/4 | 8 (4) |
| i | 10 | i/10 | 10 | i/10 | 10 (0) |
| n | 19 | n/10, h/9 | 10 (9) | n/16, h/3 | 13 (3) |
| r | 10 | r/8, j/2 | 8 (2) | r/10 | 8 (0) |
| v | 9 | v/9 | 9 | v/9 | 9 (0) |
| 합계 | 56 | | 41 (15) | | 48 (7) |

학습자가 아닌 다른 사람의 제스처 인식의 실험 결과는 Table 3에 제시하였다. Table 3A에서는 1차 DTW 실험 결과가 제시되는데, 54건의 오인된 경우들 중 **FsGr** 서브모델이 정의된 제스처들은 15건, 서브모델이 없는 제스처들에서는 39건이 발생하였고 1차 DTW 인식률은 79.2%로 평가된다.

2차 DTW 실험 결과는 Table 3B에 제시되는데, 15건의 오인된 경우 중 7건이 성공하게 되어 46%의 개선 효과가 발생하며, 전체적인 인식률은 81.9%로 판정된다. 학습자가 아닌 사람의 제스처 인식의 낮은 인식률은 주로 개인의 제스처의 특징의 차이에 기인하는 것으로 이는 DTW 학습 방법의 변경을 통해 이러한 현상을 일부 개선할 수 있을 것으로 보인다. 또한 1차 DTW 결과 FsGr 서브모델들이 정의되지 않았던 제스처들에서 많은 오인이 발생한 것은 학습 대상의 범위를 넓히면 상당 부분 해소될 것으로 보인다.

본 절의 FsGr 모델의 성능 평가의 주요 부분은 서브모델들의 정의를 기반으로 한 2차 DTW의 인식 작업의 효과를 분석하고자 하는 것으로, 위의 두 실험 결과를 볼 때 FsGr 서브모델들이 정의된 제스처의 경우 확실한 인식률의 개선 효과가 보인다. FsGr 모델의 도입으로 인한 인식 작업은 기존의 DTW를 통한 인식 작업, 즉 1차 DTW, 그리고 FsGr 서브모델이 있는 제스처인 경우에 적용되는 2차 DTW 작업으로 구성된다. 따라서 2차 DTW 작업은 추가 비용이 발생하는데, 2차 DTW의 발생 확률은 모든 제스처의 발생 빈도를 동일하다고 가정할 때, (서브모델의 수/제스처의 수)의 확률을 갖는다. 한편, 2차 DTW 작업의 비용은 서브모델의 유사 제스처 집합에 속한 제스처들의 수에 의해 결정되는데 대개의 경우 1~4의 범위에 속한다. 이론적으로는 유사 제스처 집합에 모든 제스처들이 포함될 수 있으나 이는 거의 불가능하다. 따라서 현실적으로 FsGr 모델을 사용해서 발생하는 추가 비용은 그다지 크지 않다고 사료된다.

정리하면, FsGr 모델은 기존의 DTW를 통한 제스처 인식 방법을 사용하면서 기존의 DTW로 식별하지 못하는 유사한 제스처들에 대해 차별성이 높은 데이터 시퀀스의 부분들만을 추출해 다시 DTW를 적용하는 방법이다. 제스처 인식을 위해 데이터 시퀀스에서 차별성이 높은 부분을 추출해서 이들에 대해 DTW를 적용하는 연구는, 저자들의 알고 있는 범위 내에서는 처음 시도되는 방법이며 추가적인 비용 또한 높지 않아 다양한 분야에 사용될 수 있을 것으로 사료된다.

6. 결 론

본 연구에서는 DTW 기반의 FsGr 모델을 제안하고 이를 통해 가속도 센서 기반의 제스처 인식을 시도하였다. 기존

의 연구들은 대개 다른 패턴 인식 방법의 결합을 통해 제스처 인식을 수행했으나 본 논문에서는 비슷한 동작들로 구성된 유사 제스처 집합을 정의하고 이에 속한 제스처들의 특징들을 추출하여 비교하였다. 영어의 알파벳 필기체 소문자의 인식을 통해 FsGr 모델의 성능을 확인하였다.

FsGr 모델은 DTW가 사용되는 모든 인식 작업에 적용할 수 있으므로 다양한 활용이 예상된다. 현재 FsGr 모델의 성능 개선을 위한 작업과 연속된 문자들에 대한 인식 작업이 진행 중이며 이의 결과는 추후 논문으로 발표할 예정이다.

References

- [1] R. Bellman, "Dynamic Programming," *Princeton University Press*, Princeton, NJ 1957.
- [2] N. Gillian, R. Knapp, and S. o'Modhrain, "Recognition Of Multivariate Temporal Musical Gestures Using N-Dimensional Dynamic Time Warping," *Proc. of the International Conference on New Interfaces for Musical Expression*, pp.337-342, 2011.
- [3] S. Cho, W. Bang, and J. Yang, "Two-stage Recognition of Raw Acceleration Signals for 3-D Gesture-Understanding Cell Phones," *Proc. of the 10th International Workshop on Frontiers in Handwriting Recognition*, 2006.
- [4] E. Keogh, M. Pazzani, "Scaling up dynamic time warping for datamining applications," *Proc. of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp.285-289, 2000.
- [5] M. Ko, B. West, S. Venkatesh, and M. Kumar, "Using dynamic time warping for online temporal fusion in multisensor systems," *Information Fusion* 9, pp.370-388 2008.
- [6] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing* 5, pp.657-675, 2009.
- [7] M. Muller, *Information Retrieval for Music and Motion*, Springer, 2007.
- [8] S. Kim, G. Park, S. Jeon, S. Yim, G. Han, and S. Choi, "HMM-based Motion Recognition with 3-D Acceleration Signal," *KIISE Transactions on Computing Practices and Letters*, Vol.15, No.3, pp.216-220, 2009.
- [9] S. Nam, J. Kim, S. Heo, and I. Kim, "Smartphone Accelerometer-Based Gesture Recognition and its Robotic Application," *KIPS Transactions on Software and Data Engineering*, Vol.2, No.6, pp.395-402, Mar., 2013.



권혁태

e-mail : ceo@dgmit.com

2001년 단국대학교 전산통계학과(학사)

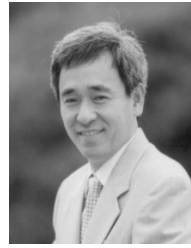
2003년 단국대학교 컴퓨터과학과(석사)

2007년 단국대학교 컴퓨터과학과 박사수료

현 재 디지엠펙정보기술(주) 대표이사

관심분야: 멀티스크린 융합모델 및

제스처 인식기반 사용자 경험



이석균

e-mail : sklee@dankook.ac.kr

1982년 서울대학교 경제학과(학사)

1990년 U. of Iowa, 전산학(석사)

1993년 U. of Iowa, 전산학(박사)

1993년~1997년 세종대학교 전임강사

현 재 단국대학교 소프트웨어학과 교수

관심분야: 데이터 모델, 불완전 정보관리, 시각질의어, 문서의
변화 탐지 및 버전 관리, 기계학습