

논문 2012-50-6-27

다중 클래스 SVM과 트리 분류를 이용한 제스처 인식 방법

(Gesture Recognition Method using Tree Classification and Multiclass SVM)

오 주 희*, 김 태 협*, 홍 현 기**

(Juhee Oh[Ⓞ], Taehyub Kim, and Hyunki Hong)

요 약

제스처 인식은 자연스러운 사용자 인터페이스를 위해 활발히 연구되는 중요한 분야이다. 본 논문에서는 키넥트 카메라로부터 입력되는 사용자의 3차원 관절(joint) 정보를 해석하여 제스처를 인식하는 방법이 제안된다. 대상으로 하는 제스처의 분포 특성에 따라 분류 트리를 설계하고 입력 패턴을 분류한다. 그리고 제스처를 리샘플링 및 정규화 하여 일정한 구간으로 나누고 각 구간의 체인코드 히스토그램을 추출한다. 트리의 각 노드별로 분류된 제스처에 다중 클래스 SVM(Multiclass Support Vector Machine)를 적용하여 학습한다. 이후 입력 데이터를 구성된 트리로 분류한 다음, 학습된 다중 클래스 SVM을 적용하여 제스처를 분류한다.

Abstract

Gesture recognition has been widely one of the research areas for natural user interface. This paper presents a novel gesture recognition method using tree classification and multiclass SVM(Support Vector Machine). In the learning step, 3D trajectory of human gesture obtained by a Kinect sensor is classified into the tree nodes according to their distributions. The gestures are resampled and we obtain the histogram of the chain code from the normalized data. Then multiclass SVM is applied to the classified gestures in the node. The input gesture classified using the constructed tree is recognized with multiclass SVM.

Keywords : Gesture recognition, Kinect, Classification tree, Multiclass SVM, Chain code

I. 서 론

제스처는 다양한 디바이스를 효과적으로 제어할 수 있는 가장 직관적이고 간단한 정보이다^[1~5]. 그리고 제스처 인식은 현재 인터랙티브 TV, 게임기, Robot 제어 등 다양한 매체에서 활용되고 있다^[1~3, 6]. 최근에는 간단한 제스처로 디바이스를 조종하는 것을 넘어, 자연스

러운 인터페이스를 지향하는 지능형(intelligent) 시스템의 입력 및 출력의 개발이 활발하게 진행되고 있다.

제스처 인식 연구는 2D 영상으로부터 사람의 피부색 정보나 시퀀스 상에서 차(difference) 영상을 이용하는 방식과 사용자에게 대한 거리 정보를 동시에 해석하는 3D 방식으로 크게 구분된다^[7]. 3D 데이터는 적외선 센서 등의 능동형과 스테레오 카메라 등을 이용하는 수동형 센싱 방법이 개발되고 있다.

스킨 컬러 값을 이용한 방법은 차 영상이나 얼굴검출기^[8]를 통하여 사람의 영역을 찾은 후에 사람의 얼굴을 제거한 후 손 영역만을 뽑아내어서^[9] 제스처를 검출한다. 그러나 차영상이나 스킨 컬러기반의 색상 검출은 빛에 많이 민감하고 움직임에 의한 블러 현상에 많은 영향을 받으며 피부색과 비슷한 영역이 있는 경우 잘못

* 학생회원, ** 평생회원, 중앙대학교 첨단영상대학원 영상학과

(Dept. of Imaging Science and Arts, GSAIM, Chung-Ang University)

※ 본 논문은 교육과학기술부 재원으로 학술연구재단의 지원을 받아 수행된 기초연구사업(2010- 0010398)임.

Ⓞ Corresponding Author(E-mail: saladin20@paran.com)
접수일자 2013년3월20일, 수정완료일 2013년5월20일

추출될 가능성이 있고 카메라 마다 피부색에 따른 컬러 값이 차이가 있기 때문에 컬러 값을 조정해줘야 된다는 단점이 있다^[10]. 또한 얼굴 검출기는 얼굴 방향에 따라 성능이 영향을 많이 받는다^[8].

스테레오 카메라를 이용한 방식은 2D 입력 영상과 함께 거리 정보 값을 이용할 수 있다. 일반적으로 얼굴과 몸통에 비해 손을 앞으로 뻗어 제스처를 표현하기 때문에 거리 값을 이용하여 해당 영역을 효과적으로 추출할 수 있다^[11~12]. 하지만 컬러 정보를 이용하며, 스테레오 영상간의 대응관계를 기반으로 하므로 색상과 주변 조명 등에 의해 영향을 많이 받는다.

제스처 인식 알고리즘에는 은닉 마코프 모델(Hidden Markov Model: HMM)^[4], 유한 상태 머신(Finite State Machine: FSM)^[5] 등의 다양한 방법이 활용된다. HMM은 제스처의 궤적과 잡음에 민감하여 안정화하는 작업이 필요하고 모든 가능성에 대한 확률 연산이 필요하기 때문에 인식 시간이 많이 소요된다. FSM은 공간 및 시간상의 상태(state)값으로 제스처를 인식하지만, 상태 값이 고정적으로 학습되어야 된다는 단점을 가지고 있다. 또한 제스처의 교차하는 부분이나 유사한 제스처간의 분석이 어려운 단점을 지니고 있다.

키넥트 카메라를 이용한 방식은 20개의 신체 관절(joint)에서 3D 각 축의 좌표 값을 추출하며, 최대 두 명까지 관절 데이터를 추출한다. 그러나 좁은 실내 영역에 제한적으로 사용되며, 유효 범위는 최대 8m 이내로 한정된다. 또한 반사되는 배경에서는 키넥트 센서의 적외선 패턴을 투과시키기 때문에 거리 값을 전혀 측정할 수가 없는 단점이 있다. 하지만 스킨 컬러기반이나 차영상 등의 방법에 비해 매우 정확하고 안정적으로 궤적 정보를 추출할 수 있다.

본 논문에서는 키넥트 센서를 이용하여 사용자의 관절에 대한 3D 정보를 추출한다. 그리고 제스처의 기본 패턴에 따라 사전에 분류 트리를 설계하고 각 제스처에 대한 체인코드의 특징 벡터를 추출한 다음, 다중 클래스 SVM (Multiclass SVM: MCSVM)으로 학습시킨 후 제스처를 분류하는 인식 알고리즘이 제안된다.

II. 본 론

데이터 입력은 키넥트의 3D 좌표 값 중에서 몸통의 중심(spin)부터 좌/우 손 끝까지의 데이터를 입력받았다. 특히 손보다는 손목 부분이 튜브 현상이 적었고 제스처의 궤적을 정확하게 표현하기 때문에 손목 관절 값을

사용한다. 제안된 방법은 오른쪽 손목의 입력 좌표 값으로 대상으로 특징별로 제스처를 트리 상에 분류한 다음, 체인코드로 특징을 추출하고 MCSVM을 사용하여 제스처를 최종 인식한다. 분류 트리를 일차적으로 사용하여 다중 클래스 분류기에서 고려되는 특징 파라미터를 조절함으로써 보다 인식 성능을 개선할 수 있다.

1. 제스처 분류 트리

대상으로 하는 제스처의 특성을 고려하여 3개의 레벨로 구성된 이진(binary) 트리를 구성하였으며, 입력되는 3D 제스처 궤적은 트리에 따라 분류된다. 각 단계별로 제스처의 시작 방향, 제스처의 시작과 끝이 일치하는지 여부, 각 제스처의 중점을 기준으로 최종 끝점의 높이를 이용하여 트리의 노드 분포를 판단한다.

가. 시작 방향의 분류

사용자가 취하는 제스처가 시작점에서부터 출발하여 왼쪽 또는 오른쪽으로 이동하는지를 식 (1)을 이용하여 판단한다. 여기서 SP는 시작 포인트를 의미하며 SP.x는 시작 포인트의 x좌표를 의미한다. 시작 포인트의 다음(SP+1)포인트의 x좌표에서 시작 포인트 SP의 x좌표의 위치 값을 뺀다. 이 값이 양수라면 오른쪽으로 이동을, 음수라면 왼쪽으로 이동한 것이므로, 이진 트리에서 양수면 오른쪽 노드로, 음수면 왼쪽 노드로 결정한다.

$$\begin{aligned} (SP + 1).x - SP.x < 0 &\rightarrow \text{Left} \\ (SP + 1).x - SP.x > 0 &\rightarrow \text{Right} \end{aligned} \quad (1)$$

제스처의 시작과 끝은 대기 상태이며, 시작과 끝의 경우는 잡음이 심하기 때문에 본 논문에서는 첫 번째 프레임이 아닌 세 번째 프레임을 시작프레임으로 결정한다. 또한 잡음 등을 고려하기 위해 샘플링 구간도 적절하게 조절할 수 있다.

나. 시작 및 끝점의 일치 여부

제스처의 시작 포인트와 끝 포인트의 일치 여부는 식 (2)를 이용하여 판단한다. 여기서 SP는 시작 포인트를 나타내며 SP.x는 시작 포인트의 x좌표, SP.y는 시작 포인트의 y좌표를 각각 나타낸다. EP는 끝 포인트를 나타내며, EP.x는 끝 포인트의 x좌표, EP.y는 끝 포인트의 y좌표를 나타낸다. SP와 EP의 x, y좌표를 이용하여 시작점과 끝점의 거리 값을 계산한다. 계산된 거리 값이 일정 임계값 이하면, 시작점과 끝점이 일치하는 제스처로 판단한다.

$$distance = \sqrt{(SP.x - EP.x)^2 + (SP.y - EP.y)^2} \quad (2)$$

다. 끝점의 높낮이

각 제스처의 전체 프레임 개수를 n 개라고 할 때, 각 포인트의 x 좌표와 y 좌표를 더한 후에 전체 프레임 개수 n 개로 나누어 각 제스처의 중점(Centroid)을 식 (3)으로 구한다. 그리고 계산된 중점의 y 좌표와 최종 끝포인트(EP)의 y 좌표를 비교하여 높낮이를 식 (4)로 판단한다. 구성된 분류 트리에서 중점보다 끝점이 높으면 참으로, 그렇지 않은 경우는 거짓으로 판단하여 제스처를 최종 리프 노드에 분류한다.

$$Centroid = \frac{1}{n} \sum_{i=0}^{n-1} (x_i, y_i) \quad (3)$$

$$\begin{aligned} EP.y > Centroid.y &\rightarrow true \\ EP.y < Centroid.y &\rightarrow false \end{aligned} \quad (4)$$

2. 제스처의 리샘플링과 정규화

사용자가 하나의 제스처를 시작하는 부분과 끝내는 영역은 상대적으로 잡음이 심하며, 트리 분류 및 체인코드 히스토그램 추출 단계에서 오차율이 높은 편이다. 따라서 각 제스처를 동일한 구간으로 리샘플링하면서 제스처 궤적 위치를 보간(interpolation)하며, 최종 단계에서 전체 제스처 길이를 1로 정규화 한다^[7, 13].

대상 제스처의 각 포인트 사이의 거리 값을 구하여 전체 길이를 구한 다음, 식 (5)와 같이 리샘플링 할 개수를 M 이라고 할 때 $M-1$ 값으로 나누어 리샘플링 될 각각의 포인트 사이의 거리 값 $Length_{sampling}$ 을 계산한다. 유클리디언 거리 함수(distance)를 이용하여 손목의 이전 위치와 현재의 좌표간 거리 값을 구한다. 궤적 시퀀스를 따라 진행하면서 식 (6)로 거리 값의 누적 값 $Length_{accum}$ 을 구하고 이 $Length_{accum}$ 값이 $Length_{sampling}$ 길이보다 같거나 커지는 경우, 바로 이전의 관절의 위치와 현재의 좌표 값을 구하여 선형보간 함으로써 리샘플링 구간의 위치를 구한다.

$$Length_{sampling} = \frac{1}{M-1} \sum_{i=1}^n dis(joint_i - joint_{i-1}) \quad (5)$$

$$Length_{accum} = \sum distance(joint_i - joint_{i-1}) \quad (6)$$

제스처의 리샘플링 개수는 16, 32, 64, 128 까지 구분하여 실험을 진행하였으며, 실험적으로 64인 경우에 상대적으로 높은 인식률이 얻어졌다. 즉, 리샘플링을 많이 한다고 해도 정확한 성능이 높은 것은 아니다. 그 이유는 리샘플링 수치가 높으면 해석 구간이 지나치게 짧아지며, 이에 따라 체인코드가 지나치게 세분화되어 전체적인 제스처 궤적의 분포를 효과적으로 표현하기 어렵기 때문이다. 또한 X, Y, Z 축에서 대상 제스처의 가로/세로/높이의 길이를 구한 다음, 가장 길이가 긴 축의 길이를 1로 놓고 그 축을 기준으로 다른 길이를 1로 놓은 길이의 비율에 맞게 정규화 한다. 그럼 모든 제스처가 가장 긴 길이가 1이 되어 제스처의 크기가 비슷하게 구성이 된다.

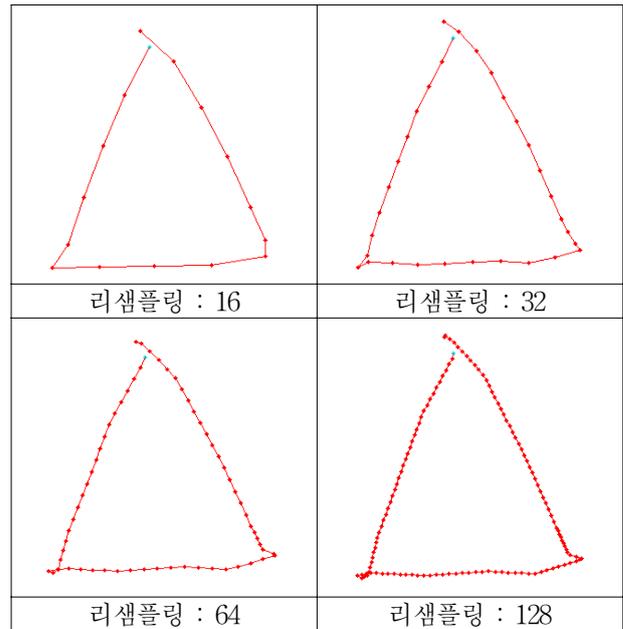


그림 1. 삼각형 제스처 리샘플링

Fig. 1. Resampling triangle gesture.

3. 체인코드 특징 추출

본 논문에서는 리샘플링된 제스처 세그먼트로부터 체인코드를 추출하며, 특징을 정확하게 기술하기 위해 전체 제스처 궤적을 일정 간격으로 균등하게 분할한다. 분할된 구간은 앞 절에서 리샘플링된 다수의 세그먼트로 구성되며, 각 구간에서 제스처의 체인코드와 히스토그램을 구하여 인식을 위한 특징 벡터로 사용한다.

먼저 구간 길이를 구하기 위해 식 (7)과 같이 제스처

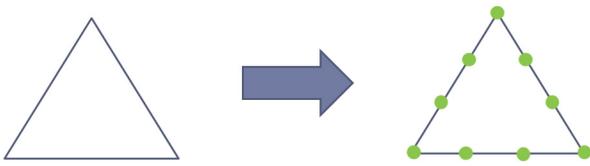


그림 2. 구간을 균등하게 나눈 삼각형 제스처
Fig. 2. triangle gesture with evenly divided intervals.

의 각 포인트를 길이 값을 누적하여 전체 길이를 구해 준 후, 히스토그램을 구할 구간의 개수(K)만큼 분할한 구간의 길이($plength$)를 구하여, 제스처의 궤적을 분할한다. 본 실험에서는 최적의 체인코드 특징 벡터를 찾기 위해 구간을 3, 6, 9, 12 구간 등으로 전체 궤적을 분할하여 체인코드 넘버링의 히스토그램을 구간별로 구한 후 인식 성능을 확인하여 구간의 개수 결정에 따른 결과를 분석하였다. 삼각형 제스처 패턴을 9 구간으로 나눈 예를 그림 2에 나타내었다.

$$plength = \frac{1}{K} \sum_{i=0}^n \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (7)$$

제스처 패턴을 기술하기 위한 체인 코드는 8 방향의 체인코드를 사용하였다. 일정하게 나눈 구간별로 제스처 패턴에 대한 체인코드 분포를 정규화 된 히스토그램으로 표현한다. 즉, 각 구간에서 체인코드 분포에 대한 빈(bin) 수를 구하고 이를 정규화 한다. 예를 들어 9개의 구간으로 분할된 경우 각 구간마다 한 개씩, 총 9개의 체인코드 히스토그램을 생성하여 해당 제스처를 기술한다.

4. MCSVM로 제스처의 학습 및 분류

대상 제스처에 대한 체인코드와 히스토그램 정보를 특징 벡터로 이용하여 MCSVM를 학습한다. 세 개 이상의 특징 벡터를 분류하도록 고안된 MCSVM은 여러 클래스를 한꺼번에 학습 및 분류하기 때문에 다양한 클래스를 구분하는 적합한 분류 방법이다^[14~15].

가. 다중 클래스 SVM

다중 클래스 SVM은 이진기반의 분류방법인 one-against-all, one-against-one, DAGSVM 등이 있으며 모든 분류를 한꺼번에 고려하는 방법인 Weston's multiclass SVM, Crammer's Multiclass SVM 등으로 구분된다^[14~15]. 본 논문에서는 분류 트리의 최종 리프 노드에 다수의 제스처가 존재하므로 one-against-all 방

법을 구현하였다.

one-against-all 방법은 분류해야 하는 제스처 클래스가 M 개일 때 학습 데이터를 대상으로 M 번 학습한 다음, 이후 입력 테스트 데이터를 비교하여 가장 높은 판별 함수 값을 갖는 클래스 레이블을 할당한다. 판별 함수는 식 (8)이며, 여기서 C_i 가 k 클래스 여부에 따라 판별 함수 y_i 가 다음과 같이 출력 값을 리턴한다.

$$y_i = \begin{cases} +1 & \text{if } C_i = k \\ -1 & \text{if } C_i \neq k \end{cases} \quad (8)$$

나. 제스처의 학습 및 분류

분류하기 위해 너무 많은 특징을 이용하거나 구분해야 하는 패턴이 너무 많은 경우, 일반적으로 선형 분류기는 정확한 인식 성능을 기대하기 어렵다. 따라서 본 논문에서는 대상으로 하는 제스처 분포에 따라 설계된 이진 트리로 제스처를 구분한 다음, 다중 클래스 SVM을 적용한다. 즉, 학습 단계에서 서로 다른 제스처이지만, 특징벡터가 비슷하여 오인식될 경우, 먼저 트리로서 제스처들을 다른 노드로 구분함으로써 다중 클래스 SVM의 인식 성능을 개선한다.

학습단계에서 이진 트리를 이용하여 제스처를 분류한 다음, 리프 노드별로 다중 클래스 SVM을 적용한다. 그러나 실제 사용자의 입력 데이터를 분류하면서 학습되지 않은 제스처가 다중 클래스 SVM에 들어가면, 오인식이 발생한다는 단점이 발생한다.

본 논문에서는 이런 단점을 보완하기 위해 다중 클래스의 최종 점수(score)를 확인하였다. 즉, 오인식되는 경우는 정확하게 분류하는 경우보다 일반적으로 점수 값이 상대적으로 낮다는 것을 확인하였다. 따라서 일정한 점수 이하에서는 전체 제스처를 대상으로 한꺼번에 학습시킨 다중 클래스 SVM으로 분류하는 방법과 결과를 III장에서 비교한다.

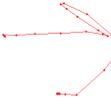
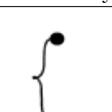
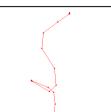
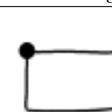
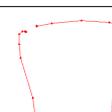
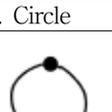
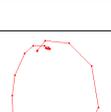
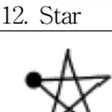
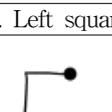
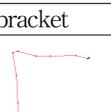
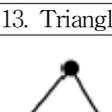
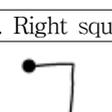
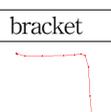
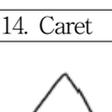
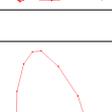
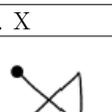
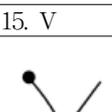
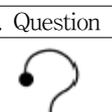
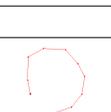
III. 실험

실험환경은 Intel Core2 Quad CPU Q8400, 메모리 4기가, 32비트 윈도우 7 Ultimate K Service Pack 1, Xbox 360용 키넥트 카메라 환경에서 실험하였다. 실험에 참여한 사람의 수는 모두 8명으로 연령은 20~30대 중반까지 추출하였고, 남자 7명, 여자 1명으로 구성되었다. 그중 남자 4명과 여자 1명의 데이터를 기존 알고리즘^[7]과 제안 알고리즘 모두 훈련 및 비교 셋으로 사용

하였으며 그 수는 750개(15개 제스처에 대해 50개씩)이다. 남자 3명의 데이터로 테스트 셋을 사용하였으며 그 수는 450개(15개 제스처에 대해 30개씩)이다.

개별 제스처는 Kinect 카메라로 초당 평균 30프레임씩 관절 좌표를 저장하였으며, 한 번에 하나의 제스처를 각각 입력받아 한사람에 대해 한 제스처 당 10번의 데이터를 저장하였다. 사용된 제스처 셋은 표 1과 같고,

표 1. 제스처 패턴^[7, 13]
Table 1. Gesture patterns^[7, 13].

제스처 번호와 이름		제스처 번호와 이름	
제스처	입력 데이터	제스처	입력 데이터
1. Arrow		9. Pigtail	
			
2. Right curly brace		10. Delete	
			
3. Left curly brace		11. Rectangle	
			
4. Circle		12. Star	
			
5. Left square bracket		13. Triangle	
			
6. Right square bracket		14. Caret	
			
7. X		15. V	
			
8. Question			
			

시작 지점을 검은 점으로 나타내었다. 각 파라미터 설정에 따라 다양한 비교 실험을 진행하였다.

1. 구간별 실험 진행

제스처의 구간을 3, 6, 9, 12 개로 나누어 실험을 진행하였으며, 각각 분류 소요시간이 0.342228ms, 0.336068ms, 0.336453ms, 0.348387ms로 구간의 크기가 수행 속도에 크게 영향을 미치지 않음을 확인하였다. 그 이유는 구간의 크기만큼 히스토그램을 생성하지만 히스토그램의 수와 상관없이 제스처를 전체 포인트의 수를 동일하게 리샘플링 하였기 때문에 bin의 수는 모두 같기 때문에 연산 속도에 영향을 크게 미치지 않는다. 또한 인식을 역시 구간별로 약간의 차이는 있었지만, 네 가지 경우에 대해 평균 98.33, 97.33, 99.00, 98.67%의 높은 인식 성공률을 확인하였다. 이후 실험에서는 전체 제스처 구간을 9개로 분할하고 인식 실험을 진행한다.

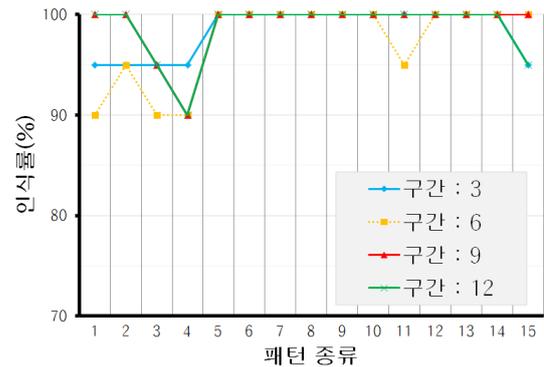


그림 3. 구간별 인식률
Fig. 3. Recognition rate by section.

2. 리샘플링 수에 따른 실험 진행

기존 연구에서 정확도를 높이기 위해 활용된 리샘플

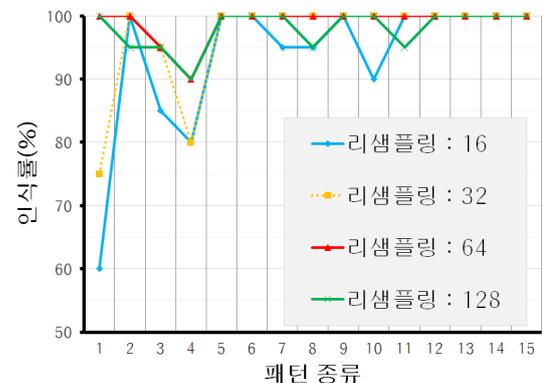


그림 4. 리샘플링 수에 따른 인식률
Fig. 4. Recognition rate by the number of resampling.

링별 실험을 다음과 같이 진행한다. 기존 연구에서 리샘플링 수를 16~256으로 설정하여 실험하였고, 32에서 가장 좋은 결과를 얻었다^[7, 13].

본 실험에서는 16, 32, 64, 128개로 리샘플링을 하는 실험을 수행하였다. 연산 시간은 리샘플링 개수가 증가할수록 연산 시간이 오래 걸리지만, 최대 0.1ms 이하의 미비한 속도차이가 발생함을 확인하였다. 또한 그림 4와 같이 리샘플링을 적게 할수록 인식률이 떨어지고, 특정 수치 이상이면 인식률이 오르다가 다시 감소하는 경향을 알 수 있다. 즉, 네 가지 경우에 대해 평균 93.67, 96.67, 99.00, 98.00%의 인식 성공률을 보인다. 총 4번의 실험 중 64로 리샘플링하는 것이 가장 좋은 인식률을 보이는 것을 확인하였다.

3. 학습방법에 따른 실험 진행

제안된 방법의 성능을 비교하기 위해 기존 알고리즘을 구현하였으며, 블록도를 그림 5에 나타내었다. 또한

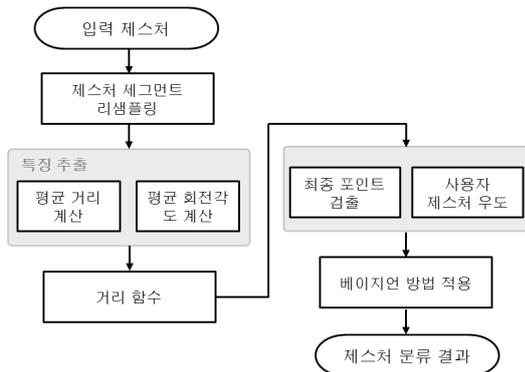


그림 5. 기존 알고리즘^[7, 13]
Fig. 5. Previous algorithm^[7, 13].

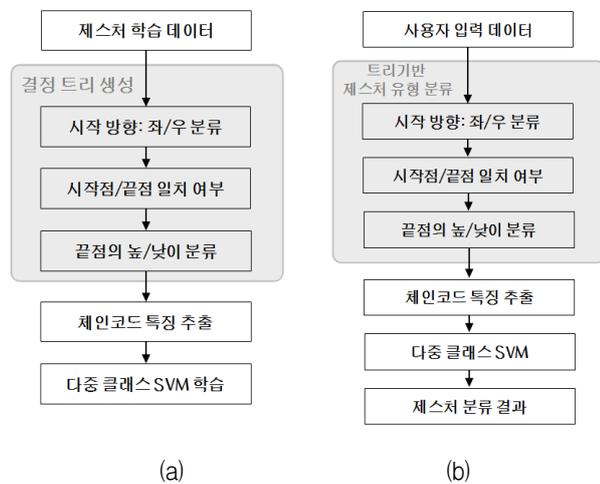


그림 6. 제안된 알고리즘의 (a) 학습 및 (b) 분류과정
Fig. 6. (a) Training and (b) classification steps of proposed algorithm.

제안된 시스템은 기존 논문과 마찬가지로 리샘플링과 정규화 부분을 적용하였지만, 트리를 이용해 제스처를 분류하고 특징을 추출하고 인식하는 접근방법은 다르다. 제안된 방법의 블록도를 학습 및 분류 단계로 구분하여 그림 6에 보였다.

기존 방법과 함께 제안된 방법을 세 가지로 구분하여 얻어진 인식 성공률과 기존 방법을 표 2에 나타내었다. 즉, 분류 트리 없이 전체 제스처를 다중 클래스 SVM으로 학습하고 인식하는 경우(NonTree), 트리를 이용하여 분류하고 각 노드별로 할당된 제스처를 대상으로 다중 클래스 SVM을 적용한 경우(Tree)를 실험한다. 세 번째 경우(Tree+NonTree), 입력 제스처를 트리으로 분류하고 다중 클래스 SVM에서 임의의 점수 값 이하(실험에서는 80 이하)인 경우를 판단하여 첫 번째인 다중 클래스 SVM로 분류하고, 이상인 경우는 트리로 분류된 다중

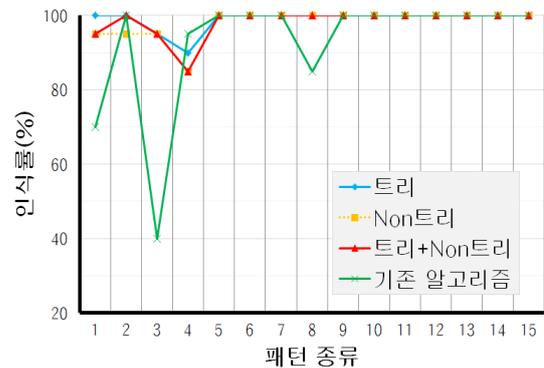


그림 7. 학습 알고리즘 별 인식률
Fig. 7. Recognition rate by the training algorithm.

표 2. 학습 알고리즘 별 인식률(%)
Table 2. Recognition rate by the training algorithm(%).

	Tree	NonTree	Tree+NonTree	기존 알고리즘
1	100	95	95	70
2	100	95	100	100
3	95	95	95	40
4	90	85	85	95
5	100	100	100	100
6	100	100	100	100
7	100	100	100	100
8	100	100	100	85
9	100	100	100	100
10	100	100	100	100
11	100	100	100	100
12	100	100	100	100
13	100	100	100	100
14	100	100	100	100
15	100	100	100	100
평균	99.00	98.00	98.33	92.67

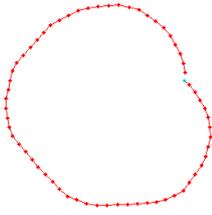


그림 8. 잘못 인식된 제스처 예
Fig. 8. Incorrectly recognized gesture example.

클래스 SVM을 적용하여 인식 성능을 확인하였다.

실제 입력 제스처가 이진 트리에 의해 사전에 분류된 학습 데이터와 다른 경우에는 제스처의 인식률이 영향 받을 수 있다. 표 2의 결과로부터 분류 트리과 다중 클래스 SVM을 적용하는 경우가 인식률이 상대적으로 높았다. 그러나 이후 다양한 사용자에 따라 학습되지 않은 제스처가 분류 트리에 입력된다면, 인식 성능이 낮아지는 단점이 있다. 따라서 제스처 특성에 따라 분류 트리를 보다 정확하게 설계해야 한다.

기존 알고리즘과의 비교 실험을 통해 제안된 방법의 인식 성능이 개선되었음을 확인하였다. 표 2에서 기존 알고리즘은 평균 92.67%의 인식률을 나타내는 반면, 제안된 방법은 99%의 인식률을 보인다.

개별적인 제스처 패턴 중에는 Circle 제스처에 대한 기존 방법의 인식률이 상대적으로 높았다. 그 이유는 기존 논문과 제안된 알고리즘 역시 같은 시작점에서 같은 방향의 제스처로 그려진다는 전제조건이 필요한데, 잘못 인식된 제스처는 그림 8와 같이 12시 방향이 시작점이 아닌 2~3시 사이에 시작점이 존재하여 체인코드의 히스토그램 분포 상에 차이가 발생한다.

IV. 결 론

본 논문에서는 키넥트로부터 입력되는 3D 제스처 데이터를 분류 트리과 다중 클래스 SVM을 적용하는 제스처 인식 방법이 제안되었다. 실험결과로부터 제안된 방법의 개선된 성능을 확인하였다.

그러나 체인 코드를 사용하여 제스처의 특징 벡터를 대체적으로 정확하게 추출하지만, 시작점이 동일하고 같은 방향으로 진행되는 제스처에 대해 양호한 성능을 보인다는 단점도 확인하였다. 또한 다중 클래스 SVM은 여러 제스처를 한꺼번에 분류할 수 있는 편리함도 있지만, 너무 많은 특징 벡터가 있는 경우에는 오분류될 가능성도 동시에 높아진다. 또한 다양한 사용자에 의해 얻어진 많은 데이터를 학습 및 입력 집합으로 구

성하여 제안된 방법의 인식 성능을 보다 정확하게 비교할 예정이다.

REFERENCES

- [1] M. Chen, L. Mummert, P. Pillai, A. Hauptmann, and R. Sukthankar, "Controlling your TV with gestures," Proc. Int'l. Conf. Multimedia Information Retrieval, pp. 405-408, 2010.
- [2] H. P. Jain and A. Subramanian, "Real-time upper-body human pose estimation using a depth camera," Technical Report, HPL-2010-190, HP Laboratories, 2010.
- [3] S. Lin, Y. Lai, L. Chan, and Y. Hung, "Real-time 3D model-based gesture tracking for multimedia control," Proc. Int'l. Conf. Pattern Recognition, pp. 3822-3825, 2010.
- [4] H. Lee and J. H. Kim, "An HMM-based threshold model approach for gesture recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 10, pp. 961-973, 1999.
- [5] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," Proc. of IEEE Int'l. Conf. on Automatic Face and Gesture Recognition, pp. 410-415, 2000.
- [6] T. Lourens, R. Berkel, and E. Barakova, "Communicating emotions and mental states to robots in a real time parallel framework using Laban movement analysis," Journal Robotics and Autonomous Systems, vol. 58, issue 12, pp. 1256-1265, 2010.
- [7] P. O. Kristensson, T. Nicholson, and A. Quigley, "Continuous recognition of one-handed and two-handed gestures using 3D full-body motion tracking sensors," Proc. of ACM Int'l. Conf. on Intelligent User Interfaces, pp. 89-92, 2012.
- [8] P. Viola and M. Jones, "Robust real-time object detection," Journal of Computer Vision, vol. 57, no. 2, pp. 137-154, 2004.
- [9] V. Vezhnevets, V. Sazonov, and A. Andreeva, "A survey on pixel-based skin color detection techniques," Proc. of the GraphiCon, pp. 85-92, 2003.
- [10] 조오영, 김형곤, 고성제, 안상철, "인터랙티브 가상 환경을 위한 손 제스처 인식 시스템," 대한전자공학회, 전자공학회논문지-S, pp. 70-82, 1999.
- [11] S. Kang, A. Roh, and H. Hong, "Using depth and skin color for hand gesture classification," Proc. of IEEE Int'l. Conf. on Consumer

- Electronics, pp. 155-156, 2011.
- [12] H. Francke, J. Ruiz-del-Solar, and R. Verschae, "Realtime hand gesture detection and recognition using boosted classifiers and active learning," Lecture Notes in Computer Science, vol. 4872, pp. 533-547, 2007.
- [13] P. O. Kristensson and L. C. Denby, "Continuous recognition and visualization of pen strokes and touch-screen gestures," Proc. of Eurographics Symposium on Sketch-Based Interfaces and Modeling, pp. 95-102, 2011.
- [14] L. Zhibin and J. Lianwen, "LATTICESVM - A new method for multi-class Support Vector machines," IEEE Int'l. Joint Conf. on Neural Networks, pp. 727-733, 2008.
- [15] 박지영, 홍태호, "다분류 SVM을 이용한 DEA기반 벤처기업 효율성등급 예측모형," Asia Pacific Journal of Information Systems, vol. 19, no. 2, pp.139-155, 2009.

저 자 소 개



오 주 희(학생회원)
2007년 공주대학교 컴퓨터공학과
학사
2011년~현재 중앙대학교 첨단영
상대학원 영상학과 석사
과정

<주관심분야: 컴퓨터비전, 패턴인식 등>



김 태 협(학생회원)
2010년 안양대학교 컴퓨터공학과
학사
2011년~현재 중앙대학교 첨단영
상대학원 영상학과 석사
과정

<주관심분야: 컴퓨터비전, 패턴인식 등>



홍 현 기(평생회원)-교신저자
1998년 중앙대학교 전자공학과
박사
1998년~1999년 서울대학교 자동
제어특화연구센터 연구원
2002년~2003년 Post-doc., Univ.
of Colorado at Denver

2000년~현재 중앙대학교 첨단영상대학원
영상학과 교수 재직 중

<주관심분야: 컴퓨터비전, 패턴인식, 증강현실
등>