손영역 획득과 손동작 인식에 의한 제스처 기반 사용자 인터페이스의 구현

Gesture-based User-Interface Through Hand-region Detection and Hand-movement Recognition

고 일 주*, 배 영 래**, 최 형 일*
Il-Ju Ko, Young-Lae Bae, Hyung-Il Choi

요 약 본 논문은 컴퓨터 시각을 이용하여 제스처를 인식함으로써 사용자에게 보다 편리한 인터페이스를 제공하는 것을 목표로 한다. 제안하는 제스처 인식 방법은 손영역을 획득하는 손영역 획득 모듈과 손영역을 인식하는 인식 모듈로 나누어 수행한다. 손영역 획득 모듈에서는 손색상 모델과 손색상 결정함수를 정의하여 칼라영상의 영역 분리를 수행하였고, 칼만필터를 이용하여 손색상 모델을 갱신하고 탐색영역을 제한하여 영역 추적을 용이하게 하였다. 영역 추적은 전 시점의 손영역 정보를 이용하여 현 시점의 손영역을 획득한다. 인식 모듈에서는 정적인 제스처를 표현하는 객체 프레임과 행동 프레임, 그리고 동적인 제스처를 표현하는 스키마를 정의한다. 그리고 획득된 손영역과 정합을 수행함으로써 제스처를 인식한다.

실험 결과로는 제안하는 제스처 기반 인터페이스를 적용한 삼목(Tic-Tac-Toe) 게임 프로그램을 구현하였다. 사용자는 제스처를 이용하여 컴퓨터와 게임을 진행한다. 제안하는 시스템은 다른 종류의 게임 프로그램이나 마우스의 역할을 수행하는 윈도우 시스템의 제어, 그리고 가상 현실 시스템에 적용될 수 있다.

주제어 : 컴퓨터 시각, 제스처 인식, 영역 분리, 칼만필터, 객체 프레임, 행동 프레임, 스키마, 제스처 기반 사용자 인터페이스

1. 서 론

인간은 사회 구성원들간의 의사소통을 통해 삶을 살아간다. 언어가 생성되기 이전의 원시 인류는 몸이나 손의 제스처에 의지하여 구성원들간의 의사를 전달하였다. 그 후, 언어가 발생되면서 의사소통 도구로써의 제스처는 중요한 역할을 수행하지 못하게 되었다. 그러나 인간은 제스처를 이용한 대화에 대한 이해 능력과 직관을 소유하고 있기 때문에 보조적인 의사 전달 수단으로 제스처의 역할을 수정하여 시각적인 언어인 제스처를 의사 전달 수단으로 사용한다. 이러한 특성을 갖고 있는 제스처를 이용하여 사용자와 컴퓨터간의 의사소통에도 제스처를 사용하려는 연구가 많이 수행되어 왔다[1, 2]. 특히 가상 현실 시스템에는 사용자와의 상호작용이 매우 중요하기 때문에 이 분야를 연구하는 사람들이 제스처의 인식에 대하여 많이 연구하였다[3].

제스처를 인식하기 위해서는 먼저 3차원 공간에서 사용자의 제스처를 입력받아야 하는데, 어떻게 입력을 받아들여야 하는 가에 대한 문제가 중요한 연구 분야로 알려져 있다. 현재까지 널리 알려진 방법으로는 전자 장갑을 입력 장치로 하는 방법과 컴퓨터 시각 시스템을 입력 장치로 하는 방법이 있다. 전자 장갑은 정확한 손의 움직임을 입력받을 수 있기 때문에 가상 현실 시스템에 많이 사용되지만, 전자 장갑을 착용하여야만 한다는 부담감은 자연스럽고 편안한 인터페이스를 원하는 사용자의 요구를 만족시킬 수 없다. 컴퓨터 시각 기반 인식 방법은 이러한 사용자가 다른 부가적인 장치를 착용하여야만 한다는 단점을 극복하고 보다 자연스러운 인터페이스를 제공할 수 있다. 그러나 이 방법은 해결해야 하는 여러 가지 문제점들을 갖고 있다. 복잡한 배경으로부터 움직이

* 숭실대학교 컴퓨터학부
** 시스템공학연구소 영상정보처리실

는 손영역을 획득하는 문제, 손영역의 위치를 추적하는 문제, 손 모양 인식에 관련된 문제, 그리고 손의 움직임에 대한 인식 문제 등이 있다. 본 논문에서는 이러한 문제점들을 해결하여 제스처 기반 인터페이스를 구현하는 것을 목표로 하였다. 문제점들의 해결 방법으로 제스처 인식을 위해 영상처리를 담당하여 손영역을 획득하는 모듈과 손영역에 대한 지식을 이용하여 손동작을 인식하는 모듈을 제안한다.

연속적으로 카메라로부터 입력되는 칼라영상에서 손영역을 획득하기 위해 손색상 모델과 손색상 결정함수를 이용하여 칼라영상에 대한 이진화와 레이블링을 수행하고 손색상 모델의 갱신과 손영역 분리를 위한 탐색영역을 제한하기 위해 칼만필터를 사용한다. 손색상 모델은 손색상 모델의 학습에 의해 획득된 YIQ 칼라공간의 I 요소와 Q 요소의 평균과 편차로 정의한다. 손영역은 탐색영역내에서 추출되는 영역들을 전 시점의 손영역에 대한 정보를 이용하여 추적함으로써 획득된다. 손영역 획득 모듈은 이러한 일련의 절차를 거쳐 현 시점의 손영역을 획득하는 역할을 수행한다.

획득된 손영역으로부터 제스처를 인식하기 위해, 인식 모듈에서는 프레임 지식 표현 방법을 사용하여 정적인 제스처와 동적인 제스처를 모델링한다. 프레임 지식은 손모양에 대한 지식, 즉 정적인 제스처에 대한 지식을 표현하는 객체 프레임과 객체의 움직임에 대한 지식을 표현하는 행동 프레임, 그리고 동적인 제스처에 대한 지식을 표현하는 스키마로 나누어 정의한다. 그리고 인식 모듈에서는 입력 영상으로부터 획득되는 손영역과 프레임 지식을 정합하기 위해 필요한 특징들을 계산하여 제스처에 대한 인식을 수행한다. 제어 모듈은 손영역 획득 모듈과 인식 모듈을 제어하며, 카메라로부터 입력 영상을 받아들여 손영역 획득 모듈에 전달하고 제스처 인식의 결과를 사용자에게 전달하여 주는 역할을 수행한다. 그림 1은 본 논문에서 제안하는 시스템에 대한 개략적인 구조를 보인다.

제안된 시스템은 영상 특징에 기반한 컴퓨터 시각을 이용하여 복잡한 배경 영상으로부터 손영역을 획득하고, 다음 단계에서 획득된 손영역에서 특징을 추출하여 프레임 지식과 정합을 수행하고 적절한 스키마를 활성화하여 제스처를 인식한다. 본 논문에서는 YIQ 칼라공간을 사용하는 손색상 모델과 손색상 결정함수를 정의하여 칼라영상의 이진화 기법을 제안하였다. 그리고 손영역의 추적에 칼만필터를 적용하여 손색상 모델의 갱신과 탐색영역의 축소를 수행하였다. 손색상 모델을 시간의 흐름에 따라 갱신함으로써 손색상 모델이 상황에 적응적인 특징을 갖게 된다. 탐색영역의 축소는 영역 분리 속도를 빠르게 할뿐만 아니라 탐색영역 내에서만 손영역이 될 가능성이 있는 영역을 획득하기 때문에 손영역 획득의 오류를 줄인다.

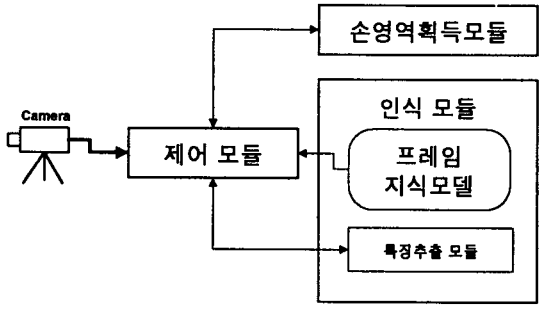


그림 1 제안하는 시스템의 개략적인 구조

프레임 지식은 영상 처리를 수행하기에 편리한 구조를 갖고 있기 때문에 컴퓨터 시각 인식을 위한 지식 표현으로 많이 사용된다[4, 5]. 그러나 대부분의 프레임 기반 시스템들은 동적으로 변화하는 영상보다는 정적인 영상의 인식에 사용되었다. 예를 들면, SIGMA 시스템[6]은 항공 영상의 인식에 프레임 지식을 사용하였다. 이처럼 정적인 영상들을 인식하기 위해 프레임 지식을 사용하였기 때문에 동적인 영상에 대한 인식은 SIGMA 시스템에서 제안하는 프레임 지식 표현 방법만으로는 쉽지 않다. 이 문제를 해결하기 위해 본 논문에서는 행동 프레임을 정의하였고, 객체 프레임과 행동 프레임의 순차적인 구조를 갖는, 즉 동적인 제스처를 나타내는 스키마를 정의하였다.

본 논문에서 제안하는 제스처 기반 인터페이스를 구현한 실험 결과를 보이기 위해 사용자의 제스처로 삼목(Tic-Tac-Toe) 게임을 할 수 있는 응용 시스템을 구현한다. 삼목 게임 프로그램에 적용된 제스처 기반 인터페이스는 유사한 다른 종류의 게임 프로그램에 적용될 수 있을 뿐 아니라 윈도우 시스템의 제어, 그리고 더욱 다양한 제스처를 인식하는 시스템으로 확장될 수 있다.

본 논문의 구성은 5장으로 구성되며 각 장의 주

요 내용은 다음과 같다. 제 2 장에서는 영역 추적에 의한 손영역 획득에 대하여 기술한다. 이장에서는 칼라영상의 이진화를 위한 손색상 모델에 대한 정의와 학습 방법, 칼만필터를 이용한 손색상 모델의 갱신과 탐색영역 축소, 그리고 영역 추적에 의한 손영역의 획득에 대하여 기술한다. 제 3 장에서는 프레임 지식을 이용한 제스처 인식에 대하여 기술한다. 4 장에서는 컴퓨터 시각을 이용한 제스처 기반 인터페이스를 적용한 시스템을 구현하여 실험 결과를 보인다. 마지막으로, 5 장에서는 결론 및 향후 연구 방향에 대하여 논한다.

2. 영역 추적에 의한 손영역 획득

2.1 손색상 모델과 손색상 결정함수

인간의 시각이 물체를 구분할 수 있는 가장 큰 특징 요소 중의 하나가 색상이므로 본 논문에서는 칼라정보를 이용하여 손영역을 획득한다. 일반적으로 칼라영상을 코딩하기 위해서 RGB 칼라공간을 많이 사용하지만, 영상 처리를 수행하려 할 때 세 가지의 RGB 칼라 값을 모두 처리해야 하기 때문에 RGB 칼라공간을 사용하여 영상 처리를 수행하는 것은 비효율적이다. 그리고 RGB 칼라공간은 영상의 밝기 변화에 민감하게 작용할 뿐 아니라 동일한 영상의 동일한 영역에 대하여서도 조명의 크기나 방향에 따라 칼라 값이 차이가 크게 발생할 수 있다. 그러나 YIQ 칼라공간은 밝기의 변화에 대하여 색상이나 채도가 변하지 않고 동양인의 손색상을 구분하기에 적합한 방법이다. 따라서 RGB 칼라영상을 NTSC(National Television System Committee)에서 정한 합성 칼라 비디오 표준 방식인 YIQ 칼라공간으로 변환하여 영상 처리에 이용한다[7, 8]. YIQ 칼라공간은 YUV 칼라공간에 의해 유도된 것으로써 Y(Luminance)요소는 밝기 값을 나타내고, I(Inphase)요소와 Q(Quadrature)요소는 두 요소를 합성하여 색상과 채도를 나타낸다. 식 (1)은 RGB 칼라공간을 YIQ 칼라공간으로 변환하는 식을 보인다[9].

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.30 & 0.59 & 0.11 \\ 0.60 & -0.27 & -0.32 \\ 0.21 & -0.52 & 0.31 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

손색상 모델은 사용자와의 상호작용에 의한 학습을 통해 획득된 I 요소와 Q 요소의 평균과 표준편차를 이용하여 식 (2)와 같이 정의한다. 손색상 모델은 칼라영상에서 손색상을 추출하기 위해 정의한다. 학습된 I 요소와 Q 요소의 평균과 표준편차가 정규분포를 갖는다고 가정을 할 때, 화소의 I 요소 값과 Q 요소의 값이 $\mu+3\sigma$에서 $\mu-3\sigma$ 까지의 구간에 속하는가를 판별하여 손색상에 해당하는 화소를 획득한다.

$$C = \begin{bmatrix} \mu_I & \sigma_I & \mu_Q & \sigma_Q \end{bmatrix} \quad (2)$$

손색상 결정 함수는 식 (3)과 같이 손색상 모델을 이용하여 칼라영상을 이진화 한다. 이 함수는 Mahalanobis 거리를 구하는 식을 이용하여 정의한다. 평균을 중심으로 3σ만큼의 범위에 속하는가를 판별하여 손색상을 결정한다. M은 손색상 모델의 I 와 Q 요소의 평균을 말하며 X는 입력 화소의 값을 나타내고 Σ^{-1}은 손색상 함수의 표준편차의 제곱으로 구성된 공분산 행렬의 역행렬이다. D(X)의 값이 3 보다 작은 경우에는 입력 화소가 손색상을 갖는 것으로 판정하고 그렇지 않은 경우에는 손색상을 갖지 않는 것으로 판정한다. 손색상 결정함수의 임계값이 3 인 것은 $\mu\pm3\sigma$의 거리를 Mahalanobis 거리를 이용하여 거리에 대한 정규화를 수행함으로써 얻는다.

$$D(\mathrm{X}) = \sqrt{[\mathrm{M}-\mathrm{X}]^{\mathrm{T}}\Sigma^{-1}[\mathrm{M}-\mathrm{X}]} \quad (3)$$

$$\mathrm{M} = \begin{bmatrix} \mu_I \\ \mu_Q \end{bmatrix} \quad X = \begin{bmatrix} f_I(x,y) \\ f_Q(x,y) \end{bmatrix} \quad \Sigma^{-1} = \begin{bmatrix} 1/\sigma_I^2 & 0 \\ 0 & 1/\sigma_Q^2 \end{bmatrix}$$

$$\begin{cases} D(X) \le 3 & X \in C \\ otherwise & X \notin C \end{cases}$$

2.2 손색상 모델의 학습

손색상 모델의 학습은 환경에 적응적인 시스템을 구축하기 위해 필요하다. 조명과 배경의 변화에 따른 사전 지식이 주어지지 않은 영상에 대하여 영역 분리를 정확하게 수행한다는 것은 어려운 문제이다. 또한 같은 의미를 갖는 제스처라 해도 사용자에 따라서 그 움직임의 특징이 조금씩 다를 수 있다. 이러한 입력 영상들은 결과적으로 고정된 인수에 의한 제스처의 인식 성능을 저하시키는 원인이 된다. 이러한 문제를

해결하기 위해 시스템은 변하는 조명 및 배경으로부터 제스처의 영역을 분할하기 위한 정보를 사전에 학습할 수 있는 기능을 갖추어야 한다[10].

사용자와의 상호작용에 의한 학습 방법은 시스템이 제시하는 손모양의 템플릿에 사용자가 자신의 손을 올려 놓으면 시스템이 연속한 영상을 획득하여 분석한 후 손영역을 검출하여 손색상 모델을 학습하는 방법이다. 학습 단계는 준비 단계, 진입 단계, 검증 단계, 완료 단계의 4 단계로 되어 있다. 첫번째, 준비 단계에는 시스템이 사용자에게 손모양의 템플릿을 제공하여 시스템을 초기화한다. 진입 단계에서는 사용자가 템플릿 위에 손을 진입시키는 단계이다. 그리고 검증 단계에서는 사용자가 템플릿 위에 손을 고정시키고 시스템이 템플릿을 분석하여 손을 확인하는 단계이다. 마지막으로 완료 단계는 템플릿으로부터 손색상 모델인 평균과 표준편차를 획득하는 단계이다. 그림 2는 각 단계별로 진행 과정을 보인다.

검증 단계에서 템플릿 위에 손이 있는가를 판별하기 위한 방법은 준비 단계에서 획득된 템플릿 내부의 배경과 현 시점의 템플릿 내부의 영역에 대한 유사성과 현 시점과 전 시점의 템플릿 내부의 영역에 대한 유사성을 측정하여 수행된다. 두 영역의 유사성은 일반적으로 많이 알려져 있는 유사성 측정 방법인 형판 정합(template matching) 기법[11]을 이용하여 계산한다.

(a) 준비 단계 (b) 진입 단계

(c) 검증 단계 (d) 완료 단계

그림 2 상호작용에 의한 손색상 모델의 학습

획득된 유사성은 적절한 임계값을 적용하여 학습 모델 추출 조건에 만족하는가를 결정한다. 학습 모델 추출 조건은 현 시점의 영역을 초기 시점의 영역과 비교하는 추출 조건 (1)과 전 시점의 영역과 비교하는 추출 조건 (2)가 있다. 학습 모듈은 추출 조건 (1)의 유사성이 낮고 추출 조건 (2)의 유사성이 높은 경우가 일정시간 동안 지속이 되면 검증을 완료하고 완료 단계를 수행한다. 완료 단계에서는 템플릿 내부에 있는 화소들을 대상으로 I 요소와 Q 요소의 평균과 표준편차를 구하여 손색상 모델을 작성한다.

2.3 손영역 레이블링

본 논문에서는 칼라영상을 이진화하여 배경으로부터 손색상을 갖는 화소를 분리한다. RGB 칼라공간을 사용하는 입력 영상을 YIQ 칼라공간으로 변환하여 I 요소와 Q 요소의 값을 계산하고 식 (2)와 식 (3)에서 정의한 손색상 모델과 손색상 결정함수를 사용하여 이진화를 수행한다. 이진화된 영상은 영역이 분리된 상태가 아니라 단지 각각의 화소들이 배경과 구분되어 있는 상태이다. 이런 물체 부분에 해당하는 각각의 화소들을 하나의 영역으로 묶어 각 영역에 레이블 값을 할당하는 과정을 레이블링이라 한다[12]. 레이블링 방법은 영상의 첫번째 라인부터 마지막 라인까지 스캔하면서 레이블을 할당하는 래스터 스캔(ratser scan)방법과 레이블이 할당된 초기 화소로부터 레이블을 전파하는 방법으로 나눌 수 있다[13]. 그러나 후자의 방법은 병렬 처리가 지원되지 않는 컴퓨터에서는 수행 시간이 매우 길어지게 되므로 주로 전자의 방법을 사용한다.

본 논문에서는 기존의 2-패스 레이블링 알고리즘보다 수행 속도가 빠르고 기억 장소의 낭비가 적은 경계 추적에 의한 1-패스 레이블링 알고리즘을 사용한다. 이 알고리즘은 레이블링 윈도우를 사용하여 영상을 스캔하면서 레이블을 할당한다. 레이블링 윈도우는 2×2 윈도우로써 이미 스캔한 다른 화소들의 레이블 값을 확인하여 적절한 레이블을 할당하는 역할을 수행한다. 레이블링 윈도우는 영상의 좌에서 우로 위에서 아래로 스캔을 한다. 레이블링 윈도우를 사용하여 레이블을 할당하기 위한 두 가지 조건이 있다. 첫 번째 조건은 레이블링 윈도우의 다른 화소들이 레이블을 갖고 있지 않는 경우이다. 이런 경우에는 새로운 레이블을 레이블링 윈도우의 기준 화소에 할당

한다. 두 번째 조건은 레이블링 윈도우의 다른 화소에 이미 레이블이 할당되어 있는 경우이다. 이런 경우에는 이미 할당되어 있는 레이블을 기준 화소에 할당한다.

그림 3은 경계 추적 레이블링 알고리즘의 적용 예를 보인다. S_0는 외부 경계 시작점을 S_1는 내부 경계 시작점을 나타낸다. S_0에서 레이블을 할당하는 첫 번째 조건을 만족하기 때문에 새로운 레이블을 할당하고 S_0를 시작점으로 하여 외부 경계 추적을 시작한다. 외부 경계 추적은 외부 경계 시작점과 같은 레이블을 할당한다. 추적 알고리즘이 다시 시작점으로 돌아오면 추적을 정상적으로 종료하고 추적 알고리즘을 적용하기 전의 방향으로 스캔을 다시 시작한다.

그러나 그림 3의 예와 같이 영역 안에 구멍이 존재하는 경우에, 외부 경계 추적은 외부 경계에 이미 시작점과 다른 레이블이 할당되어 있어서 추적 알고리즘이 시작점으로 돌아오지 못하고 다른 레이블과 충돌하게 된다.

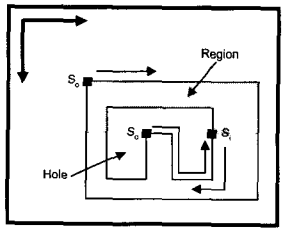
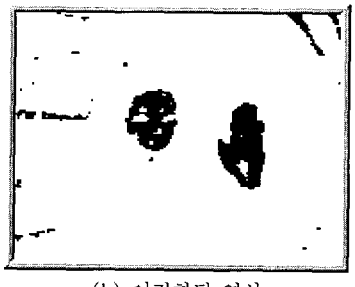


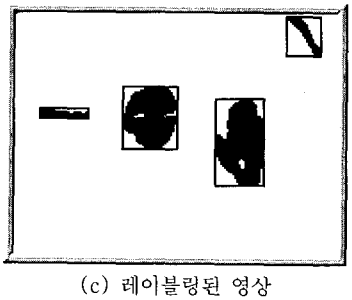
그림 3 경계 추적 레이블링 알고리즘의 적용 예

이 화소를 내부 경계의 시작점인 S_1로 설정하고 외부 경계 추적 방향과 반대 방향으로 내부 경계 추적을 시작하며 시작점 S_1와 같은 레이블을 할당한다. 내부 경계 추적이 다시 S_1로 돌아오면 추적을 종료하고, 내부 경계 추적을 하기 전 단계인 외부 경계 시작점에서 원래의 진행 방향으로 다시 스캔을 시작한다. 이러한 단계를 반복하여 수행함으로써 레이블링 작업을 수행한다. 그림 4는 이진화와 레이블링의 예를 보인다. 그림 4 (c)는 하나의 레이블을 사각형으로 표시하여 레이블링된 결과를 보인다.

(a) 입력 영상



(b) 이진화된 영상



(c) 레이블링된 영상

그림 4 레이블링의 예

2.4 손영역의 추적

전 시점으로부터의 손영역 정보를 이용하여 현 시점의 손영역을 추적하기 위해 초기 손영역을 추출하여야 한다[14]. 본 논문에서는 시작영역을 설정하고 사용자와의 상호작용을 통하여 시작영역의 범위 내에서 초기 손영역을 추출한다. 시작영역을 설정함으로써 시스템은 시작영역 내에서 찾아지는 영역이 두 가지의 초기 손영역 추출 조건을 만족하는가 확인하고 만족한다면 초기 손영역으로 인식한다. 초기 손영역 추출 조건은 2.2절에 기술한 학습 모델 추출 조건과 같다. 인식 준비 단계에서 획득된 시작영역의 영상과 현 시점의 시작영역의 영상의 유사성을 비교하고 현 시점과 전 시점의 시작영역내의 영상의 유사성을 비교하여, 전자의 유사성이 낮고 후자의 유사성이 높은

경우에 획득된 현 시점의 손영역을 초기 손영역으로 추출한다. 그림 5는 응용 시스템에 설정된 시작영역과 초기 손영역을 추출하는 예이다.

칼만필터는 이동 물체의 움직임에 대한 정보가 칼만필터의 상태 인수들에 누적되어 저장될 수 있기 때문에 이동 물체의 추적에 많이 이용된다[15, 16]. 본 논문에서는 칼만필터를 손색상 모델의 갱신과 영역 분리를 위한 탐색영역의 제한에 사용한다. 칼만필터는 시스템의 상태의 최적 예측인 선형 최소 오차(LMV: Linear Minimum Variance of error) 예측을 위한 순차적이면서 재귀적인 알고리즘을 제공한다. 먼저 상태 모델이 선형이라고 가정하고 식 (4)와 같이 정의한다[17].

(a) 준비 단계 (b) 진입 단계

(c) 검증 단계 (d) 완료 단계

그림 5 초기 영역의 추출 예

$$x(t) = \Phi(\Delta t)x(t-\Delta t) + w(t-\Delta t) \quad (4)$$

여기서 x(t)는 시점 t 에서의 시스템 상태를 나타내고 Φ(Δt)는 상태 전이 행렬을 나타낸다. 본 논문에서는 시스템의 상태를 16 차원의 백터로 표현하고 단위 시간 동안의 손영역의 위치 변화와 크기 변화, 그리고 손색상 모델의 변화를 표현한다. 즉, 식 (5)의 Δx 와 Δy 는 각각 손영역의 x 축과 y 축 상의 중점의 변위를 나타내고 μ_I, σ_I, μ_Q, σ_Q는 손색상 모델을 나타낸다. 여기서 $\overline{\Delta x}$, $\overline{\Delta y}$, $\overline{xs}$, $\overline{ys}$, $\overline{\mu_I}$, $\overline{\sigma_I}$, $\overline{\mu_Q}$, $\overline{\sigma_Q}$는 각각 Δx, Δy, xs, ys, μ_I, σ_I, μ_Q, σ_Q 의 시간 t 에 대한 변화율을 나타낸다. 손영역의 궤적은 등가속도 운동하고 물체의 크기와 손색상 모델은 선형적으로 변화한다고 가정할 때 상태 전이 행렬은 식 (5)와 같이 정의된다.

칼만필터 알고리즘은 측정값들의 집합에 기반하여 시스템의 상태를 예측한다. 그리고 시스템의 상태와 측정값들의 집합 사이에는 식 (6)과 같이 선형 관계가 있다고 가정한다.

$$y(t) = H(t)x(t) + v(t) \quad (6)$$

여기서 y(t)는 측정값들의 집합을 나타내고, H(t)는 관찰 행렬을 나타낸다. 그리고 v(t)는 측정 오류를 나타낸다. 각 시점에서의 손영역의 위치 변화와 크기, 그리고 손색상 모델을 측정하여 y(t)의 값들을 얻는다. 시스템 모델과 측정 모델이 정의되면, 동작 인수들의 LMV 예측을 얻기 위해 재귀적 칼만필터 알고리즘을 적용할 수 있다. 재귀적 칼만필터 알고리즘은 초기화, 상태 예측, 측정 갱신의 세 단계의 작업으로 구성된다[18].

초기화 단계에서는 초기 상태 예측 $\hat{x}(0)$, 실제 초기 상태인 $x(0)$ 와 $\hat{x}(0)$ 의 오차를 나타내는 초기 오류 공분산 행렬인 P(0), 예측 오류에 대한 상관 관계 행렬 $Q(t) = E(w(t)w(t)^T)$, 측정 오류에 대한 상관 관계 행렬 $R(t) = E(v(t)v(t)^T)$ 등을 결정한다. $\hat{x}(0)$ 의 값들을 계산하기 위해서는 초기 손영역의 정보를 사용한다. 입력 영상의 중심의 위치로부터 손영역의 중심 위치까지의 변위를 Δx(0), Δy(0)의 값으로 설정한다. 그리고 손영역의 최소인접사각형의 가로, 세로 크기를 이용하여 xs(0)와 ys(0)를 설정한다. P(0), Q(0), R() 등은 일반적인 통계적인 방법으로 결정되는데, 본 논문에서는 단위 행렬을 사용한다.

상태 예측 단계에서는 사전 LMV 예측을 결정하고 전 시점의 상태 예측과 오류 공분산에 기반하여 현 시점의 상태의 오류 공분산 행렬을 결정한다. 식 (7)은 이러한 과정을 수식으로 표현한 것이다.

$$x(t)=\begin{bmatrix}\Delta x(t)\\ \Delta y(t)\\ xs(t)\\ ys(t)\\ \mu_I(t)\\ \sigma_I(t)\\ \mu_Q(t)\\ \sigma_Q(t)\\ \overline{\Delta x}(t)\\ \overline{\Delta y}(t)\\ \overline{xs}(t)\\ \overline{ys}(t)\\ \overline{\mu_I}(t)\\ \overline{\sigma_I}(t)\\ \overline{\mu_Q}(t)\\ \overline{\sigma_Q}(t)\end{bmatrix} \quad \Phi(\Delta t)=\begin{bmatrix}1&0&0&0&0&0&0&0&\Delta t&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&\Delta t&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0&\Delta t&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&\Delta t&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&\Delta t&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0&0&0&\Delta t&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&\Delta t&0\\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&\Delta t\\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\end{bmatrix} \quad (5)$$

$$\begin{aligned}\hat{x}^-(t)&=\Phi(\Delta t)\hat{x}(t-\Delta t)\\ P^-(t)&=\Phi(\Delta t)P(t-\Delta t)\Phi^-(\Delta t)+Q(t-\Delta t)\end{aligned} \quad (7)$$

여기서 $\hat{x}^-(t)$는 측정값 y(0), y(1),...,y(t-Δt)를 기반으로 t 시점에서의 시스템 상태에 대한 사전 예측을 나타내고, $\hat{x}(t-\Delta t)$는 측정값 y(0), y(1),...,y(t-Δt)를 기반으로 t-Δt 시점에서의 시스템 상태에 대한 최적의 예측을 나타낸다. 본 논문에서는 $\hat{x}^-(t)$를 이용하여 손영역을 획득하기 위한 탐색영역을 제한하고 손색상 모델을 갱신한다. 측정 갱신 단계에서는 식 (8)과 같이 LMV 예측과 현재 상태에 대한 오류 공분산 행렬을 제공하기 위해 예측된 정보를 새로운 측정값들과 결합한다. 여기서 $K(t)(y(t)-H(t)\hat{x}^-(t))$는 $\hat{y}(t)$를 기반으로 $\hat{x}(t)-\hat{x}^-(t)$에 대한 최적의 LMV 예측을 제공한다. 즉 $\hat{x}(t)$의 예측 $\hat{x}^-(t)$로부터 발생하는 오류에 대한 최적의 보정을 표현한다.

$$\begin{aligned}K(t)&=P^-(t)H^T(t)(H(t)P^{-(t)}H^T(t)+R(t)^{-1})\\ P(t)&=(I-K(t)H(t))P^-(t)\\ \hat{x}(t)&=\hat{x}^-(t)+K(t)(y(t)-H(t)\hat{x}^-(t))\end{aligned} \quad (8)$$

2.5 손영역의 획득

칼만필터 알고리즘은 손영역의 예측된 중점의 변위와 크기를 제공한다. 이들 예측된 중점의 변위인 Δx(t)와 Δy(t)는 중점을 구하는데 사용되고, 예측된 크기 xs(t)와 ys(t)를 이용하여 탐색영역을 설정한다. 탐색영역의 중심은 예측된 중점으로 하고 길이와 너비는 예측된 크기 xs(t)와 ys(t) 의 1.5 배로 설정한다. 손영역 분리는 탐색영역에서만 수행되어 후보영역들을 추출한다. 전 시점에서 획득된 손영역 정보를 이용하여 전 시점의 손영역과 현 시점의 영역들 사이의 유사도를 계산함으로써 후보 영역들 중의 하나를 손영역으로 결정한다. 즉, 전 시점의 손영역을 추적하여 현 시점의 손영역을 결정한다[19].

유사도(Likelihood)는 유사성과 이동성의 가중치의 합으로써, 유사성은 전 시점의 손영역과 현 시점의 영역이 겹치는 정도를 나타내고 이동성은 현 시점의 영역의 이동 정도를 나타낸다. 유사성과 이동성에

대한 전제 조건으로는 손의 움직임이 너무 빠르지 않고 손의 형태가 빠르게 변하지 않는다는 조건, 그리고 손영역은 지속적으로 움직인다는 조건이 있다. 두 가지의 특징은 서로 상반되는 개념으로서 유사성이 높아지면 이동성이 낮아지고 이동성이 높아지면 유사성이 낮아지는 특성을 갖고 있지만 두 특징값에 가중치의 합을 함으로써 서로의 단점을 보완하는 역할을 수행할 수 있게 된다. 유사성과 이동성에 대한 정의는 식 (9)와 같다.

$$Simil(R_i(t), R_h(t-1)) = 2 \times \frac{Area(R_h(t-1) \cap R_i(t))}{Area(R_h(t-1)) + Area(R_i(t))}$$

$$Mobil(R_i(t)) = \frac{Area(R_i(t)) - Area(R_i(t) \cap (\bigcup_j R_j(t-1)))}{Area(R_i(t))} \quad (9)$$

유사성은 전 시점의 손영역 $R_h(t-1)$를 참조하여 계산한다. 후보영역 $R_i(t)$가 전 시점에서 획득된 손영역 $R_h(t-1)$와 완전히 겹쳐지면 1의 값을 갖고 $R_i(t)$가 $R_h(t-1)$과 전혀 겹쳐지는 부분이 없다면 0의 값을 갖는다. 반면에, 이동성은 전 시점의 모든 후보영역들을 참조하여 계산한다. $R_i(t)$가 전 시점의 위치로부터 완전히 움직였으면 1의 값을 갖고, 전혀 움직이지 않았으면 0의 값을 갖는다. 유사도는 유사성과 이동성에 적당한 α와 β를 곱하여 더함으로써 계산한다. 현 시점의 모든 영역들에 대하여 유사도를 계산하여 그 값들 중에서 가장 큰 값을 갖는 영역을 현 시점에서 획득된 손영역으로 결정한다. 유사도는 식 (10)과 같다.

$$Likelihood(R_i(t)) = \alpha \times Simil(R_i(t), R_h(t-1)) + \beta \times Mobil(R_i(t))$$
$$\text{where } \alpha, \beta \geq 1,\ \alpha > \beta \quad (10)$$

손영역 획득에 실패하는 경우는 사용자의 손이 카메라의 시야 밖으로 이동하는 경우와 손의 이동 속도가 빨라서 예측된 탐색영역을 벗어나는 경우가 있다. 첫 번째 경우는 손이 카메라의 시야를 벗어난 지점에서 일정 시간 동안 손이 재진입하기를 기다린다. 그러나 일정시간이 경과하여도 손이 재진입되지 않는다면 손영역 획득 모듈은 초기 상태로 돌아가서 새로운 초기 손영역을 추출하기 위해 사용자에게 시작영역을 제시한다.

두 번째의 경우에는 입력 영상 전체를 탐색영역으로 설정하여 손영역 추출을 시도한다. 영역 추적에 실패한 경우에는 손이 빠르게 움직이는 때에는 손 모양의 변화가 많지 않다는 전제조건을 사용하여 전 시점과 비교하여 영역의 형태가 매우 유사하고 매우 많이 이동한 영역을 손영역으로 획득한다. 형태의 유사성과 영역의 이동성을 구하기 위한 계산식은 식 (9)와 같다. 최종적으로 현 시점의 모든 영역들에서 형태 유사성과 영역 이동성의 결과값이 임계값 이상인 영역을 손영역으로 획득한다.

획득된 손영역으로부터 구분력이 좋은 특징값을 얻기 위해서는 먼저 손가락의 끝이 위를 향하도록 위치를 정규화 시켜야 한다. 손영역의 방향을 구하기 위해 기본축의 기울기를 계산한다. 기본축은 영역의 무게중심을 지나는 장축과 단축을 말한다. 기본축의 기울기 θ는 2 차 모멘트(moment)를 이용하여 모멘트의 고유벡터(eigenvector)를 계산함으로써 구한다. 고유벡터는 두 개의 기본축인 주 기본축과 부 기본축을 표현하고, 고유값(eigenvalue)인 K는 2 차 모멘트로 정의되어 있다. K는 식 (11)과 같으며, 기본축의 기울기는 식 (12)와 같이 정의된다.

$$K = \frac{(m_{20} + m_{02}) + \sqrt{(m_{20} - m_{02})^2 + 4m_{11}^2}}{2} \quad (11)$$

$$\theta = \tan^{-1}\left(\frac{K - m_{02}}{m_{11}}\right) \quad (12)$$

주기본축의 단일 방향을 결정하기 위해서 다음의 두 가지 제약 사항을 고려한다. 첫 번째 조건은 $m_{20} \geq m_{02}$이다. 만일 첫번째 조건이 만족되지 않으면 θ는 90 도씩 증가된다. 두 번째 조건은 $m_{03} < 0$이다. 만일 두 번째 조건이 만족되지 않으면 θ는 180 도씩 증가된다. 최종적으로 θ는 두 가지의 제약 사항을 고려한 값을 갖는다[20]. 식 (12)에 의해 계산된 기본축의 기울기는 손영역을 손가락의 끝이 위를 향하도록 회전시킨다.

3. 프레임 지식을 이용한 제스처 인식

3.1 프레임 지식의 표현

제스처를 인식하기 위해서는 인식 대상인 손 자체에 대한 지식과 손의 움직임에 대한 지식, 그리고 두 가지 형태의 지식을 적절하게 연결하여 시간의 흐름에 따라 동적으로 변화하는 제스처에 대한 특징을 표현하는 지식이 필요하다. 본 논문에서는 객체 프레임과 행동 프레임, 그리고 스키마를 사용하여 제스처에 대한 지식을 표현한다. 객체 프레임은 손 자체에 대한 지식으로서 인식하려는 물체의 특징들을 표현하고, 행동 프레임은 손의 움직임에 대한 지식으로서 객체 프레임이 전 시점에서 현 시점까지 어떻게 움직였는가를 표현한다. 하나의 의미를 갖는 제스처는 손에 대한 객체 정보와 행동 정보를 모두 포함하기 때문에 동적으로 변화하는 제스처를 모델링하기 위해 스키마를 사용한다. 스키마는 하나의 의미를 갖는 제스처에 대한 지식으로서 객체 프레임과 행동 프레임이 순차적으로 연결된 구조를 갖는다.

객체 프레임은 인식하고자 하는 대상 물체에 대한 지식을 표현한다. 즉, 손의 형태를 인식하기 위해 사용하는 지식을 의미한다. 객체 프레임의 구조는 물체의 특징을 표현하는 제약 슬롯(constraint slot)의 집합으로 정의하고, 객체 프레임의 인식은 획득된 손 영역이 특정한 객체 프레임에 속한 제약 슬롯들을 모두 만족하는 가를 평가함으로써 이루어 진다. 제약 슬롯은 처리기(procedure), 특징(feature), 결과값(value)으로 구성된다.

처리기는 제약 슬롯을 평가하기 위해 필요한 특징 추출을 담당하는 프로세스들로 정의된다. 처리기를 통해 계산된 결과는 특징에 정의되어 있는 조건(condition)과 비교하여 제약 슬롯의 만족도를 평가한다. 계산된 만족도는 결과값의 상태(status)에 저장된다. 조건은 상한 값과 하한 값을 이용하여 특징값의 범위를 정의하는 역할을 한다. 만약 동일한 입력 영상에서 같은 제약 슬롯이 반복적으로 활성화될 경우에 다시 처리기를 수행하여 그 결과와 조건을 반복적으로 비교하지 않고, 결과값에 저장되어 있는 만족도를 그대로 사용함으로써 제약 슬롯의 중복으로 인한 불필요한 계산 시간을 줄일 수 있다.

객체 모델을 구축할 때 계층성을 갖도록 하는 것은 문제 영역에 존재하는 객체를 효과적으로 구축하게 한다. SIGMA는 상위 프레임과 하위 프레임의 종속적인 관계로 객체 프레임을 표현함으로써 명시적으로 지식을 표현한다. 그러나 본 논문에서는 수평적인 제약 슬롯의 포함 관계로 객체 프레임의 계층 구조를 표현함으로써 비명시적으로 지식을 표현한다. 즉, 하나의 제약 슬롯은 여러 개의 객체 프레임에 포함될 수 있다. 제약 슬롯을 객체 프레임의 포함 관계로 볼 때, 제약 슬롯은 각각의 객체 프레임과 일대일의 관계를 갖는 것이 아니라 제약 슬롯 자체가 하나의 지식원으로서 독립적으로 존재한다는 것을 의미한다. 이렇게 제약 슬롯을 정의함으로써, 객체의 특성에 대한 제약 슬롯을 객체 프레임이 많이 포함하고 있으면 객체에 대한 지식은 점점 상세화 되고, 반면에 객체의 특성에 대한 제약 슬롯이 제거되면 객체에 대한 지식은 일반화됨을 알 수 있다. 이러한 계층 구조에서는 상위 프레임에서 계산된 특징들이 하위 프레임에서 필요한 제약 슬롯이라면 그 제약 슬롯은 다시 계산될 필요가 없다. 예를 들면, 그림 6에서 객체 1을 "손", 객체 2를 "주먹을 쥔 손", 객체 3을 "검지를 편 손"이라 하자. 논리적인 표현으로는, 객체 1의 두 개의 제약 슬롯이 색상 정보와 위치 정보를 나타낸다고 할 때 객체 2와 객체 3이 활성화되면 두개의 제약 슬롯은 객체 1로부터 계산된 색상 정보와 위치 정보를 상속 받아 그대로 사용할 수 있다. 물리적인 표현으로는, 객체 2가 객체 1의 제약 슬롯을 포함함으로써 객체 1로부터 색상 정보와 위치 정보를 상속 받는 것을 알 수 있다. 이 시스템에서는 논리적인 표현으로 객체 프레임을 구현하지 않고 물리적인 표현을 이용하여 객체 프레임을 구현한다.

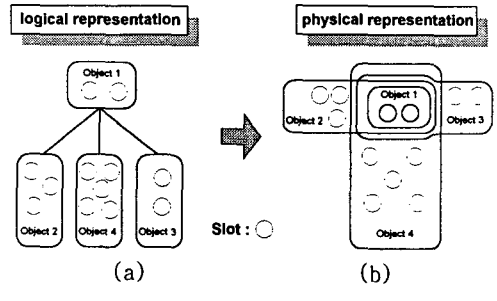


그림 6 객체 프레임의 계층 구조

동적인 제스처를 인식하기 위해서는 객체 프레임이 전 시점에서 현 시점까지 어떻게 변화하였는가를 인식해야 한다. 이러한 객체 프레임의 변화를 인식하

기 위해 행동 프레임을 사용한다. 즉, 행동 프레임은 손의 움직임에 대한 지식을 표현한다. 행동에 대한 사전적 의미는 객체의 움직임에 대한 구체적인 설명을 말한다. 그러나 본 논문에서 사용하는 행동은 어떤 구체적인 설명을 갖는 것이 아니라 임의의 객체가 단순히 어떠한 형태로 움직였다는 것에 대한 지식으로 정의한다. 예를 들면, 사용자가 손을 흔드는 동작을 하고 있을 때 손 객체가 좌우로 움직이고 있다는 것을 행동으로 정의한다.

행동 프레임은 전 시점에서부터 현 시점까지의 객체 프레임들간의 관계를 표현하는 지식으로서 이동(move), 정지(pause), 변환(change)의 3가지 기본 틀을 갖고 있다. 전 시점에서 인식된 객체 프레임과 현 시점에서 추출된 인식 영역이 얼마나 유사한가를 계산하여 현 시점의 행동 프레임에 3가지 기본 틀 중 하나를 할당하게 된다. 유사성이 높고 인식 영역의 위치가 변한 경우에는 이동을 할당하고 유사성이 높고 인식 영역의 위치가 변하지 않은 경우에는 정지를 할당하게 된다. 그러나 유사성이 낮은 경우에는 변환을 할당한다.

행동 프레임은 여러 개의 평가 슬롯으로 구성되어 있다. 이동 프레임인 경우에는 객체 프레임이 시작하는 위치(start position)와 끝나는 위치(stop position)를 나타내는 평가 슬롯으로 구성되어 있고 정지 프레임의 평가 슬롯은 움직임이 멈춘 위치(pause position)와 정지한 동안의 시간(duration)으로 구성되어 있다. 그리고 변환 프레임의 평가 슬롯은 시작 위치(start position), 끝난 위치(stop position), 변환을 시작한 객체 프레임 이름(start frame), 변환된 객체 프레임 이름(stop frame), 변환되는 동안의 시간(duration)으로 구성된다. 행동 프레임은 시간에 매우 종속적이므로 일정한 간격의 시간을 하나의 시간 단위로 정의하여 사용한다. 하나의 시간 단위 동안에 위치가 변한 경우는 객체 프레임이 움직인 것으로 인식하고 하나의 시간 단위 동안 위치가 변하지 않은 경우는 객체 프레임이 움직이지 않은 것으로 인식한다. 적당한 시간 단위를 정의함으로써 행동 프레임의 이동과 정지를 구별한다. 두개의 연속된 영상 프레임에서 시간 간격은 매우 중요한 의미를 갖는다. 시간 간격이 너무 작다면, 실제 연속적으로 움직이고 있는 물체를 정지해 있는 것으로 잘못 판단하기 쉽다. 그리고 시간 간격이 너무 길면, 실제 정지해 있는 물체를 움직이고 있다고 잘못 판단할 수 있다는 문제점이 있다. 즉, 물체의 움직임은 절대적인 시간의 개념이 아니라 상대적인 시간의 개념을 갖는다. 현재 시스템에서 사용하는 시간 간격은 영상을 획득하는 시간을 포함하여 하나의 입력 영상을 분석하기 위해 필요로 하는 전체 시간을 하나의 시간 간격으로 정의한다.

하나의 의미를 갖는 제스처를 모델링하기 위해 객체 프레임과 행동 프레임을 통합하는 스키마를 정의한다. 스키마는 하나의 의미를 갖는 물체의 움직임들에 대한 지식, 즉 하나의 제스처를 표현하며 객체 프레임과 행동 프레임이 순차적으로 연결된 구조를 갖는다. 스키마는 미리 구축되어 있는 지식으로서 인식 대상이 되는 제스처를 나타낸다. 스키마의 기본 구조는 시작 동작으로 정의된 주먹 프레임부터 스키마를 시작한다. 그리고 객체 프레임과 객체 프레임 사이에는 행동 프레임을 정의하여서 객체 프레임의 움직임을 표현한다. 최종적으로 스키마에 정의된 마지막 객체 프레임을 인식하게 되면 현재 활성화된 스키마를 인식한 것으로 하고 새로운 제스처를 인식하기 위한 상태로 전환된다.

그림 7은 스키마의 기본 구조를 보여준다. 이 예에서 객체 프레임은 주먹을 쥔 형태인 주먹 프레임과 손바닥을 편 형태인 손바닥 프레임을 사용하고 행동 프레임은 변환, 이동, 정지를 사용하고 있다. 이 스키마는 주먹을 쥐고 있다가 손가락을 펴서 다른 곳으로 움직이고 그 위치에 정지하고 있다가 다시 주먹을 쥔 상태로 돌아오는 제스처를 표현한 것이다. 하나의 제스처를 표현하는 각각의 스키마는 계층적인 트리 구조로 통합되어 구축된다.

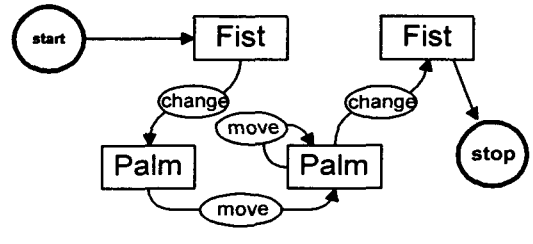


그림 7 스키마의 예

3.2 특징 추출

프레임 지식은 적절하게 배치된 특징이 필요하다. 객체 프레임의 제약 슬롯은 다른 객체와 구분할 수

있는 특징이 필요하고 행동 프레임의 평가 슬롯은 객체 프레임의 형태와 위치 변화를 측정할 수 있는 특징이 필요하다. 제스처의 실시간 인식을 위해서는 빠르게 계산되어야 하기 때문에 특징 추출은 간단하고 효율적으로 이루어져야 한다. 객체 프레임에 사용되는 특징[21, 22]은 장단축 비율, 밀집성, 원형성, 손영역의 상단과 하단의 크기 비율, 손영역과 최소인접사각형(Least Enclosing Rectangle, LER)의 비율, 손가락의 개수 등이 있다.

장단축 비율은 손영역의 높이와 너비의 비율을 의미한다. 식 (13)과 같이 손영역의 최소인접사각형의 높이와 너비를 계산하여 장단축 비율을 계산한다. 이 특징은 주먹과 다른 손모양을 구별하는데 유용하다. 손영역이 주먹을 표현할 때는 특징값이 1에 가깝고 다른 손모양인 경우에는 1보다 작다.

$$\frac{Length\ of\ horizontal\ axis(LER(R_h(t)))}{Length\ of\ vertical\ axis(LER(R_h(t)))} \quad (13)$$

밀집성은 손영역의 밀집 정도를 표현한다. 식 (14)와 같이 손영역 둘레의 길이의 제곱과 최소인접사각형의 면적의 비율로 계산한다. 이 특징은 주먹을 쥔 손 또는 손바닥을 편 손과 손가락을 편 손을 구별하는데 유용하다. 손영역이 손가락을 편 손인 경우의 특징값이 주먹을 쥔 손 또는 손바닥을 편 손인 경우의 특징값보다 큰 값을 갖는다.

$$\frac{Perimeter^2(R_h(t))}{Area(LER(R_h(t)))} \quad (14)$$

원형성은 밀집성과 유사한 특징으로써 손영역이 얼마나 원형과 유사한가를 측정하는 특징이다. 이 특징은 밀집성보다 손영역의 외곽선의 굴곡에 덜 민감하다. 특징값은 영역의 무게중심으로부터 외곽선까지의 평균 거리와 외곽선까지의 거리 차의 합에 반비례한다. 영역이 원에 가까울수록 작은 값을 갖는다.

손영역 상단과 하단의 크기 비율은 식 (15)와 같다. 이 특징은 손영역이 손가락을 편 상태인지 아닌지를 구분하는데 좋은 특징이다. 만약 특징값이 1에 가깝다면 손가락을 편 상태가 아니라 주먹을 쥔 상태이거나 손바닥을 편 상태를 나타낸다. 손가락을 편 상태에서는 손영역 상단의 크기보다 손영역 하단의 크기가 커지므로 특징값이 1보다 큰 값을 갖게 된다.

$$\frac{Area(Upper(R_h(t)))}{Area(Lower(R_h(t)))} \quad (15)$$

손영역 좌상단과 우상단의 크기 비율은 식 (16)과 같다. 이 특징은 두 개의 손가락을 편 상태, 즉 검지와 중지를 편 상태와 검지와 약지를 편 상태를 구분하기 위한 특징이다. "검지와 중지를 편 손"의 특징값은 두개의 손가락이 오른쪽에만 나타나므로 매우 큰 값을 갖게 되고, "검지와 약지를 편 손"의 특징값은 왼쪽과 오른쪽의 영역에 각각 한 개씩의 손가락이 나타나므로 1에 가까운 값을 갖게 된다. 손영역의 하단은 손가락의 상태를 나타내지 못하므로 손가락의 상태를 잘 나타내는 손영역의 상단만을 특징값 추출 대상 영역으로 한다.

$$\frac{Area(RightUpper(R_h(t)))}{Area(LeftUpper(R_h(t)))} \quad (16)$$

손영역의 크기와 최소인접사각형의 크기의 비율은 식 (17)과 같이 계산한다. 특징값은 1보다 작거나 같은 값을 갖으며, 이 특징은 손영역에 대한 최소인접사각형과 실제의 손영역이 얼마나 유사한가를 판별함으로써 식 (14)와 같이 주먹을 쥔 손 또는 손바닥을 편 손과 손가락을 편 손을 구분하는 특징이다. 만약 손영역과 최소인접사각형이 완전히 일치하는 경우에는 1의 값을 갖으며 손영역이 주먹을 쥔 상태이거나 손바닥을 편 상태임을 나타낸다.

$$\frac{Area(R_h(t))}{Area(LER(R_h(t)))} \quad (17)$$

손가락의 개수는 히스토그램의 분석에 의해 추출된다. 히스토그램은 손영역에서 수평 방향으로 진행하면서 검은 화소에서 흰 화소로 변하는 부분인 런(run)의 개수를 누적하여 구한다. 일반적으로 손영역은 다섯 개의 손가락을 갖고 있으므로 런의 길이가 손영역의 너비의 1/7보다 길고 2/5 보다 짧은 경우에 히스토그램에 누적을 한다. 이 특징은 히스토그램에 누적된 런의 개수의 평균을 계산하여 손가락의 개수

를 추출한다. 그림 8은 두개의 다른 형태의 손영역의 손가락 개수를 추출하기 위한 히스토그램을 보인다.

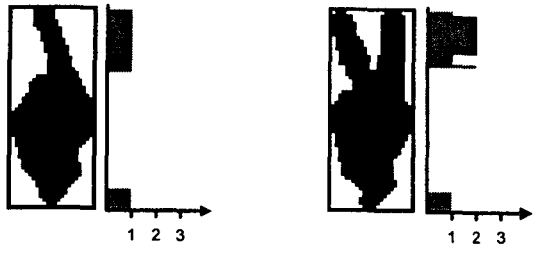


그림 8 손가락 개수의 히스토그램

손 객체의 움직임을 세가지 유형(이동, 변환, 정지)중에 하나로 결정하기 위해서, 행동 프레임의 평가 슬롯은 전 시점의 손영역과 현 시점의 손영역의 형태와 위치 유사성을 계산해야 한다. 위치 유사성의 계산은 두 손영역의 중심의 거리를 이용하여 구한다. 형태 유사성은 식 (18)과 같이 두 손영역의 크기를 정규화하여 겹쳐지는 부분을 계산하여 구한다.

$$2\times\frac{Area(R_h(t-1)\cap R_h(t))}{Area(R_h(t-1))+Area(R_h(t))} \qquad (18)$$

3.3 지식의 제어

일반적으로 인간은 먼저 물체의 모양이나 색상을 예측하고 그 예측을 확인하는 과정을 통하여 물체를 인식한다. 본 논문에서는 이러한 특징을 이용하여 손을 찾고자 할 때 먼저 적당한 손 모델을 활성화하고 영상으로부터 추출된 영역에 활성화된 모델이 부합하는 가를 결정하여 손을 인식한다. 스키마는 손 객체와 손 객체의 움직임에 대한 정보를 갖고 있기 때문에 현재의 상황에 적합한 손 모델을 활성화시키기 위해 매우 중요한 역할을 한다. 제스처 인식 과정은 활성화된 스키마의 객체 프레임과 행동 프레임이 예시화 되는 과정으로 볼 수 있다. 하나의 스키마가 활성화되면 인식 모듈은 활성화된 스키마에 있는 프레임들을 순차적으로 예시화 시킨다. 예시화는 현재 인식 대상이 되는 영상이 갖고 있는 제약 사항들을 활성화된 프레임이 만족시키는 가를 판별하는 과정이다. 만약 선택된 프레임의 예시화에 실패하면, 인식 모듈은 다른 프레임을 활성화시키거나, 현재의 중간 인식 결과를 만족하는 다른 스키마를 활성화시킨다.

다른 객체 프레임을 활성화시킬 때 두 가지 경우를 고려해야 한다. 첫번째는 행동 프레임이 변환 프레임인 경우이다. 이 경우에는 현재 활성화된 스키마에 있는 다음 단계의 객체 프레임을 활성화한다. 그러나 즉시 다른 객체 프레임을 활성화시킬 수 없다. 왜냐하면, 하나의 객체가 다른 객체로 변화할 때 변화하는 동안에는 인식 대상이 되는 영역의 제약 사항이 계속적으로 조금씩 변화하기 때문에 일정한 형태의 객체로 모델을 할 수 없기 때문이다. 제약 슬롯을 갖지 않는 가상의 객체 프레임인 여분 객체 프레임(temporary object frame)을 정의하여 다른 객체로 변환하는 동안의 중간 형태를 처리할 수 있게 하였다. 두 번째 경우는 활성화된 스키마가 현재의 상황에 적절하지 않는 경우이다. 이런 경우에는 스키마의 트리 구조를 참조하여 다른 스키마를 활성화시킨다.

그림 9는 제스처 인식 시스템의 전체적인 제어 구조를 보인다. 제어의 기본 골격은 네 개의 테스트로 구성되어 있다. 각 테스트에 따라서 시스템은 다른 형태의 동작을 취한다. 첫번째 테스트는 추출된 영역이 시작 동작인지 아닌지를 판별한다. 모든 제스처는 "주먹을 편 손"으로 정의된 시작 동작으로 시작된다. 추출된 영역이 시작 동작으로 판별되면 시스템은 적절한 스키마를 활성화하여 제스처 인식을 시작한다. 그렇지 않다면 시스템은 시작 동작을 인식하기 위해 반복적으로 계속 첫번째 테스트를 수행한다. 일단 스키마가 활성화된 후, 제어는 연속적으로 획득된 영상에 대하여 적절한 객체 프레임(Object Frame, OF)과 행동 프레임(Behavior Frame, BF)을 활성화하고 활성화된 프레임들을 평가하는 과정을 반복한다. 반복되는 제어는 먼저 활성화된 스키마로부터 다음 행동 프레임과 객체 프레임을 선택한다. 그리고 다음 시점의 영상을 읽어 손영역을 획득한 후에 현 시점의 손영역의 형태나 위치 같은 특징값들을 계산하고 현 시점과 전 시점의 손영역을 비교하여 유사성을 계산한다. 특징값들이 계산이 된 후에는 현재 활성화된 객체 프레임(Active Object Frame, AOF)과 행동 프레임(Active Behavior Frame, ABF)이 추출된 특징들을 만족하는 가를 확인한다. 만족 한다면 현재 인식된 객체 프레임이 마침 동작인가를 판별하고 마침 동작이 아닌 경우에는 활성화되어 있는 객체 프레임과 행동 프레임을 추가함으로써 중간 인식 결과를 갱신하고 다음 시

점의 영상을 인식한다.

마침 동작인 경우에는 인식 모듈의 수행을 마친다. 그러나 활성화된 객체 프레임 또는 행동 프레임이 추출된 특징값들을 만족하지 못하면 스키마 뿐만 아니라 현재의 객체 프레임과 행동 프레임을 현재의 중간 인식 결과를 이용하여 계층적 트리 구조에서 적절한 스키마와 객체 프레임, 그리고 행동 프레임을 선택한다. 현 시점의 영역과 전 시점의 영역 사이의 유사성 측정은 행동 프레임을 결정하기 위해 사용된다. 시스템은 이런 제어 구조를 반복함으로써 제스처에 대한 인식을 수행한다. 결과적으로 프레임 지식에 대한 제어는 스키마에 정의되어 있는 객체 프레임과 행동 프레임이 어떻게 연결되어 있는 가에 의하여 정의된다.

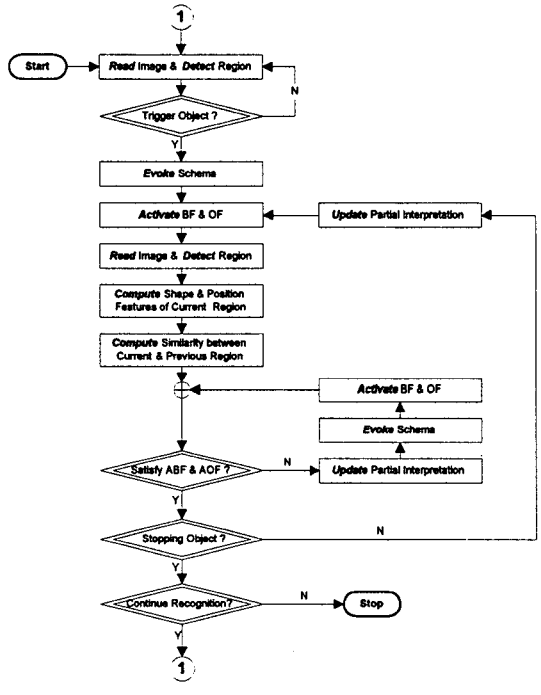


그림 9 프레임 지식의 제어 흐름도

4. 실험 및 결과

4.1 실험환경

실험은 펜티엄 133MHz 칩이 장착된 PC에서 수행하였다. 운영체제로는 MS Windows 95를 사용하였고 컴파일러로는 Visual C++ 4.0을 사용하였다. 영상 입력 장비로는 3.5mm 와이드 렌즈가 장착된 CCD 카메라와 Matrox사에서 제작한 Meteor 보드를 사용하였다. 영상 처리를 위해 사용한 영상의 크기는 160x120 크기의 칼라영상을 사용하였다. Meteor 보드에서는 기본적으로 640x480 크기의 칼라영상이 입력되지만 영상 처리 속도와 제스처 인식의 정확성을 고려하여 영상의 크기를 160x120 크기로 하였다. 실험 환경은 CCD 카메라가 모니터 위에 장착되어 있고 사용자는 모니터의 정면에서 제스처를 취한다. 카메라에 입력되는 배경 영상은 책상과 책이 진열되어 있는 책장, 그리고 사무실의 벽 등 일반적인 사무실의 배경과 유사하게 설정하였다.

삼목 게임, 일명 Tic-Tac-Toe 게임은 사각형의 판 위에 한 사람은 O을, 그리고 다른 한 사람은 X를 사각형 안에 놓아 먼저 일렬로 세 개를 만든 사람이 이

기는 게임이다. 이 게임은 다른 말판 게임들과 비교하여 볼때 비교적 간단한 게임 규칙을 갖고 있어 사용자가 게임 규칙을 쉽게 이해할 수 있을 뿐만 아니라 프로그램의 구현도 간단하여 본 논문에서 제안하는 제스처 기반 인터페이스의 효용성을 보이기에 매우 적합한 특징을 갖는다. 사용되는 제스처는 삼목 게임뿐만 아니라 다른 응용 프로그램들에게도 쉽게 적용하기 위해 삼목 게임 프로그램에만 종속적이지 않은 공통적인 제스처를 정의하였다. 삼목 게임 프로그램은 게임을 시작하기 위한 게임시작 버튼, 실행을 취소하기 위한 실행취소 버튼, 게임을 종료하고 프로그램을 빠져나가기 위한 게임종료 버튼, 3x3 의 9 개의 게임 보드, 게임의 승패 점수를 보여주는 스코어 보드 등으로 구성되어 있다. 사용자가 게임 보드중에서 하나를 선택하면 O 를 표시하고 바로 컴퓨터가 적절한 위치에 X 를 한다. 항상 사용자가 먼저 게임을 시작한다. 그리고 하나의 게임이 끝나면 게임 보드위의 O 과 X 가 모두 지워지고 게임시작 버튼을 선택했을 때와 마찬가지로 새로운 게임이 시작된다.

삼목 게임 프로그램에서 사용되는 제스처는 게임을 새로 시작하는 것을 나타내는 ***시작 제스처***, 게임의 종료를 나타내는 ***종료 제스처***, 실행을 취소하는 ***취소 제스처***, 말을 움직이는 ***이동 제스처***, 그리고 말을 움직여서 말판 위에 놓는 ***선택 제스처***가 있다. 시작 제스처는 주먹을 쥔 상태에서 검지와 중지를 폈다가 다시 주먹을 쥐는 동작, 종료 제스처는 손바닥을 펴서 좌우로 흔드는 동작, 취소 제스처는 주먹을 쥔 상태에서 검지와 약지를 폈다가 다시 주먹을 쥐는 동작, 이동 제스처는 검지 손가락을 펴서 원하는 위치로 움직이다가 주먹을 쥐는 동작, 선택 제스처는 이동 제스처와 같이 검지 손가락을 펴서 원하는 위치로 움직이다가 손바닥을 폈다 주먹을 쥐는 동작으로 정의한다. 정의된 제스처는 그림 10 과 같다.

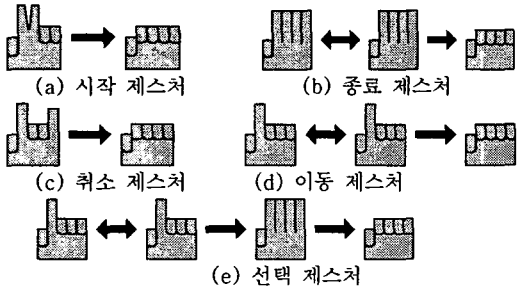


그림 10 정의된 제스처

다섯 가지의 제스처에 대한 스키마 트리 구조는 그림 11과 같다. 사용된 객체 프레임은 ***주먹을 쥔 손***을 표현하는 Fist 프레임, ***손바닥을 편 손***을 표현하는 Palm 프레임, ***검지를 편 손***을 표현하는 Fin1 프레임, ***검지와 중지를 편 손***을 표현하는 Fin2 프레임, ***검지와 약지를 편 손***을 표현하는 Fin3 프레임이 있다.

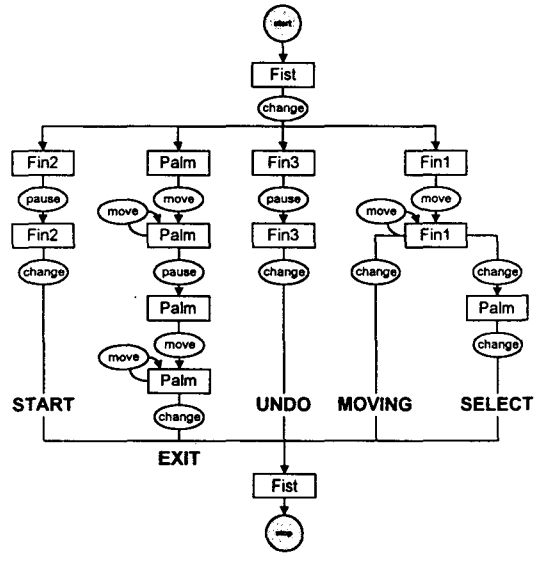


그림 11 스키마 트리 구조

4.2 실험 결과

손영역 획득 모듈이 입력 영상으로부터 손영역을 획득하는 결과를 그림 12 에서 보인다. 그림 12 의 손영역 획득 실험은 1 분동안 수행되었고 초당 3-4 프레임을 처리하므로 1 분 동안에 획득되어 처리되어진 입력 영상은 약 200 프레임이다. 그림 12 는 200 프레임 중에서 15 프레임을 적당한 시간 간격으로 획득하여 순서대로 보인다. 왼쪽 영상은 입력 영상을 나타내고 오른쪽 영상은 획득된 손영역을 보인다. 오른쪽 영상에서 손영역을 가장 근접하게 둘러싸고 있는 사각형은 손영역의 최소인접사각형을 나타낸 것이고 조금 큰 형태의 사각형은 칼만필터에 의해 예측된 탐색영역을 나타낸다.

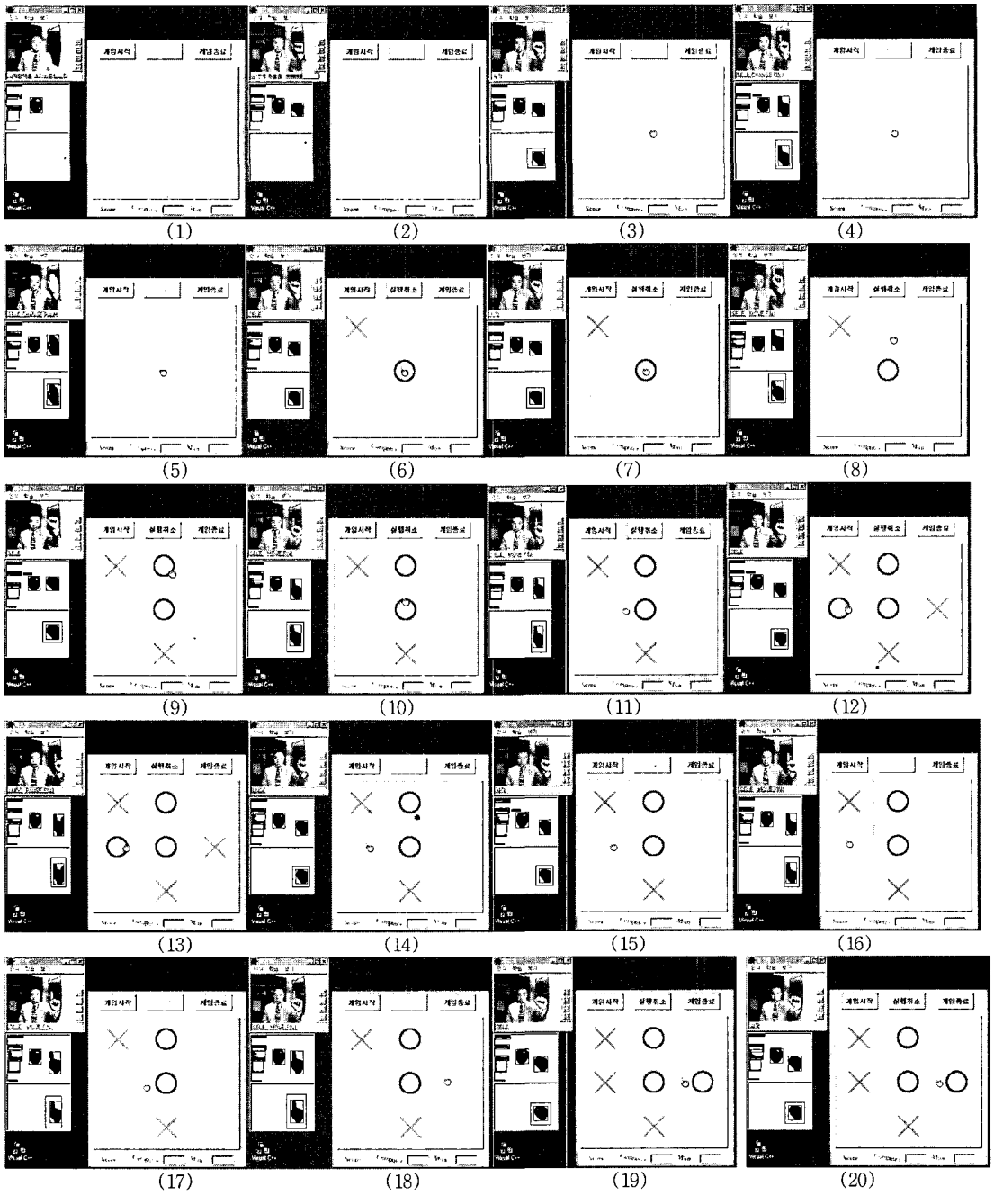
그림 12 의 (a)부터 (e)까지는 초기 손영역 추출

단계를 보이며 전 시점으로부터 손영역을 추적하여 획득하는 단계는 그림 12 (e)부터 이다. 그림 12의 (e)부터 (f)까지는 손을 위에서 그리고 다시 아래로 위로 움직이고 있다. 이때 칼만필터의 예측에 의한 탐색영역이 손이 움직이던 방향으로 치우쳐 있다. 또한 그림 12 (g)와 같이 움직임이 크지 않은 경우에는 탐색영역의 중심과 손영역의 중심이 유사하다. 이러한 결과는 칼만필터에 의한 예측이 잘 수행되고 있음을 보인다. 만약 그림 12 (j)와 같이 다른 영역과 손영역이 겹쳐진 경우에 다른 영역과 손영역이 하나의 영역으로 합쳐진다. 그러나 다시 손영역이 분리되면 그림 12의 (k) 에서 보는 것과 같이 원래의 손영역을 계속 추적한다. 이 결과에 의하여 손영역 획득 모듈은 손영역이 다른 영역과 겹치는 경우에도 안정적으로 다음 시점의 손영역을 계속 추적할 수 있다는 것을 알 수 있다.

그림 12 손영역 획득 결과

그림 13은 본 논문에서 구현한 제스처 기반 인터페이스를 이용하여 삼목 게임 프로그램을 수행하는 실행 결과를 보인다. 왼쪽 상단에 위치한 메인 윈도우는 카메라로부터 입력되는 입력 영상과 제스처 인식 결과를 상태바에 출력한다. 그 메인 윈도우의 하단에 위치한 윈도우는 손색상 모델과 손색상 결정함수, 그리고 경계 추적 레이블링 알고리즘을 사용하여 추출된 손색상을 갖는 영역의 레이블링된 결과를 이진 영상의 형태로 보인다. 레이블링 윈도우 하단의 윈도우는 칼만필터에 의해 예측된 탐색영역과 손영역

획득 모듈에 의하여 획득된 손영역을 이진 영상의 형태로 출력한다. 그리고 오른쪽에 위치한 가장 큰 원도우는 삼목 게임 프로그램이다. 실행 결과는 초기 손영역 추출 단계에서부터 시작하여 본 논문에서 정의한 5가지의 제스처를 이용하여 게임을 진행한다. 초기 손영역을 추출하면 커서가 가운데에 위치하게 되고 커서의 모양이 손모양으로 변한다.



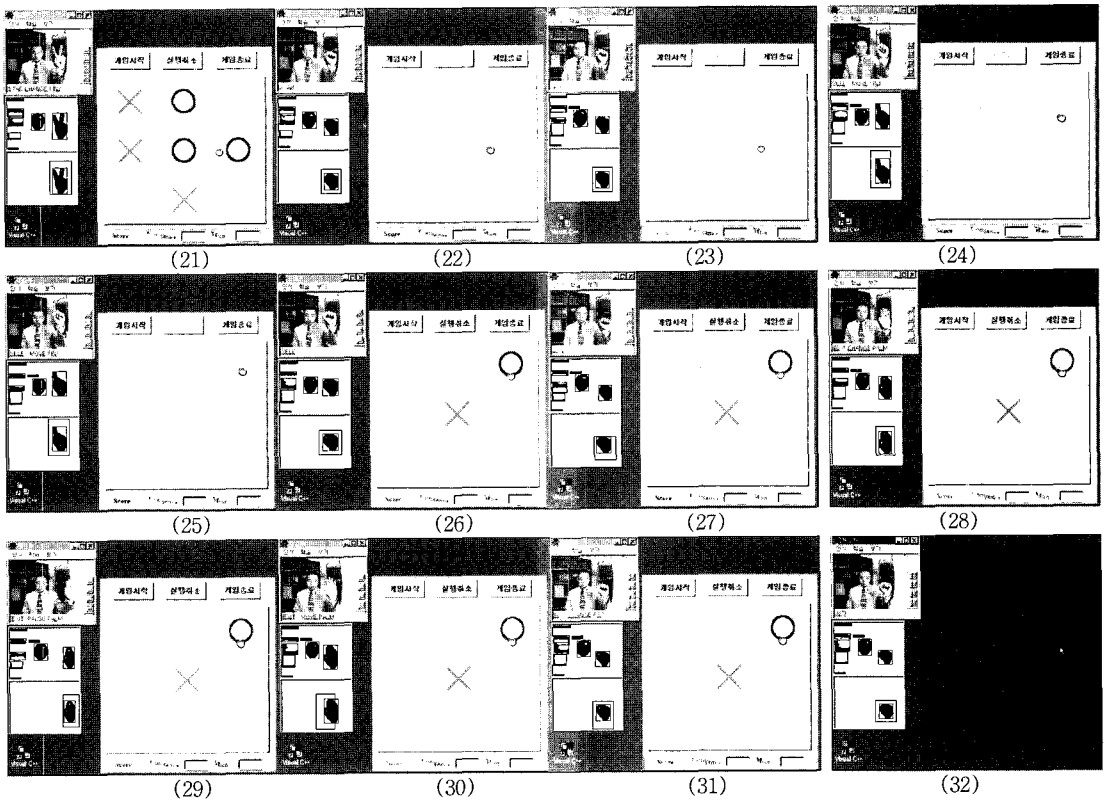


그림 13 제스처 기반 인터페이스의 실행 결과

그림 13 (1)에서 그림 13 (3)까지는 초기 손영역을 추출하는 과정을 보인다. 그리고 그림 13 (4)에서 그림 13 (12)까지는 선택 제스처를 이용하여 말판을 선택하는 과정을 보인다. 사용자는 선택 제스처에 정의되어 있는 ***검지를 편 손***을 이용하여 커서의 위치를 움직인다. 커서의 위치는 검지 손가락 끝을 인식하여 계산한다. 그림 13 (13)과 그림 13 (14)는 취소 제스처를 인식하여 게임 판의 왼쪽 중간에 있는 O을 지우는 과정을 보인다. 삼목 게임 프로그램의 취소 명령은 단 한 번만 유효하며, 취소 명령이 유효한 동안만 삼목 게임 프로그램의 상단에 있는 실행취소 버튼을 선택할 수 있다. 그림 13 (15)를 보면 실행취소 버튼이 선택할 수 없도록 비활성화된 것을 알 수 있다. 그림 13 (22)는 시작 제스처를 인식하여 게임을 새로 시작하는 과정을 보인다. 게임을 새로 시작하면 게임 판 위에 그려진 O과 X를 모두 지운다. 그리고 처음 상태로 돌아가서 새로운 게임을 시작한다. 최종적으로 사용자가 종료 제스처를 취하면 그림 13 (32)와 같이 삼목 게임 프로그램이 종료된다.

5. 결론 및 향후 연구 방향

본 논문에서는 컴퓨터 시각을 이용한 제스처 기반 인터페이스를 구현하였으며, 구현된 제스처 기반 인터페이스를 적용한 삼목 게임 프로그램을 구현하였다. 영상 특징에 의한 컴퓨터 시각 인식 방법을 사용함으로써 사용자에게 부가적인 장치를 착용하게 하지 않고 자연스러운 상태에서 제스처 기반 인터페이스를 제공할 수 있었다. 칼라영상의 영역 분리를 쉽게 하기 위해 정의한 손색상 모델과 손색상 결정함수는 손색상에 해당하는 영역을 배경 영상으로부터 잘 분리하였고, 손색상 모델의 갱신과 탐색영역의 제한을 위해 사용한 칼만필터도 좋은 결과를 보여주었다. 손영

역 획득은 초기 손영역을 사용자와의 상호작용에 의해 획득한다. 그리고 다음 시점부터는 칼만필터에 의해서 탐색영역을 제한하고, 탐색영역 내에서만 손색상 모델을 이용한 영역 분리를 수행하여 추출된 후보 손영역들 중에서 전 시점의 손영역과 유사도가 가장 높은 후보 손영역을 현 시점의 손영역으로 획득하였다. 결론적으로 손색상 모델과 칼만필터를 이용한 영역 추출과 추적에 의한 손영역 획득 방법은 복잡한 배경 영상으로부터 안정적으로 손영역을 획득할 수 있는 가능성을 제시하였다.

획득된 손영역은 활성화된 스키마로부터 생성된 객체 프레임과 정합을 수행하였고 전 시점의 손영역과 비교하여 행동 프레임과 정합을 수행하였다. 그리고 스키마는 정합된 객체 프레임과 행동 프레임을 이용하여 동적인 제스처를 인식하였다. 프레임 지식을 객체 프레임과 행동 프레임, 그리고 스키마로 정의함으로써 정적인 제스처뿐만 아니라 동적인 제스처도 인식을 할 수 있도록 하였다. 프레임 지식을 이용한 제스처 기반 인터페이스를 적용한 응용 시스템으로는 삼목 게임 프로그램을 구현하였다. 모두 다섯 가지의 제스처를 정의하여 사용하였는데 다른 종류의 게임 프로그램에도 적용 가능할 것이다. 그리고 이미 정의되어 있는 제스처의 수정이나 정의되어 있지 않은 다른 제스처를 추가하는 것은 객체 프레임과 스키마를 수정하거나 갱신함으로써 간단하게 이루어 질 수 있다.

향후 연구 과제로서 먼저 수행되어야 할 과제는 객관적인 성능 평가에 대한 것이다. 인식 시스템의 제스처 인식 결과에 대한 평가는 주어진 환경이나 사용자의 제스처 인식 시스템에 대한 숙련도에 따라서 매우 많은 차이가 있을 것으로 생각된다. 그러므로 제스처 인식 방법의 비교를 위해서는 최종적인 인식율에 대한 비교보다는 제스처 인식 시스템의 전체적인 구조나 각 모듈마다의 성능 비교, 제스처가 정의된 지식 표현 방법, 지식의 제어 방법, 인식 시스템의 확장성, 다양한 환경에 대한 적응성 등을 성능 평가를 위한 비교 대상으로 사용할 수 있을 것이다.

제안하는 제스처 인식 방법은 손영역의 크기 비율이나 손가락의 개수 등과 같은 간단하고 직관적인 특징들을 사용하였다. 그러나 제스처 인식 시스템이 더욱 다양하고 복잡한 형태의 손 모양을 인식하기 위해서는 직관적인 특징들 외에도 복잡하고 정교한 형태를 구별할 수 있는 특징들을 개발해야 한다. 또한 카메라의 정면을 향하고 있는 손 모양만을 인식하는 2차원적인 특징들에 대한 연구뿐만 아니라 손 모양의 공간적인 특성을 표현할 수 있는 3차원적인 특징들에 대한 연구가 필요하다. 그리고 여러 특징들 중에서 적절한 특징들을 자동으로 선택하여 새로운 제스처를 인식할 수 있는 제스처 학습 모듈에 대한 연구도 수행되어야 할 것이다.

제스처 기반 인터페이스는 가상 현실 시스템, 화상 회의 시스템, 의학 시스템, 광고 시스템, 공장 자동화 시스템 등 매우 다양한 응용 분야에 적용할 수 있다[23, 24]. 의학 시스템에는 의사가 수술을 할 때 간단한 제스처를 이용하여 수술에 필요한 기계 장비들을 제어하는데 사용할 수 있다. 광고 시스템은 가상 박물관을 예로 들 수 있다. 가상 박물관에서 관람객의 위치에 따라서 화면에 보여지는 소장품들이 다른 모습을 보여줄 수 있다. 이렇게 함으로써 사용자에게 더욱 생동감 있는 관람 방법을 제공할 수 있을 것이다. 공장에서는 작업자가 기계를 제어하기 위해서는 장갑을 낀 손이나 기름이 묻은 손으로 키보드나 터치 스크린을 눌러야 한다. 이런 불편함을 제스처 기반 인터페이스를 사용함으로써 제거할 수 있다. 이렇듯 제스처 기반 인터페이스는 많은 응용 분야에서 사용될 수 있을 것이다. 향후 연구 방향의 가장 큰 목표는 여러 응용 분야에 본 논문에서 제안한 제스처 기반 인터페이스를 적용하는 것이다.

참 고 문 헌

[1] Francis K.H. Quek, "Toward a Vision-Based Hand Gesture Interface," Conference on Virtual Reality Software and Technology, pp.23-26, 1994.

[2] Thomas S. Huang, Vladimir I. Pavlovic, "Hand Gesture Modeling, Analysis, and Synthesis," International Workshop on Automatic Face- and Gesture-Recognition, pp.73-79, 1995.

[3] Alan Wexelbl at, "Natural Gesture in Virtual Environments," Conference on Virtual Reality Software and Technology, pp.5-16, 1994.

[4] Ching-Huei Wang, Sargur N. Srihari, "A Framework for Object Recognition in a Visually Complex Environment and its

Application to Locating Address Blocks on Mail Pieces," International Journal of Computer Vision, Vol.2, pp.125-151, 1988.

[5] John K. Tsotsos, John Mylopoulos, H. Dominic Covvey, Steven W. Zucker, "A Framework for Visual Motion Understanding," IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol.2, No.6, pp.563-573, 1980.

[6] Takashi Matsuyama, Vincent Shang-Shoug Hwang, SIGMA A Knowledge-Based Aerial Image Understanding System, Plunum Press, 1990.

[7] Ying Dai, Yasuaki Nakano, "Extraction of Facial Images from Complex Background using Color Information and SGLD Matrices," International Workshop on Automatic Face- and Gesture-Recognition, pp.238-242, 1995.

[8] Yu-Ich Ohta, Takep Kanade, Toshiyuki Sakai, "Color Information for Region Segmentation," Computer Vision, Graphics, and Image Processing, Vol.13, pp.222-241, 1980.

[9] Christopher c. Yang, Jeffrey J. Rodriguez, "Efficient Luminance and Saturation Processing Techinques for Bypassing Color Coordinate Transformations," IEEE Systems, Man and Cybernetics, pp.667-672, 1995.

[10] Haiyuan Wu, Qian Chen, Masahiko Yachida, "An Application of Fuzzy Theory : Face Detection," International Workshop on Automatic Face- and Gesture-Recognition, pp.314-319, 1995.

[11] Dana H. Ballard, Christopher M. Brown, Computer Vision, Prentice Hall, 1982.

[12] Ronald Lumia, Linda Shapiro, Oscar Zuniga, "A New Connected Components Algorithm for Virtual Memory Computers," Computer Vision, Graphics, and Image Processing, Vol.22, pp.287-300, 1983.

[13] Hironobu Takahashi, Fumiaki Tomita, "Fast Region Labeling with Boundary Tracking," IEEE International Conference on Image Processing, pp.369-373, 1989.

[14] François G. Meyer, Patrick Bouthemy, "Region-Based Tracking using Affine Motion Models in Long Image Sequences," Computer Vision, Graphics, and Image Processing : Image Understanding, Vol.60, No.2, pp.119-140, 1994.

[15] Nuria Oliver, Alex Pentland, "LAFTER : Lips and Face Real Time Tracker," M.I.T Media Laboratory Perceptual Computing Section TR-396, 1996.

[16] A. Azarbayejani, T. Starner, B. Horowitz, A. Pentland, "Visually Controlled Graphics," IEEE Transactions of Pattern Analysis and Machine Intelligence, Vol.15, No.6, pp.602-605, 1993.

[17] Mohinder S. Grewal, Angus P. Andrews, Kalman Filtering Theory and Practice, Prentice Hall, 1993.

[18] G. Minkler, J. Minkler, Theory and Application of Kalman Filtering, Magellan, 1994.

[19] Guido Tascini, Primo Zingaretti, "Image Seguence Recognition," SPIE 2308 Visual Communications and Image Processing, pp.838-847, 1994.

[20] C. H. Chien, J. K. Aggarwal, "A Normalized Quadtree Representation," Computer Vision, Graphics, and Image Processing, No.26, pp.331-346, 1984.

[21] Ioannis Pitas, Digital Image Processing Algorithms, Prentice Hall, 1993.

[22] Harley R. Myler, Arthur R. Weeks, The Pocket Handbook of Image Processing Algorithm in C, Prentice Hall, 1995.

[23] Christoph Maggioni, "GestureComputer - New Ways of Operating a Computer," International Workshop on Automatic Face- and Gesture-Recognition, pp.166-171, 1995.

[24] Richard Watson, "A Survey of Gesture Recognition Techniques," Trinity College Dublin TCD-CS-93-11, 1993.