

논문 2010-47CI-5-1

# 가이드라인을 이용한 동적 손동작 인식

## (Dynamic Hand Gesture Recognition using Guide Lines)

김 건 우\*, 이 원 주\*\*, 전 창 호\*\*\*

(Kun Woo Kim, Won Joo Lee, and Chang Ho Jeon)

### 요 약

일반적으로 동적 손동작 인식을 위해서는 전처리, 손 추적, 손 모양 검출의 단계가 필요하다. 본 논문에서는 전처리와 손 모양 검출 방법을 개선함으로써 성능을 향상시킨 동적 손동작 인식 방법을 제안한다. 전처리 단계에서는 동적테이블을 이용하여 노이즈제거 성능을 높이고, YCbCr 컬러공간을 이용한 기존의 피부색 검출 방식에서 피부색의 범위를 조절할 수 있도록 하여 피부색 검출 성능을 높인다. 특히 손 모양 검출 단계에서는 가이드라인을 이용하여 동적 손동작 인식의 요소인 시작이미지(Start Image)와 정지 이미지(Stop Image)를 검출하여 동적 손동작을 인식하기 때문에 학습예제를 사용한 손동작 인식 방법에 비해 인식 속도가 빠르다는 이점이 있다. 가이드라인이란 웹캠을 통해 입력되는 손의 모양과 비교하여 검출하기 위해 화면에 출력하는 손 모양의 라인이다. 가이드라인을 이용한 동적 손동작 인식 방법의 성능을 평가하기 위해 웹캠을 사용하여 복잡한 배경과 단순한 배경으로 구분된 9가지 동영상을 대상으로 실험하였다. 그 결과 CPU 점유율이 낮고, 메모리 사용량도 적기 때문에 시스템 부하가 높은 환경에 효과적임을 알 수 있었다.

### Abstract

Generally, dynamic hand gesture recognition is formed through preprocessing step, hand tracking step and hand shape detection step. In this paper, we present advanced dynamic hand gesture recognizing method that improves performance in preprocessing step and hand shape detection step. In preprocessing step, we remove noise fast by using dynamic table and detect skin color exactly on complex background for controlling skin color range in skin color detection method using YCbCr color space. Especially, we increase recognizing speed in hand shape detection step through detecting Start Image and Stop Image, that are elements of dynamic hand gesture recognizing, using Guideline. Guideline is edge of input hand image and hand shape for comparing. We perform various experiments with nine web-cam video clips that are separated to complex background and simple background for dynamic hand gesture recognition method in the paper. The result of experiment shows similar recognition ratio but high recognition speed, low cpu usage, low memory usage than recognition method using learning exercise.

**Keywords :** 손추적(hand tracking), 동적 손동작 인식(dynamic hand gesture recognition), 가이드라인(guide line)

## I. 서 론

컴퓨터의 급속한 발달로 인해 인간과 컴퓨터간의 의

사전달 수단이 키보드와 마우스 중심에서 음성, 센서, 또는 영상을 이용한 새로운 인터페이스들로 확대 되고 있다. 영상을 이용하는 방법은 컴퓨터에 연결된 카메라에서 인간의 동작, 얼굴 표정, 손동작 등을 인식하고, 다양한 의미를 부여하여 처리하는 방법이다. 특히 손동작은 표현력이 뛰어나기 때문에 널리 이용된다.

손동작 인식은 글러브 데이터(glove data)를 사용하는 방법과, 영상을 사용하는 방법으로 구분할 수 있다. 글러브 데이터를 사용하는 손동작 인식 방법은 손의 각 부위에 광센서 또는 굴절계측기를 장착한 장갑의 출력 값을

\* 학생회원, \*\*\* 평생회원, 한양대학교 컴퓨터공학부  
(School of Computer Science & Engineering  
Department, Hanyang University)

\*\* 정회원, 인하공업전문대학 컴퓨터정보공학부 컴퓨터  
정보과  
(Department of Computer Science, Inha Technical  
College)

접수일자: 2010년3월5일, 수정완료일: 2010년8월31일

이용하는 기기 기반의 측정 방식으로 손동작에 대한 정확한 위치를 얻을 수 있고, 3차원 공간의 다양한 손동작을 인식할 수 있다는 장점이 있다. 하지만 고가의 센서와 사람과 센서간의 연결선이 필요하기 때문에 사용이 불편하고, 손동작의 행동반경이 제한된다는 단점이 있다. 영상을 이용하는 손동작 인식 방법은 별도의 기계적인 장치를 직접 착용하지 않고, 카메라를 이용하기 때문에 인터페이스가 단순하다<sup>[1]</sup>. 하지만 손 영역을 추출하는 과정이 복잡하고, 손의 움직임 분석하거나 자세를 인식하는 과정이 어렵고 복잡하다. 이러한 단점을 극복하기 위해 손에 특정한 색표지(marker)를 붙이거나 표지가 있는 장갑을 사용하여 환경을 제한하기도 한다<sup>[2]</sup>.

실생활 환경에서 손동작 인식을 위해서는 다음과 같은 문제점을 해결해야 한다. 첫째, 컴퓨터에 연결된 카메라의 입력 영상에서 배경과 손 영역을 정확하게 분리할 수 있어야 한다. 실생활 환경의 영상은 다양한 조명과 색상, 패턴을 가지고 있기 때문에 배경과 손 영역을 정확하게 분리하는 것은 어렵다. 둘째, 손 영역에서 추가적인 환경 요소(노이즈, 조도, 배경색상)에 영향을 받지 않고 손동작을 정확하게 인식할 수 있어야 한다. 손 영역을 분리하더라도 동영상은 여러 프레임의 집합이기 때문에 각 프레임마다 환경의 변화가 생겨 잘못된 손동작 인식이 발생할 수 있다. 셋째, 학습 예제를 사용하지 않고 자유로운 손동작을 인식할 수 있어야 한다. 기존의 학습 예제를 사용하는 방법은 인식률이 높은 반면 학습을 위해 많은 시간이 필요하며, 구현하는 방법도 어렵다. 또한, 인식해야 할 동작을 추가하기 위해서는 1000~2000장의 예제 이미지가 필요하기 때문에 동작이 제한된다<sup>[5]</sup>.

본 논문에서는 이러한 문제점을 개선한 동적 손동작 인식(dynamic hand gesture recognition) 방법을 제안한다. 이 방법은 YCbCr 색상공간을 사용하여 RGB, 정규화된 RGB, HSV 색상공간보다 정확하게 피부색을 추출하여 카메라 입력 영상의 배경과 손 영역을 정확하게 분할한다. 또한 무게중심과 거리변환 행렬을 사용하여 정확한 손 추적이 가능하다. 특히, 학습 예제를 사용하지 않고, 가이드라인을 이용하기 때문에 연산량이 적어 검출 속도가 빠르다. 또한, CPU 점유율이 낮고, 메모리 사용량도 적다는 장점도 가진다.

## II. 관련 연구

동적 손동작 인식을 위해 많이 사용되는 영상 기반의

손 추적 방법과 손 모양 검출 방법은 다음과 같다.

### 1. 손 추적 방법

손 모양을 추출하고 추출된 손 영역의 궤적을 추적하는 방법은 차동분석 방법<sup>[3]</sup>, Optical Flow 방법<sup>[4]</sup>, Mean Shift 방법<sup>[6]</sup>, CAM Shift 방법<sup>[7]</sup> 등이 있다. 특히, Optical Flow 방법은 움직이는 영역 추출을 위해 입력 영상의 각 프레임에 대한 밝기 패턴의 가시적인 운동 변화를 검출한다. 이 방법은 차동분석 방법에 비해 속도가 빠르고 추적 정확도가 높지만 명도의 변화가 없다는 환경적 제약이 따른다. CAM Shift 방법은 Mean Shift 알고리즘을 다양한 환경에서 사용하기 위해 개선한 색상기반의 추적 알고리즘이다. 이 방법은 추적 대상의 폭과 높이를 계산한 후에, 중심 좌표와 크기에 비례하여 추적 범위를 변경하고, 임계값보다 추적 범위의 크기가 커지지 않을 때 까지 반복 수행하여 물체를 추적한다.

### 2. 손 모양 검출 방법

손 모양 검출 방법에는 예제를 사용하는 방법과 예제를 사용하지 않고 손 모양의 벡터 값과 좌표의 분포를 확인하여 인식하는 방법이 있다. 예제를 사용하는 방법에는 Template Matching<sup>[4]</sup>, 인공신경망<sup>[3]</sup>, 은닉마르코프 모델(HMM)<sup>[8]</sup> 등이 있으며, 예제를 사용하지 않는 방법에는 손의 특징점과 외곽선을 응용하는 방법이 있다.

Template Matching 방법은 입력 영상의 한 프레임과 예제 영상과 비교하여 유사한 패턴을 찾는 방법이다. Template Matching 알고리즘은 크게 곱셈연산을 사용하는 CC(Cross Correlation) 방식과 두 영상의 차를 이용하는 SAD (Sum of Absolute Difference) 방식이 있다. CC방식은 정합을 정확하게 할 수 있다는 장점이 있지만 곱셈연산으로 인해 연산량이 증가하여 처리 속도가 느려지는 단점이 있다. SAD방식은 CC방식에 비해 연산량이 적고 빠르지만 정확도가 떨어진다<sup>[4]</sup>.

인공신경망은 측정된 변수들 사이의 관계가 알려지지 않거나 입력력 값들이 비선형적으로 변하는 문제를 해결하는데 많이 사용된다. 인공신경망을 이용한 영상 인식은 다양한 예제 영상을 학습시키는 학습 단계와 학습된 영상을 분류하는 재생 단계로 구성되어 있다. 학습 단계에선 패턴의 종류에 따라 신경망의 연결가중치를 적당한 값으로 지정하고, 재생 단계에선 학습된 가중치들과 입력벡터들과의 거리를 계산하여 가장 가까운 클래스로 분류한다.

### III. 가이드라인을 이용한 동적 손동작 인식

기존의 손동작 인식 방법은 비전카메라 기반으로서 전처리 단계가 없기 때문에 그대로 웹캠 환경에 적용하면 인식률이 떨어진다. 웹캠을 이용하면서 인식률을 높이고자면 노이즈제거 및 조도 보정과 같은 전처리 단계가 필요한데 그러면 연산 량이 증가하여 처리속도가 느려지는 문제점이 발생한다.

이러한 문제점을 개선하기 위해 전처리 단계에서 노이즈 제거시 동적테이블을 사용하여 연산량을 줄이고, 손동작 검출 단계에서는 가이드라인을 이용하여 학습예제를 사용하는 인식 방법에 비해 인식 속도를 높인다. 가이드라인이란 웹캠을 통해 입력되는 영상에서 손 모양을 정확히 검출하기 위해 사용하는 손 모양의 윤곽선이다. 이 방법은 예제 이미지를 사용하지 않기 때문에 Template Matching이나 인공신경망, HMM에 비해 연산속도가 빠르다. 그리고 인식 전 예제 학습을 위한 과정이 필요 없으며, 동작을 추가하기 쉽다. 가이드라인을 이용한 동적 손동작 인식 방법은 입력 영상의 전처리 모듈, 손 추적 모듈, 손 모양 검출 모듈로 구성되며 그림 1과 같다.

그림 1의 초기화 단계에서는 캠을 초기화 하고, 변수들을 초기화 한다. 또한, 입력 영상의 이진화, 노이즈 제거, 가이드라인 생성, 손가락 끝점 검출, 색상공간(정규화된 RGB, YCbCr)변환하는 전처리 모듈을 수행한다.

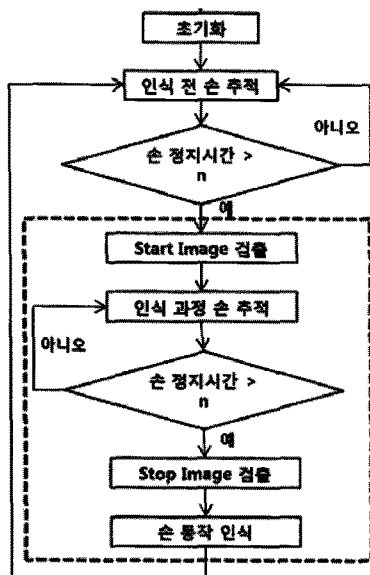


그림 1. 가이드라인을 이용한 동적 손동작 인식  
Fig. 1. Dynamic Hand Gesture Recognition using Guide Line.

인식 전 손 추적은 손 추적 모듈을 사용하여 손의 움직임 추적한다. 영상의 한 영역에서 손이 일정시간(n)이상 정지해 있으면 동적 손동작 인식 과정을 시작하게 된다. 점선 부영역은 손 모양 검출 모듈을 이용하여 start image를 검출하고, 손 추적 모듈을 이용하여 인식 과정의 손 추적을 시작한다. 인식 과정의 손 추적은 인식 전 손 추적과는 다르게 움직이는 궤도를 저장하여 손이 움직이는 방향을 알 수 있다. 마지막으로 stop image를 검출하고 start image의 결과 값과 추적된 손의 방향 값, stop image의 결과 값을 분석하여 최종적으로 손동작을 인식한다.

#### 1. 전처리 모듈

영상의 이진화, 가이드라인 생성 및 손 영역 검출, 노이즈 제거, 손가락 끝점 검출, 색상공간 변환의 기능을 수행하는 전처리 모듈의 알고리즘은 그림 2와 같다.

그림 2에서 oT는 끝점 검출 카운트 변수이고, C는 손가락의 수이다. Binarize(m\_pImage) 함수는 입력 영상을 이진화하고, 손 모양의 가이드라인을 생성한다. 가이드라인을 이용하여 가이드라인 내의 피부색 비율로 초기 손 영역을 검출한다. 피부색 검출은 성능을 고려

```

Preprocessing() {
    m_pBImage=Binarize(m_pImage); //입력영상 이진화
    CreateLine(m_pBImage); //가이드라인 생성
    while(){
        init(); //초기화
        skin=피부색 검출; //피부색/이외의색=1/0
        if(피부색 검출 완료){
            손 영역 좌표 저장;
            break; }
    }
    CreateEndLine(); //끝점 검출라인 작성
    C = 5; //손가락 개수 5개
    EndPointCount(); //끝점 찾기
    EndNo[손가락번호]=Pointangle(끝점 좌표);
} //손가락 위치 적용
EndPointCount(){
    oT=0; // 초기 손가락 개수 초기화
    for(r=cn;oT==C;r++){ // 끝점 검출을 위한 원 작성
        if(원의 라인에 끝점 검출) {
            끝점 포인트 좌표 저장;
            oT++;
        }
    }
}
}
    
```

그림 2. 전처리 알고리즘  
Fig. 2. Pre-processing algorithm.

하여 YCbCr 칼라 공간을 이용한다. YCbCr 칼라 공간을 이용한 피부색 검출식은 식 1과 같다<sup>[9]</sup>.

$$\begin{aligned}
 c_x &= 109.38, c_y = 152.02, \theta = 2.53 (\in \text{radian}) \\
 ec_x &= 1.60, ec_y = 2.41, a = 25.39, b = 14.03 \\
 \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} Cb - c_x \\ Cr - c_y \end{bmatrix} \\
 \frac{(x - ec_x)^2}{a^2} + \frac{(y - ec_y)^2}{b^2} &= 1
 \end{aligned} \tag{1}$$

식 1은 색의 범위가 백인 피부색에 최적화된 식이기 때문에 동양인 피부색 검출에는 적합하지 않다. 따라서 본 논문에서는 실험을 통하여 식 1을 식 2와같이 수정한다<sup>[10]</sup>.

$$48 < \frac{(x - ec_x)^2}{a^2} + \frac{(y - ec_y)^2}{b^2} < 56 \tag{2}$$

초기 손 영역은 그림 3과 같이 가이드라인의 최대 크기(150\*163)만큼 영역을 설정하고, A, B, C 영역으로 나누어 각 영역의 피부색 비율을 계산한다. A영역의 크기는 110\*80, B영역의 크기는 110\*83, C영역의 크기는 36\*83 이다. 모든 영역에서 피부색 변수 값이 정해놓은 임계값 이상이면 손 영역으로 정한다. 손 영역이 검출되었다 하더라도 전체 프레임에서 노이즈로 인한 오류가 발생할 수 있다.

본 논문에서는 검출된 손 영역을 제외한 모든 픽셀을 노이즈라 정하고 동적테이블을 구성하여 기존의 주변 픽셀을 비교하는 노이즈 제거기법을 사용한다<sup>[10-11]</sup>. 노이즈 제거기법은 중심 픽셀로부터 상하좌우의 인접 픽셀을 비교하여 반대의 값을 가지면 노이즈로 판단하여 픽셀의 값을 수정한다. 하지만 이러한 방법은 불필요한 픽셀의 중복 연산으로 인해 연산량이 증가하고 처리속도가 저하된다. 따라서 본 논문에서는 불필요한 픽셀의 중복 연산을 피하기 위해 식 3과 같은 재귀관

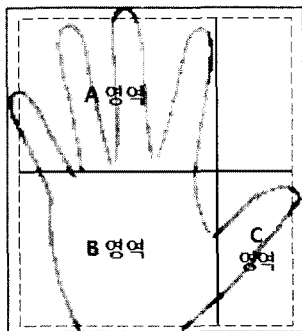


그림 3. 손 영역 검출  
Fig. 3. Hand Sphere Detection.

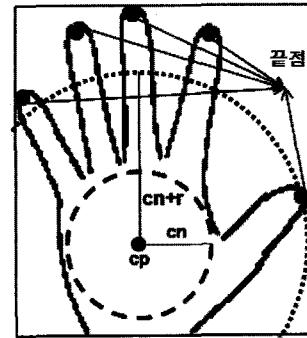


그림 4. 끝점 검출  
Fig. 4. End point Detection.

계식을 가진 동적테이블을 이용한 노이즈 제거기법을 사용한다<sup>[11]</sup>.

$$T[i,j] = \begin{cases} c[i,j] & \text{if } i=0 \text{ and } j=0 \\ c[i,j] - T[0,j-1] & \text{if } i=0 \\ c[i,j] - T[i-1,0] & \text{if } j=0 \\ c[i,j] - T[i-1,j] - T[i,j-1] & \\ \quad + T[i-1,j-1] & \text{if } i \neq 0 \text{ and } j \neq 0 \end{cases} \tag{3}$$

식 3에서 T[i, j]는 동적테이블이고, c[i, j]는 현재 픽셀의 상태이다. 이 기법은 기존의 노이즈 제거 방법에 비해 연산량이 적고 속도가 빠르다는 장점이 있다.

노이즈를 제거한 영상에서 정확한 손 영역 추적과 손 모양 검출에 사용할 손가락의 끝점과 각도를 검출한다. 손가락 끝점의 개수는 5이기 때문에 손가락 수(C)의 초기 값은 5로 설정한다.

그림 4에서 중심점 cp로부터 반지름 cn인 원을 그린다. T의 초기값은 0이며, T = 0T 이면 끝점을 찾지 못한 것으로 원의 반지름을 r만큼 늘리고 다시 끝점 검출을 시작한다. 만약 T ≠ 0T이면 끝점의 위치값을 끝점 위치 저장 변수에 저장한다. 그리고 C-T의 값을 구하여 현재 검출된 끝점의 개수를 확인한다. C-T = 0 이면 모든 손가락의 끝점이 검출된 것으로 인정하여 중심점과 끝점을 연결 하고, 그렇지 않으면 다시 끝점을 찾기 시작한다. 그림 4는 중심점으로부터 원의 반지름을 키워나가며 끝점을 찾는 과정이다.

### 2. 손 추적 모듈

전처리 모듈에서 검출된 손 영역과 중심점을 이용하여 손 영역을 정확히 추적하는 손 추적 알고리즘은 그림 5와 같다.

그림 5에서는 이동한 손 영역의 무게중심을 검출하기 위해 현재프레임(cf)과 이동프레임(df)을 합하여 추적프레임(tf)을 생성한다. 그리고 추적프레임(tf)의 무게

```

for(손 영역의 width크기){
  for(손 영역의 height크기){
    f(x,y)=of(x,y)-cf(x,y);
    종료; }
} //이동프레임 검출
for(손 영역의 width크기){
  for(손 영역의 height크기){
    if(현재픽셀==피부색){
      skincolor++;
      종료; }
    } else 종료; }
} //손 영역 확인
if(skincolor < 피부색 비율){
  이동프레임 재검출;
  종료; }
else {
  현재 손 영역의 크기 계산; //거리변환 함수 사용
  for(현재 손 영역의 width크기){
    for(현재 손 영역의 height크기){
      tf(x,y)=cf(x,y)+df(x,y);
      종료; }
    } }
} //이동된 손 검출
추적프레임(tf)에 대한 무게중심 검출(gc);
현재 무게중심(gc)와 이전프레임 무게중심(ogc) 비교;
손 추적;
}
    
```

그림 5. 손 추적 알고리즘  
Fig. 5. Hand tracking algorithm

중심(gc)를 구하여 이전프레임의 무게중심(ogc)과 좌표 값을 비교하여 손의 이동방향을 구한다. 이 알고리즘을 이용하여 손 추적을 하면 프레임 전체가 아닌 손 영역만을 비교하기 때문에 기존의 차등분석방법, 스캔칼라 모델방법에 비해 연산량이 감소하고, 처리속도는 증가한다.

3. 손 모양 검출 모듈

손 모양 검출 모듈은 추적된 손 영역으로부터 손 모양을 판별하는 모듈이다. 손 영역의 중심점과 손가락 끝점, 그리고 거리비율을 이용하기 때문에 학습 예제 이미지를 만드는 과정이 필요 없고, 처리속도 또한 빠르다. 현재 끝점의 개수 확인, 무게중심과 끝점 연결, 무게중심과 끝점의 각도를 계산하는 방법은 전처리 모듈을 이용한다.

4. 동적 손동작 인식

동적 손동작 인식을 하기 위해서는 전처리 완료 영상과 손 추적 모듈, 손 모양 검출 모듈이 필요하다.

인식 과정의 손 추적은 인식 전의 손 추적과 다르게 방향을 알기 위해 중심점의 이동 방향을 계속 확인함으로써 Start 이미지로부터 Stop 이미지의 이동 방향을 알 수 있다. 그림 6과 같이 Stop 이미지가 정해지면 Start 이미지의 EndPoint 중심점  $G(x_s, y_s)$ 와 Stop 이미

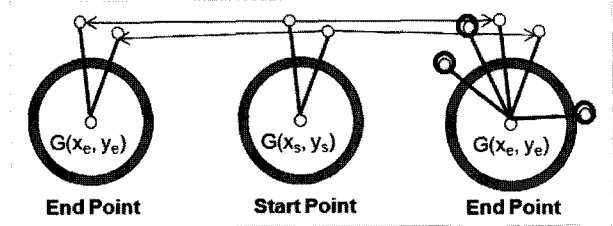


그림 6. 동적 손동작 인식  
Fig. 6. Dynamic Hand Gesture Recognition.

지의 EndPoint 중심점  $G(x_e, y_e)$ 를 이용하여 손의 이동 방향을 검출한다. 만약,  $x_e - x_s$ 의 결과가 양수일 경우, 손은 오른쪽으로 이동하고, 음수일 경우는 왼쪽으로 이동한다. 그리고  $y_e - y_s$ 의 결과가 양수일 경우는 손은 위쪽으로 이동하고, 음수일 경우는 아래쪽으로 이동한다. 이러한 이동 방향이 결정되면 손의 모양을 인식한다. Start 이미지에서 각도가 계산된 끝점과 Stop 이미지에서의 끝점을 비교하여 손의 모양의 변화를 확인할 수 있다.

IV. 성능 평가

1. 실험 환경

본 논문에서는 제안한 가이드라인을 이용한 동적 손 동작 인식 방법의 성능을 평가하기 위해 OpenCV<sup>[12]</sup>

표 1. 시스템 사양  
Table 1. System Specification.

	Spec.
CPU	AMD Athlon 64X2 Dual Core Processor 4800+
Memory	2GB
OS	MS Windows XP Professional SP3
WebCAM	Logitech Communication STX (130만 화소)

표 2. 실험 영상  
Table 2. Experimental Images

	프레임(개수/초)	손 모양	배경
Hand-1	232/23	가위	단순
Hand-2	222/22	바위	
Hand-3	244/24	보	
Hand-4	187/18	가위	복잡
Hand-5	230/23	바위	
Hand-6	229/23	보	
Hand-7	224/23	보→가위	
Hand-8	180/17	가위→보	
Hand-9	222/23	바위→보	

API를 이용하여 손동작 추출과 손동작 검출 알고리즘을 구현한다. 그리고 표 1의 시스템에서 표 2의 실험 영상을 대상으로 프레임 당 평균연산시간, 추적정확도, 검출정확도를 측정하고 그 결과를 분석한다. 또한 CPU 점유율과 메모리 사용량을 측정하고, 그 결과를 분석한다.

2. 실험 결과 및 분석

본 논문에서는 동영상을 프레임별로 분할한 이미지를 이용하여 가이드라인을 이용한 동적 손동작 인식 방법의 손 추적 모듈과 Optical Flow, CAM Shift의 성능을 평가한다. 가이드라인을 이용한 방법의 손 추적 모듈과 Optical Flow, CAM Shift의 프레임당 평균연산시간을 측정된 결과는 그림 7과 같다.

그림 7을 살펴보면 가이드라인을 이용한 손동작 인식 방법의 손 추적 모듈의 프레임 당 평균연산시간이

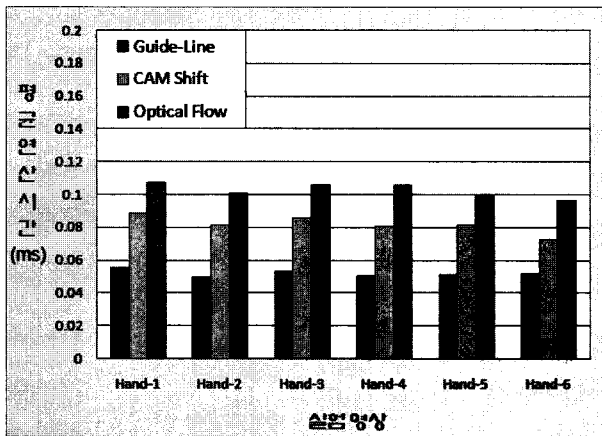


그림 7. 프레임 당 평균연산시간  
Fig. 7. The average execution time per frame.

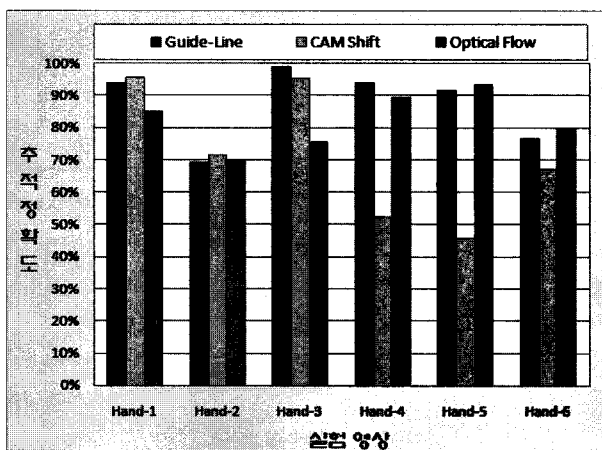


그림 8. 추적 정확도  
Fig. 8. The accuracy of Hand Tracking.

Optical Flow, CAM Shift에 비해 50%정도 향상되었음을 볼 수 있다. 이것은 가이드라인을 이용한 방법의 손 추적 모듈이 프레임 전체가 아닌 손 영역만을 추적하기 때문이다. 또한 손 영역의 윈도우를 새로 계산할 필요가 없이 무게중심과 거리변환 함수를 스킨칼라에 적용하여 새로운 프레임의 손 영역을 적은 연산으로 찾을 수 있기 때문이다.

손 추적 모듈에서 가장 중요한 부분은 추적 정확도이다. 가이드라인을 이용한 방법의 손 추적 모듈과 Optical Flow, CAM Shift의 추적 정확도를 측정된 결과는 그림 8과 같다.

그림 8을 살펴보면 단순 배경인 Hand-1~Hand-3에서 추적 정확도가 비슷하다. 하지만 복잡한 배경인 Hand-4~ Hand-6에서 가이드라인을 이용한 방법의 정확도가 높음을 볼 수 있다. 이러한 결과는 배경 복잡도를 줄이고 YCbCr 칼라공간을 사용했기 때문이다.

복잡한 배경인 Hand-5, Hand-6에서는 Optical Flow 방법이 제안하는 방법에 비해 추적 정확도가 높다. 그림 7을 보면 Optical Flow 방식은 프레임당 평균연산시간이 0.1초를 초과한다. 프레임당 평균연산시간이 0.1초를 초과하면 동영상으로 조합이 되었을 때 초당 10프레임 미만의 영상이 나오므로 실시간 추적에서는 사용하기 힘들다는 단점이 있다.

가이드라인을 이용한 방법의 손 검출 모듈과 Template Matching 방법, HandVu 방법의 프레임당 평균연산시간을 측정된 결과는 그림 9와 같다.

그림 9를 살펴보면 가이드라인을 이용한 방법의 손 검출 모듈의 프레임 당 평균연산시간이 Template Matching 방법, HandVu 방법보다 우수함을 볼 수 있다.

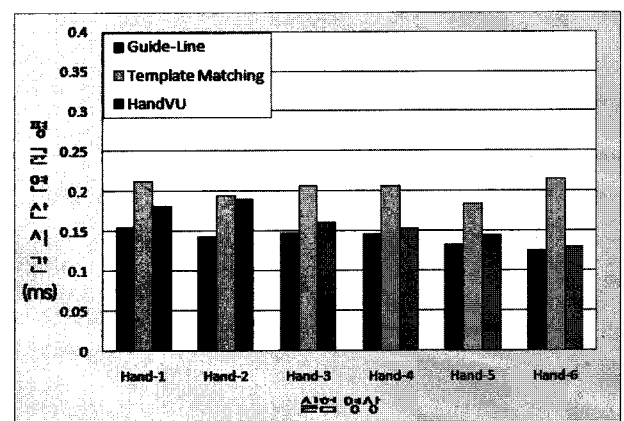


그림 9. 프레임당 평균연산시간  
Fig. 9. The average calculation time per frame.

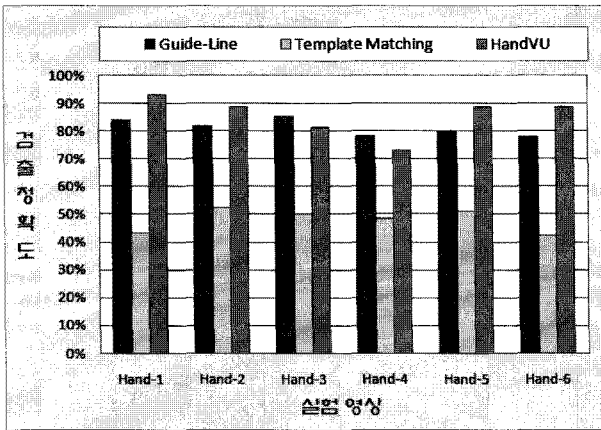


그림 10. 검출 정확도  
Fig. 10. The accuracy of Detection.

표 3. 실험 프로그램  
Table 3. Experimental Programs.

P-1	추적	Guide-Line 방법
	검출	Guide-Line 방법
P-2	추적	Guide-Line 방법
	검출	Template Matching
P-3	추적	Guide-Line 방법
	검출	HandVu
P-4	추적	CAM Shift
	검출	Guide-Line 방법
P-5	추적	CAM Shift
	검출	Template Matching
P-6	추적	CAM Shift
	검출	HandVu
P-7	추적	Optical Flow
	검출	Guide-Line 방법
P-8	추적	Optical Flow
	검출	Template Matching
P-9	추적	Optical Flow
	검출	HandVu

가이드라인을 이용한 방법의 손 검출 모듈과 Template Matching 방법, HandVu 방법의 추적 정확도를 측정 한 결과는 그림 10과 같다.

그림 10을 살펴보면 단순 배경인 Hand-1~Hand-3 과 복잡한 배경인 Hand-4~Hand-6에서 HandVu 방법의 검출 정확도가 가장 우수함을 알 수 있다. 학습 예제를 이용하는 HandVu 방법은 학습 예제의 수에 비례하여 정확도가 증가하기 때문이다. 학습 예제를 이용하지 않는 가이드라인을 이용한 방법의 검출 정확도는 HandVu 방법에 비해 10% 정도 낮다.

손 추적 방법인 가이드라인을 이용한 방법의 손 추적

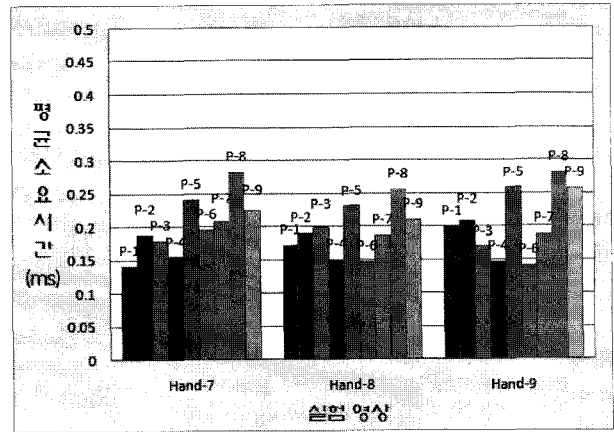


그림 11. 평균소요시간  
Fig. 11. The average turnaround time.

모듈, Optical Flow 방법, CAM Shift 방법과 손 검출 방법인 가이드라인을 이용한 방법의 손 검출 모듈, Template Matching 방법, HandVu 방법을 이용하여 표 3과 같이 9가지 조합을 구성한다.

가이드라인을 이용한 동적 손동작 인식에 소요되는 시간을 측정하기 위해 Start 이미지를 시작으로 Stop 이미지 까지 도달하는데 걸리는 시간을 측정한다. 따라서 동적 손동작 인식에 소요되는 시간(dt)은 식 4를 이용하여 구한다.

$$dt = at + ht + bt \tag{4}$$

식 4에서 at는 start 이미지 검출시간이고, ht는 인식 과정의 손 추적시간이다. 또한 bt는 stop 이미지 검출시간이다.

가이드라인을 이용한 동적 손동작 인식에 소요되는 시간을 측정한 결과는 그림 11과 같다. 그림 11을 살펴보면 CAM Shift 손동작 추적 방법과 가이드라인을 이용한 손동작 검출 방법을 적용한 P-4의 성능이 우수함을 볼 수 있다. 본 논문에서 제안한 손동작 추적 및 검출 알고리즘을 적용한 P-1~P-4, P-7의 경우 적용하지 않은 P-6, P-8, P-9에 비해 성능이 우수함을 볼 수 있다.

가이드라인을 이용한 동적 손동작 인식 방법의 CPU 점유율과 메모리 사용량을 분석한다. CPU 점유율은 VMWare 가상머신을 이용해 운영체제를 새로 설치하고, 필수 프로세서를 제외한 모든 프로세서를 종료한 후에 실험함으로써 정확한 값을 구하였다. 표 2의 손동작 추적 및 검출 방법을 조합한 P-1~P-9에 대한 CPU 점유율을 측정 한 결과는 그림 12와 같다.

그림 12를 살펴보면 손동작 추적 및 검출 방법으로

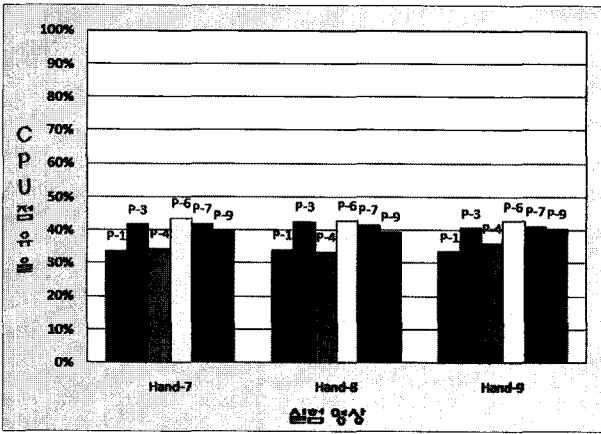


그림 12. CPU 점유율  
Fig. 12. CPU utilization.

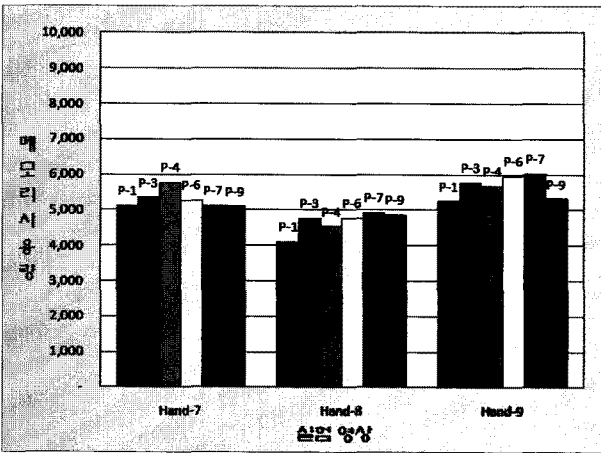


그림 13. 메모리 사용량  
Fig. 13. Memory usage.

본 논문에서 제안한 가이드라인을 이용한 방법을 적용한 P-1의 CPU 점유율이 가장 낮음을 볼 수 있다.

표 3의 손동작 추적 및 검출 방법을 조합한 P-1~P-9에 대한 메모리 사용량을 측정된 결과는 그림 13과 같다. 메모리 사용량은 프레임당 메모리 사용량을 측정하여 구한 평균 메모리 사용량이다. 그림 13을 살펴보면 손동작 추적 및 검출 방법으로 본 논문에서 제안한 가이드라인을 이용한 방법을 적용한 P-1의 메모리 사용량이 가장 적음을 볼 수 있다.

본 논문에서 제안하는 가이드라인을 이용한 손동작 인식 방법은 평균연산시간이 빠르고, CPU 점유율 및 메모리 사용량이 낮다. 따라서 시스템 부하가 높은 환경에 본 논문에서 제안하는 가이드라인을 이용한 손동작 인식 방법을 적용하면 효과적이다.

## V. 결 론

본 논문에서는 가이드라인을 이용한 손동작 인식 방법을 제안하였다. 이 방법은 손 영역 추적 알고리즘과 손 모양을 검출 알고리즘으로 구성된다. 손 영역 추적 알고리즘은 차등분석 방법과 스킨칼라모델 방법의 장점만을 이용하여 손 영역을 찾고, 손 영역에서 무게 중심점을 이용하여 정확하게 손을 추적한다. 이 방법은 기존의 Optical Flow 방법 또는 CAM Shift 방법 보다 속도가 빠르고 정확도가 높다. 손 모양 검출 알고리즘은 가이드라인을 이용하여 얻은 끝점과 중심점 가지고 거리와 각도를 계산한 후에 손 모양을 검출 한다.

본 논문에서는 시뮬레이션을 통하여 가이드라인을 이용한 손동작 인식 방법의 성능을 검증한다. 가이드라인을 이용한 손 모양 검출 방법은 기존의 Template Matching 방법에 비해 인식률이 높고, 인식 속도가 빠르다. 또한 HandVu 방법에 비해 인식률은 비슷하지만 인식 속도는 향상되었다. 그리고 가이드라인을 이용한 손 모양 검출 방법은 기존의 방법에 비해, CPU 점유율이 낮고, 메모리 사용량도 적다. 따라서 본 논문에서 제안하는 가이드라인을 이용한 손동작 인식 방법은 시스템 부하가 높은 환경에 적용하면 효과적이다.

하지만 본 논문에서 제안한 방법은 손의 단면을 사용하기 때문에 손의 각도 변화에 따라 인식률이 저하될 수 있는 단점이 있다. 이러한 문제점은 멀티 카메라를 이용한 다양한 각도의 손 이미지 분석 및 인식을 향상 등에 대한 향후 연구로 보완될 수 있을 것이다.

## 참 고 문 헌

- [1] A. Shamaie and A. Sutherland, "A dynamic model for real-time tracking of hands in bimanual movements," in 5th International Gesture Workshop, Geneva, Apr. 2003.
- [2] 조오영, 김형곤, 고성제, 안상철, "인터랙티브 가상 환경을 위한 손제스처 인식 시스템," 대한전자공학 회논문지-S, 제36권, 제4호, pp. 70-82, 1999.
- [3] Jacek M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, pp. 94-98, 1992.
- [4] S.T. Bow, Pattern Recognition and Image Preprocessing, Marcel Dekker, 1992
- [5] D. M. Gavrilu, "The Visual Analysis of Human Movement: A Survey," CVIU, [1] v.73, no.1,



pp.82-98, 1999.

[6] Robert, T.C., "Mean-Shift Blob Tracking through Scale Space," IEEE CVPR'03, Vol. 1, pp. 234-240, 2003.

[7] Bradski, G.R., "Computer video face tracking for use in a perceptual user interface," Intel Technology Journal, QW2, 1998.

[8] T. Otsuka and J. Ohya, "Recognizing of Facial Expression using HMM with Continuous Output Probabilities," Proceedings of 5th IEEE International Workshop on Robot and Human Communication RO-MAN, pp. 323-328, 1996.

[9] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, Anil K. Jain, "Face Detection in Color Images," IEEE Transactions on pattern analysis and machine intelligence, Vol.23, No.12, pp. 1102-1114, Nov. 1. 2005.

[10] 김건우, 이원주, 전창호, "웹캠을 이용한 손동작 인식방법," 대한전자공학회 2008년 하계종합학술대회 논문집, 제31권, 제1호, pp. 619-620, 2008.

[11] 김건우, 김창현, 전창호, 이원주, "제스처 인식 시스템 설계 및 구현," 한국컴퓨터정보학회 2008년 하계학술대회논문집, 제16권, 제1호, pp. 231-235, 2008.

[12] 정성환, 이문호, 오픈소스 OpenCV를 이용한 컴퓨터 비전실무프로그래밍 기본편, 홍릉과학출판사, 2007.

저 자 소 개



김 건 우(학생회원)  
 2007년 백석대학교  
 정보통신학부 학사.  
 2009년 한양대학교  
 컴퓨터공학과 석사.  
 2010년 현재 한양대학교 컴퓨터  
 공학과 박사과정.

<주관심분야 : 클라우드컴퓨팅, 가상화>



전 창 호(정회원)  
 1977년 한양대학교  
 전자공학과 학사.  
 1982년 Cornell University  
 컴퓨터공학과 석사.  
 1986년 Cornell University  
 컴퓨터공학과 박사.

1977년~1979년 전자통신연구소 연구원.  
 2010년 현재 한양대학교 전자컴퓨터공학부 교수.  
 <주관심분야 : 병렬처리시스템, 성능분석, Grid  
 컴퓨팅, 클라우드컴퓨팅>



이 원 주(정회원)-교신저자  
 1989년 한양대학교  
 전자계산학과 공학사.  
 1991년 한양대학교  
 컴퓨터공학과 공학석사.  
 2004년 한양대학교  
 컴퓨터공학과 공학박사.

현재 인하공업전문대학 컴퓨터정보과 부교수.  
 <주관심분야 : 병렬처리시스템, 성능분석, Grid 컴  
 퓨팅, 클라우드컴퓨팅>