韓國 컴퓨터情報學會 論文誌
第13卷 第5號, 2008. 9.

2008-13-5-1-3

YCbCr정보와 아다부스트 알고리즘을 이용한 실시간 얼굴검출 시스템

김 형 균*, 정 기 봉**

Real-time Face Detection System using YCbCr Information and AdaBoost Algorithm

Kim Hyeong gyun *, Jung gi bong **

요 약

본 논문에서는 실시간 얼굴검출을 위하여 감시카메라에서 입력된 RGB영상을 YCbCr 영상으로 변환한 후 연속된 두 영상의 차를 구하고 Glassfire 라벨링을 실시했다. 라벨링 결과 가장 넓은 구역의 면적과 Area 임계치 값을 비교하여 임계값 이상의 면적이면 동작변환으로 인식하고 영상을 추출하였다. 이렇게 추출된 동작변환 영상을 대상으로 얼굴 검출을 실시하였다. 얼굴 검출에 필요한 특징을 추출하기 위해 AdaBoost알고리즘을 사용하였다.

Abstract

In this paper, we converted an RGB into an YCbCr image input from CCD camera and then after compute difference two consecutive images, conduct Glassfire Labeling. We extract an image become ware of motion-change, if the difference between most broad(area) and Area critical value more than critical value. We enforce the detection of facial characteristics to an extracted motion-change images by using AdaBoost algorithm to extract an characteristics.

▸ Keyword : 얼굴검출(face detection), 아다부스트(AdaBoost), YCbCr

• 제1저자 : 김형균
• 접수일 : 2008. 6. 3, 심사일 : 2008. 8. 1, 심사완료일 : 2008. 9. 25.
* 조선대학교 컴퓨터공학과 ** 목포과학대학 컴퓨터정보과

Ⅰ. 서 론

비디오나 카메라로부터 입력된 동영상 또는 정지영상에서 얼굴의 위치를 정확하게 찾아내는 것을 얼굴검출이라 하는데, 이러한 얼굴검출 기술은 얼굴인식시스템, 출입통제시스템, 보안시스템, 컴퓨터 시각에 의한 자동화 시스템 등에 적용되는 핵심 요소기술로 많이 활용되고 있다.

얼굴검출과 관련된 연구에는 다양한 조명환경, 복잡한 배경, 얼굴 형태의 변화 등으로 인하여 정확한 얼굴검출을 어렵게 하는 중요한 요인들이 존재하여 입력된 영상내에서 얼굴을 정확하게 검출하는 것은 매우 어려운 일이다.

얼굴 영역을 검출하는 기법에는 신경망을 이용한 탐색기법[1], 색상을 이용하여 검출하는 기법[2], 움직임을 감지하여 검출하는 차영상 기법[3], 얼굴영상을 통한 훈련과 약한 분류기의 결합을 통해 강한 분류기를 얻는 아다부스트(AdaBoost) 기법[4] 등이 있다.

신경망 기법은 검출율이 좋으나 속도가 너무 느려 실시간 시스템에는 적용하기 어렵고, 색상을 이용하는 기법은 비교적 빠른 처리속도를 보이나 인종에 따라 색상기준을 따로 마련해야 하고 빛과 주변색상에 대한 영향을 많이 받으므로 다른 알고리즘과 연계해야만 활용이 가능하다. 차영상은 실시간 감시 시스템에 맞게 처리속도가 가장 우수하나 동영상에서만 적용이 가능하며 정확한 얼굴검출을 할 수 없는 단점이 있어 다른 알고리즘과 연계하여 좋은 시스템을 구축할 수 있다.

아다부스트 기법에서 얼굴영상과 얼굴이 포함되지 않은 배경영상이 입력영상으로 들어오게 되면 각각의 입력영상마다 비중이 계산되고, 비중은 웨이블릿 특징에 따라 얼굴과 배경을 구별하게 되며, 얼굴영상을 배경영상으로 오검출한 경우 비중을 다시 계산하게 되고, 이 경우 비중의 차를 고려해 에러율이 가장 낮은 웨이블릿 특징을 고르게 된다. 이러한 낮은 에러율과 높은 검출율을 가지는 분류기를 하나의 특징만을 가지고 있는 약한 분류기를 통해 생성하기에는 문제점이 많다. 또한 아다부스트 기법은 정면 얼굴을 기준으로 10°이상 기울어진 얼굴을 검출하지 못하는 문제점과 실시간 감시 환경에서는 영상에 대한 불필요한 검출로 인해 연산량 증가와 오검출율이 증가하는 문제점이 발견되었다.

본 연구에서는 기존의 얼굴검출시스템이 영상의 명암 변화에 따른 검출률 저하를 보완하기 위하여 웹캠에서 입력된 RGB영상을 YCbCr영상으로 변환하고자 한다. 또한 기존의 얼굴검출 시스템에서 연산양이 많아 실시간 얼굴검출이 어려운 부분을 개선하기 위해 영상처리 기법인 차영상(Difference Picture) 기법을 이용하여 먼저 얼굴검출의 대상이 되는 동작검출 영상을 추출하고자 한다. 이 영상을 대상으로 아다부스트 알고리즘을 통하여 실시간으로 얼굴영역 검출이 가능한 시스템을 제안하고자 한다.

Ⅱ. 실시간 얼굴검출 시스템

2.1 시스템의 구현 범위

얼굴검출을 위하여 사용된 시스템의 구축 환경은 퍼스널 컴퓨터에 부착된 Web Camera(이하 웹캠)를 통하여 연속적으로 입력되는 영상을 대상으로 얼굴을 검출하게 된다. USB 연결 방식의 웹캠으로 입력되는 영상을 직접 화면에 재생한다. 웹캠에서 연속적으로 입력되는 영상은 USB 포트를 통해서 연속영상을 메인보드로 전송하고, 이 전송된 영상들은 WDM과 Direct Show를 기반으로 영상을 재현하는 것이다.

본 연구에서는 기존의 얼굴검출시스템이 영상의 명암 변화에 따른 검출률 저하를 보완하기 위하여 웹캠에서 입력된 RGB영상을 YCbCr영상으로 변환하고자 한다. 인간의 시각 시스템은 색상 간의 변화보다는 밝기의 변화에 더욱 민감하므로 YCbCr색상모델을 사용하면 색체정보에서 밝기 값의 영향을 배제한 데이터를 이용할 수 있어 감시영역의 조명의 변화에 의한 얼굴검출의 에러율을 최소화할 수 있다.

또한 기존의 얼굴검출 시스템에서 연산양이 많아 실시간 얼굴검출이 어려운 부분을 개선하기 위해 영상처리 기법인 차영상(Difference Picture) 기법을 이용하여 먼저 얼굴검출의 대상이 되는 동작검출 영상을 추출하고자 한다. 이 영상을 대상으로 아다부스트 알고리즘을 통하여 얼굴영역을 검출한다.

2.2 실시간 얼굴검출 시스템의 구성

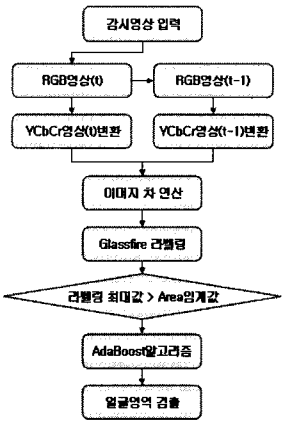

그림 1. 얼굴검출 시스템 흐름도
Fig. 1. Flowchart of face detection system

본 연구에서 제안한 얼굴검출 시스템의 전체적인 구성은 그림 1과 같이 나타낼 수 있다. 실시간 얼굴검출을 위하여 전 단계로 감시카메라에서 입력된 RGB영상을 YCbCr 영상으로 변환한 후 연속된 두 영상의 차를 구하고 Glassfire 라벨링을 실시했다. 라벨링 결과 가장 넓은 구역의 면적과 Area 임계치 값을 비교하여 임계값 이상의 면적이면 동작변환으로 인식하고 영상을 추출하였다. 이렇게 추출된 동작변환 영상을 대상으로 얼굴 검출을 실시하였다. 얼굴 검출에 필요한 특징을 추출하기 위해 AdaBoost알고리즘을 사용하였다.

2.3 YCbCr 영상으로 변환

YCbCr색체모델은 영상 시스템에서 사용되는 색공간의 일종이다. Y는 휘도 성분이며 Cb와 Cr은 색차 성분이다. YCbCr 은 가끔 YCC 라고 줄여 부르기도 한다. YCbCr 은 절대 색공간이 아니며 RGB 정보를 인코딩하는 방식의 하나로, 실제로 보여지는 이미지의 색은 신호를 디스플레이 하기 위해 사용된 원본 RGB 정보에 의존한다[6].

인간의 시각시스템은 색상 간의 변화보다는 밝기의 변화에 더욱 민감하므로 YCbCr색상모델을 사용하면 색체정보에서 밝기 값의 영향을 배제한 데이터를 이용할 수 있어 감시영역의 조명의 변화에 의한 얼굴검출의 에러율을 최소화할 수 있다. RGB 색상모델에서 YCbCr색상모델로 변환은 식(1)과 같다.

$$Y = k_r R + (1 - k_b - k_r) G + k_b B$$

$$C_b = \frac{0.5}{1 - k_b}(B - Y)$$

$$C_r = \frac{0.5}{1 - k_r}(R - Y) \quad \cdots\cdots (1)$$

본 연구에서 차영상 기법을 사용하기 앞서 연속된 두 영상 중에서 직전 영상을 YCbCr영상으로 변환 후 이미지 버퍼에 저장하는 것은 다음과 같은 단계로 이루어진다.

```
// 이미지 버퍼 복사
for(i = 0; i < 이미지 높이; i++){
  for(j = 0; j < 이미지 넓이; j++){
   // 입력된 영상의 RGB 값을 구한다.
   R = 영상의 배열값 중에서 Red 값 추출;
   G = 영상의 배열값 중에서 Green 값 추출;
   B = 영상의 배열값 중에서 Blue 값 추출;

   // YCbCr 컬러공간으로 변환
   Y = (BYTE)(0.299 * R + 0.587 * G + 0.114 * B);
   Cb = (BYTE)((B - Y) * 0.564 + 128);
   Cr = (BYTE)((R - Y) * 0.713 + 128);

   // YCbCr 값을 이미지 버퍼에 저장
   copyBuffer[i][j][0] = Y;
   copyBuffer[i][j][1] = Cb;
   copyBuffer[i][j][2] = Cr;
  }
}
```

본 연구에서는 변환을 위하여 k_b=0.114와 k_r=0.299를 변환 가중치로 선택하였다. 또한 얼굴 색차의 정보의 YCbCr 표본형식은 사용자가 선택할 수 있도록 하였고 기본값은 4:4:4를 기본 표본 형식으로 채택하였고 사용자의 요구에 따라 조정할 수 있도록 구현하였다.

2.4 차영상 기법

본 연구에서 제안하는 얼굴검출 시스템은 기존의 시스템에서 연산양이 많아 실시간 얼굴검출이 어려운 부분을 개선하기 위해 영상처리 기법인 차영상(Difference Picture) 기법을 이용하여 먼저 얼굴검출의 대상이 되는 동작검출 영상을 추출하였다.

차 영상을 직접 사용하여 이동 물체의 영역을 구하는 방법은 잡음이나 카메라의 움직임, 조명 변화의 영향을 심하게 받는다는 단점을 갖는다. 또한 이동 물체의 밝기가 배경과 비슷하거나 느린 속도의 움직임일 경우에는 차 영상에서 이동 물체가 일으킨 변화가 잡음에 의해 발생한 변화보다 크지 않을 경우에 움직임이 전혀 나타나지 않을 수도 있다.

본 연구에서는 이러한 문제를 다루기 위해서 먼저, 입력된 RGB영상을 YCbCr 영상으로 변환하였다. YCbCr색상모델을 사용하면 색체정보에서 밝기 값의 영향을 배제한 데이터를 이용할 수 있어 감시영역의 조명의 변화에 의한 얼굴검출의 에러율을 최소화할 수 있다. 이렇게 YCbCr 영상으로 변환한 후 연속된 두 영상의 차를 구하게 된다.

본 연구에서 YCbCr영상으로 변환 후 연속한 두 영상의 차연산은 다음과 같은 단계로 이루어진다.

```
//YCbCr영상으로 변환후 이미지버퍼와 현재 이미지간 차연산
for(i = 0; i < 이미지 높이; i++){
 for(j = 0; j < 이미지 넓이; j++){
   // 입력된 영상의 RGB 값을 구한다.
   R = 영상의 배열값 중에서 Red 값 추출;
   G = 영상의 배열값 중에서 Green 값 추출;
   B = 영상의 배열값 중에서 Blue 값 추출;

   // YCbCr 컬러공간 변환
   Y = (BYTE)(0.299 * R + 0.587 * G + 0.114 * B);
   Cb = (BYTE)((B - Y) * 0.564 + 128);
   Cr = (BYTE)((R - Y) * 0.713 + 128);
```

```
// 버퍼에 저장된 이전 영상과 현재 영상의 YCbCr값 차연산
  if(abs(Buffer[i][j][0,1,2] - Y,Cb,Cr)  > YCBCR임계값
   {
// 차연산의 결과가 임계값 보다 크면 결과를 검정색으로 저장
     subBuffer[i][j][0,1,2] = 0;
// 차연산의 결과가 임계값 보다 작으면 결과를 흰색으로 저장
   }else
       { subBuffer[i][j][0,1,2] = 255;  }
}}
```

2.5 Glassfire 라벨링

감시영상에서 움직임을 감지하기 위하여 현재 프레임의 영상 정보와 이전 프레임의 영상 정보와의 차영상 기법을 이용하였다. 차영상 기법을 사용하기 위해서 그레이화 시킨 값을 이용하고 차연산 결과가 실험에서 구한 임계값을 넘는다면 움직임이 발생했다고 판단하는데 이때 임계값 비교시 속도를 줄이기 위해서 픽셀 기반 프로세싱 (Pixel-based processing)이 아닌 블록 기반 프로세싱 (Block-based processing)을 한다. 픽셀 기반 프로세싱은 1 대 1 매칭을 수행하기 때문에 초당 많은 프레임을 처리해야 하는 실시간 환경에서는 다소 무리가 있다. 따라서 블록 기반 프로세싱을 하여 정확도는 약간 떨어지지만 속도를 높이는데 중점을 뒀다. 블록 기반 프로세싱이란 픽셀 기반 프로세싱과는 달리 여러 픽셀들을 한 블록으로 정하고 블록 대표 값을 구하여 프로세싱하는 라벨링(labeling)방법을 말한다.

라벨링된 영상에서 임의의 번호를 가진 영역만 추출하면 영역 분리가 이루어지게 되며 특별한 영역에 대해서만 크기, 중심좌표, 원주길이 등을 추출해 내는 것이 가능하게 된다. 라벨링 단계는 이진화된 영상을 탐색하다가 밝기가 255인 화소값을 만나면 라벨링을 수행하고 본 연구에서는 이 라벨링을 점을 8-근방 중심으로 이동 후 다시 인접화소의 미방문 255 화소값을 라벨링하는 방식으로 반복한다. 영역 라벨링의 단계는 그림 2와 같이 수행된다.

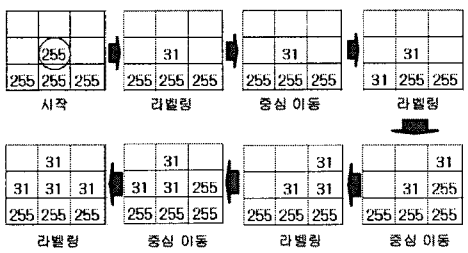

그림 2. 영역 라벨링 단계
Fig. 2. step of area labelling

이진화된 영상으로부터 라벨링 알고리즘 식 2를 이용하여 해당 영상을 라벨링한다.

$$P(R_i^{(k)} \cup x) = \begin{cases} TRUE & \text{if } x > T \\ FALSE & otherwise \end{cases}, \text{for } i = 1,2,...N, \quad \cdots\cdots (2)$$

P는 (R,x,T)의 논리적 판단 자, (k)는 각 단계, x는 영상의 픽셀 밝기 값, T는 임계치 값을 나타낸다. 각 영역 R은 region growing 알고리즘에 의하여 라벨링 된다.

Glassfire 알고리즘은 자기호출(recursive call)을 이용하여 모든 인접요소가 라벨링될 때까지 현재 관심화소의 주변 인접화소를 차례로 검사하면서 라벨링을 한다. 또한 자기홀출에 있어서 스택 자료구조를 사용하기 때문에 고속으로 아주 큰 영역을 라벨링하는 것도 가능하다.

본 연구에서는 입력된 RGB영상을 YCbCr 영상으로 변환한 후 연속된 두 영상의 차를 구하고 Glassfire 라벨링을 실시했다. 라벨링 결과 가장 넓은 구역의 면적과 Area 임계치 값을 비교하여 임계값 이상의 면적이면 동작변환으로 인식하도록 하였다.

Area 임계치 값은 기본값 1280으로 할당되어 있고 사용자의 요구에 따라 변경할 수 있도록 구현하였다.

```
//라벨링된 영상을 저장하기 위한 배열의 메모리 할당
short *coloring = new short [height*width];
// 모든화소를 미방문점으로 일단마킹
for (i=0; i<height*width; i++) coloring[i]=0;
// 입력영상의 라벨링
for (i=0; i<height; i++)
  for (j=0; j<width; j++)
    // 물체영역(255)이고 미방문점이라면 라벨링 시작
     if (m_InImg[i][j]==255 && coloring[i*width+j]==0)
     {
        curColor++; // 현재의Color값
        grass(coloring, height, width, i, j, curColor);
     }
  grayGap = 250/curColor;
// 라벨링된 데이터를 m_OutImg 배열을 이용하여 화면출력
for (i=0; i<height; i++){
  for (j=0; j<width; j++){
      intvalue = (int)(coloring[i*width+j]*grayGap);
      if (value==0) m_OutImg[i][j] = 255;
      else m_OutImg[i][j] = value;
}}
// grass() 함수 기능 정의
for (k=i-1; k<=i+1; k++) { // 8 근방
  for (l=j-1; l<=j+1; l++) {
     // 영상의 경계를 벗어나면 라벨링하지 않음
     if (k<0 || k>=height || l<0 || l>=width) continue;
```

```
    index = k*width+l;
// 값이 물체영역(255)이고 미방문 픽셀이라면 라벨링 함
    if (m_InImg[k][l]==255 && coloring[index]==0){
        coloring[index] = curColor;
        grass(coloring, height, width, k, l,curColor);
}}}
```

2.6 아다부스트 기법을 이용한 얼굴검출

본 연구에서는 앞서 실시간 얼굴검출을 위한 전 단계로 감시카메라에서 입력된 RGB영상을 YCbCr 영상으로 변환한 후 연속된 두 영상의 차를 구하고 Glassfire 라벨링을 실시했다. 라벨링 결과 가장 넓은 구역의 면적과 Area 임계치 값을 비교하여 임계값 이상의 면적이면 동작변환으로 인식하고 영상을 추출하였다. 이렇게 추출된 동작변환 영상을 대상으로 얼굴 검출을 실시하였다.

얼굴 검출에 필요한 특징을 추출하기 위해 AdaBoost알고리즘을 사용하였다. AdaBoost알고리즘의 기본 개념은 약한 분류기(Weak classifier)를 선형적으로 결합하여 최종적으로 높은 검출 성능을 가진 강한 분류기(Strong classifier)를 생성하는 것이다. AdaBoost알고리즘에 의해 생성된 강한 분류기는 그림 3과 같이 계층적인 체계를 이룬다. 이것은 기존의 다른 방법들이 하나의 복잡한 마스크 형태의 분류기를 이용한 것과 달리 간단한 마스크를 여러 개의 층으로 형성한 것이다. 기존의 하나의 복잡한 분류기들은 실제 영상에서 얼굴을 검출하는데 복잡한 마스크와의 계산이 이루어지기 때문에 계산량이 많고 시간이 많이 소요되었다. 그러나 AdaBoost알고리즘을 이용하여 생성된 계층적 분류기는 앞쪽 부분에 간단하면서도 얼굴을 가장 잘 검출하는 것을 배치하고 뒤쪽 부분에 잘못 검출된 제거하는 형식으로 구성되어 있다. 이는 실시간 검출에 있어서 기존의 방법에 비하여 뛰어난 성능을 보인다. 이런 점을 고려하여 실시간으로 얼굴을 검출하고자 본 연구에서는 AdaBoost알고리즘을 사용하였다.

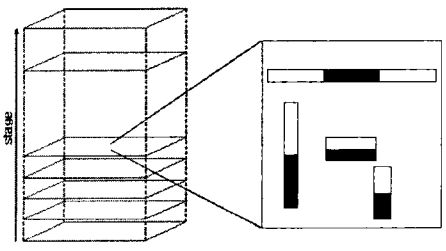

그림 3. 계층적 분류기
Fig. 3. Hierarchical Classifier

강한 분류기는 약한 분류기의 선형적 결합형태로 여러 개의 특징을 약한 분류기를 결합해서 실질적으로 얼굴의 패턴을 구별하는 역할을 한다. 본 연구에서 사용한 아다부스트 알고리즘은 다음과 같다.

1. 입력
훈련영상 집합 S = (x1, y1),...,(xn, yn),
여기서, $x_i \in R^k$, $y_i \in \begin{cases} 0 & \text{배경영상} \\ 1 & \text{얼굴영상} \end{cases}$

2. 비중 초기화
$$\omega_{1,i} = \begin{cases} \frac{1}{2m} & \text{for } y_i = 0 \\ \frac{1}{2l} & \text{for } y_i = 1 \end{cases}$$
여기서, m은 훈련영상집합 S내의 배경영상의 개수이며, l은 얼굴영상개수.

3. 반복 t = 1, ..., T
(a) 비중의 표준화,
$$\omega_{t,j} = \frac{\omega_{t,i}}{\sum_{j=1}^{n} \omega_{t,j}}$$
여기서, ωt,i 는 t번째 약한 분류기에 입력되는 i번째 훈련영상의 비중을 의미.
(b) 약한 분류기(hj)의 에러 εj,
$$\epsilon_j = \sum_i \omega_i |h_j(x_i) - y_i|$$
(c) 분류기의 선택,
가장 낮은 에러율을 εt라 하고, εt를 가지는 약한 분류기 hj를 선택.
(d) 비중 업데이트,
$$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i}$$
여기서, 입력영상 xi가 hj에 값에 따라 올바르게 분류가 된 경우 ei = 0,
올바르게 되지 않은 경우 ei = 1, 그리고 $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

4. 최종 강한 분류기,
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & Otherwise \end{cases}$$
여기서, $\alpha_t = \log \frac{1}{\beta_t}$

여기서 입력값 중 x는 훈련영상을 y는 얼굴과 비얼굴을 나타낸다. 훈련 영상의 앞부분은 얼굴 영상이며 뒷부분은 비얼굴 영상이다. 2단계에서 가중치(w)를 초기화한다. 가중치(w)는 각각의 훈련 영상의 중요도를 나타내고 후에 학습과정에서 어느 영상을 더 중요하게 학습할 것인지를 결정하는 역할을 한다.

3단계는 약한 분류기를 생성하는 단계이다. 약한 분류기는 최소의 에러를 갖는 하나의 특징만을 선택하나 본 연구에서는 효율성을 고려하여 에러의 임계치를 주어 한 단계에서 여러 개의 특징을 선택하도록 하였다. 이는 후에 강한 분류기가 계층적으로 분류되는 역할을 한다. 그리고 T는 단계를 나타내는 역할을 한다. (b)에서 j는 특징의 수를 나타내며 j만큼 반복하면서 임계치보다 높은 에러값을 찾는다. 에러값은 positive일 경우와 negative일 경우를 나누어서 계산하였다. 이는 학습 초기에는 positive의 에러율을 줄이는데 집중하고 후반부에는 negative의 에러율을 줄이기 위함이다. (d)에서는 가중치를 업데이트하는 부분이다. 이때 잘못 분류된 훈련 영상은 가중치 w(i)를 증가시키고, 옳게 분류된 훈련 영상은 가중치 w(i)를 감소시킨다. 이것은 초기에 선택된 특징들은 쉽게 얼굴 영상과 비얼굴 영상을 구별할 수 있는 역할을 담당하지만 후반부에 선택된 특징들은 얼굴 영상과 비얼굴 영상의 구별이 어려운 것을 구별하는 역할을 담당하기 위함이다.

그림 4는 본 연구에서 제안한 시스템에 의하여 얼굴을 검출한 결과를 보여주고 있다.

그림 4. 얼굴검출 결과
Fig. 4. result of face detection

III. 실험 및 분석

본 연구에서 제안한 얼굴검출 시스템은 실시간 얼굴검출을 위하여 전 단계로 감시카메라에서 입력된 RGB영상을 YCbCr 영상으로 변환한 후 연속된 두 영상의 차를 구하고 Glassfire 라벨링을 실시했다. 라벨링 결과 가장 넓은 구역의 면적과 Area 임계치 값을 비교하여 임계값 이상의 면적이면 동작변환으로 인식하고 영상을 추출하였다. 이렇게 추출된 동작변환 영상을 대상으로 얼굴 검출을 실시하였다. 얼굴 검출에 필요한 특징을 추출하기 위해 AdaBoost알고리즘을 사용하였다. 이렇게 제안된 시스템의 성능을 검증하기 위해서 일반 퍼스널 컴퓨터와 웹캠(webcam)을 이용하여 실험을 실시하였다. 본 실험에서는 1 초당 30 프레임을 입력으로 받았다.

3.1 감시자 1명인 경우 실험 결과

먼저 감시자가 1명인 경우 기존의 아다부스트 기법과 제안한 기법과의 검출율을 비교한 결과 표 1과 같이 제안한 기법이 기존의 기법에 비해 4.1% 높게 나타났다.

표 1. 얼굴 검출율 비교 1
Table 1. compare of face detection rate 1

구 분		감시자수 1명
총 횟수		123
검출성공	제안기법	116
	기존기법	111
검출실패	제안기법	7
	기존기법	12
검출확률	제안기법	94.3%
	기존기법	90.2%

3.2 감시자 2명인 경우 실험 결과

감시자가 2명인 경우 기존의 아다부스트 기법과 제안한 기법과의 검출율을 비교한 결과 표 2와 같이 제안한 기법이 기존의 기법에 비해 5% 높게 나타났다.

표 2. 얼굴 검출율 비교 2
Table 2. compare of face detection rate 2

구 분		감시자수 2명
총 횟수		120
검출성공	제안기법	111
	기존기법	105
검출실패	제안기법	12
	기존기법	15
검출확률	제안기법	92.5%
	기존기법	87.5%

3.3 감시자 3명인 경우 실험 결과

감시자가 3명인 경우 기존의 아다부스트 기법과 제안한 기법과의 검출율을 비교한 결과 표 3과 같이 제안한 기법이 기존의 기법에 비해 5.1% 높게 나타났다.

표 3. 얼굴 검출율 비교 3
Table 3. compare of face detection rate 3

구 분		감시자수 3명
총 횟수		116
검출성공	제안기법	102
	기존기법	96
검출실패	제안기법	14
	기존기법	20
검출확률	제안기법	87.9%
	기존기법	82.8%

3.4 감시자 4명 이상인 경우 실험 결과

다음으로 감시자수를 4명 이상으로 한정한 경우 얼굴을 검출하였고 검출된 얼굴 영상을 캡처하였다. 감시자가 4명 이상인 경우 기존의 아다부스트 기법과 제안한 기법과의 검출율을 비교한 결과 표 4와 같이 제안한 기법이 기존의 기법에 비해 11.7% 높게 나타났다.

표 4. 얼굴 검출율 비교 4
Table 4. compare of face detection rate 4

구 분		감시자수 4명 이상
총 횟수		94
검출성공	제안기법	62
	기존기법	51
검출실패	제안기법	32
	기존기법	43
검출확률	제안기법	65.9%
	기존기법	54.2%

V. 결론

얼굴 검출은 대부분 얼굴의 움직임 정보를 이용한다. 기존에 얼굴 검출 방법은 프레임 간의 차를 이용하여 움직임을 검출하는 방법이 사용되어 왔으나 대부분이 실시간을 고려하지 않은 수학적 접근법을 사용하거나 알고리즘이 지나치게 복잡하여 실시간 구현에 용이하지 않았다.

본 연구에서는 실시간 얼굴검출을 위하여 전 단계로 감시카메라에서 입력된 RGB영상을 YCbCr 영상으로 변환한 후 연속된 두 영상의 차를 구하고 Glassfire 라벨링을 실시했다. 라벨링 결과 가장 넓은 구역의 면적과 Area 임계치 값을 비교하여 임계값 이상의 면적이면 동작변환으로 인식하고 영상을 추출하였다. 이렇게 추출된 동작변환 영상을 대상으로 얼굴 검출을 실시하였다. 얼굴 검출에 필요한 특징을 추출하기 위해 아다부스트 기법을 사용하였다.

제안된 시스템의 성능을 검증하기 위해서 일반 퍼스널 컴퓨터와 웹캠(webcam)을 이용하여 실험을 실시하였다. 본 실험에서는 1 초당 30 프레임을 입력으로 받았다. 사람 수에 따른 얼굴검출 결과는 한 명일 경우 검출되는 확률이 94.3%, 두 명일 경우 검출되는 확률이 92.5%, 세 명일 경우 검출되는 확률이 87.9%, 네 명 이상일 경우 검출되는 확률은 65.9% 이다. 기존의 기법과의 비교 결과를 살펴보면 감시자가 1명인 경우 제안한 기법이 기존의 기법에 비해 4.1%, 감시자가 2명인 5%, 감시자가 3명인 경우 5.1%, 감시자가 4명 이상인 경우 제안한 기법이 기존의 기법에 비해 11.7% 높게 나타났다.

향후에는 실시간 얼굴검출 시 측면얼굴에 대한 검출율을 보다 향상시키도록 해야 할것이며, 본 시스템을 응용하여 걸어 다니는 사람, 자동차 등 다양한 다른 물체의 검출에도 적용이 가능할 수 있을 것이다.

참고문헌

[1] H. Rowley, S. Baluja, T. Kanade, "Neural network-based face detection," In IEEE Patt. Anal. Mach. Intell, 20, pp 22-38, 1998.

[2] E. Osuna, R. Freund, and F. Girosi, "Traning support vector machines: An application to face detections," In IEEE Proc. of Int. Conf. on Computer Vision and Pattern Recognition, 6, 1997.

[3] A. L. Yuille, P. W. Hallinan, and D. S. Cohen, "Feature extraction from faces using deformable templates," Int. J. Comput. Vision, 8, pp 99-111, 1992.

[4] J. L. Crowley and F. Berard, "multi-model tracking of faces for video communications," In IEEE Proc. of Int, Conf. on Computer Vision and Pattern Recognition, Puerto Rico, June. 1997.

[5] H. Graf, E. Cosatto, and T. Ezzat, "Face analysis for the synthesis of photo-realistic talking

heads," In Proceedings Fourth IEEE International Conf on Automatic Face and Gesture Recognition, 2000.
[6] R. Herpers, M. Michaelis, "Edge and keypoin detection in fatial regions," In IEEE Proc. of 2nd Int. Conf. on Automatic Face and Gesture Recognition, pp 212-217, Oct. 1996.
[7] Paul Viola, M. Jones, "Robust real-time object detection," International Conference on Computer Vision, 2001.
[8] Yoav Feund, Robert E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," In computational Learning Theory : Eurocolt'95, pp 23-37, 1995.
[9] A. Haar, "Zur theorie der orthogonalen funktionen system," Math, Ann, 69, pp 331-371, 1910.
[10] K. R. Catleman, Digital Image Processing, Prentice -Hall, 1996.

저 자 소 개

김 형 균

2004년 2월 : 조선대학교 대학원 컴퓨터공학과 공학박사
2002년~ 2007년 8월 : 동강대학 컴퓨터인터넷과 초빙전임강사
2007년 9월 ~ 현재 : 호남대학교 공학교육혁신센터 선임연구원, 조선대학교 컴퓨터공학과 외래강사
관심분야 : 모바일 응용, 영상처리

정 기 봉

2003년 2월 : 조선대학교 대학원 컴퓨터공학과 공학박사
2007년 2월 : 목포과학대학 컴퓨터정보과 전임강사
관심분야 : 영상처리, 가상현실,