

논문 2009-46SD-9-6

휴대용 멀티미디어 기기를 위한 실시간 얼굴 추적 시스템

(Real-Time Face Tracking System for Portable Multimedia Devices)

윤 석 기*, 한 태 회**

(Suk-Ki Yoon and Tae Hee Han)

요 약

사람의 얼굴 추적은 디지털 캠코더, 디지털 카메라, 휴대폰 등과 같은 휴대용 멀티미디어 장치에 대해 점차 중요한 이슈가 되어 왔다. 갈수록 확대되어 가는 얼굴 추적 응용 서비스 요구에 대해 소프트웨어 구현 대응은 성능 및 전력 소모 면에서 한계가 있다. 따라서 본 논문에서는 실시간으로 동작할 수 있는 하드웨어 기반의 저전력 얼굴 추적 시스템을 제안하고자 한다. 제안된 시스템은 FPGA 프로토타이핑과 삼성 65nm CMOS 공정으로 구현하여 검증하였고, 8.4 msec 미만의 추적 속도와 15만 게이트의 크기를 가지며 평균 20 mW의 동작 전력소모를 보여 실시간으로 동작하는 저전력 휴대용 멀티미디어 기기에 적합함을 입증하였다.

Abstract

Human face tracking has gradually become an important issue in applications for portable multimedia devices such as digital camcorder, digital still camera and cell phone. Current embedded face tracking software implementations lack the processing abilities to track faces in real time mobile video processing. In this paper, we propose a power efficient hardware-based face tracking architecture operating in real time. The proposed system was verified by FPGA prototyping and ASIC implementation using Samsung 65nm CMOS process. The implementation result shows that tracking speed is less than 8.4 msec with 150K gates and 20 mW average power consumption. Consequently it is validated that the proposed system is adequate for portable multimedia device.

Keywords : Face Tracking, Low Power, Real Time

I. 서 론

얼굴의 검출, 추적과 인식은 디지털 캠코더, 디지털 카메라, 휴대폰과 같은 휴대용 멀티미디어 장치나 감시 시스템 등에서 매우 중요한 기술이다. 이들 장치로부터 얻어진 이미지로부터 사람의 얼굴은 대다수의 경우에 있어서 초점이나 노출, 화이트밸런스나 다른 상황인식(context-aware)과 같은 이미지 후처리 기술의 목표가 된다. 나아가 많은 수의 제품들은 얼굴의 주체가 누구인지에 대한 얼굴 정보를 사용해 카메라나 보안 등의 응용 프로그램을 개발해 왔다.

얼굴 검출은 다양한 얼굴 관련 응용 제품에 대해 첫 번째의 중요한 단계이다. 얼굴 검출 단계에서 하는 일은 하나의 이미지나 연속적인 비디오 영상에서 하나 이상의 얼굴들의 위치나 크기를 찾는 일이다.

얼굴 추적은 얼굴 인식을 위한 두 번째 단계일 뿐만 아니라 얼굴 기반 Auto Focusing (AF), Auto White Balance (AWB)와 Auto Exposure (AE)를 위해 반드시 필요한 과정으로, 영상에서 얼굴의 위치나 방향, 그리고 환경적 조건에 상관없이 얼굴을 포함하는 모든 영상 부분을 추적하는 과정이라고 정의된다.

얼굴 인식은 신원확인을 위한 데이터베이스 내의 모든 얼굴들의 특징들에 따라서 입력 얼굴 영상을 비교하기 위해 필요하다. 정확한 얼굴 인식은 많은 보안 응용 기기들에게 매우 중요하다. 우리는 이 세 가지 요소 기

* 학생회원, ** 평생회원, 성균관대학교 정보통신공학부
(School of Information and Communication
Engineering, Sungkyunkwan University)
접수일자: 2009년5월14일, 수정완료일: 2009년9월3일

술 중에서 얼굴 추적에 집중할 것이다.

얼굴 추적 알고리즘은 지난 수십 년 간에 걸쳐서 성능과 속도 측면에서 비약적인 발전을 해 왔다. 그러나 이동식 멀티미디어 환경에서 소프트웨어 구현으로는 실시간 동작이 어려울 뿐만 아니라 전력 소모도 크다. 그러므로 하나의 SoC (System-on-Chip) 안에 집적될 수 있는 빠르고 작은 하드웨어 구현이 필수 요소로 요구되어지고 있다.

본 논문에서는 저전력 실시간 하드웨어 기반 얼굴 추적 시스템을 제안한다. 이 시스템의 주된 목적은 이동식 멀티미디어 시스템에서 얼굴 기반 AF, AWB 그리고 AE 동작을 가능하게 하는데 있다. 나아가, Adaboost 알고리즘^[1]과 색상 기반 평균이동 (mean-shift) 알고리즘^[2]의 적용은 다측면 (Multi-view) 얼굴 추적 성능을 향상시킨다.

논문은 다음과 같이 구성된다. II장은 알고리즘 관련 연구와 제안된 알고리즘을 비교 설명한다. 제안된 전체 하드웨어 얼굴 추적 시스템은 III장에서 보인다. IV장에서는 제안된 시스템의 실험결과를 나타내고, 마지막으로, V장에서 결론을 맺는다.

II. 관련 연구 및 제안된 알고리즘

1. 얼굴 추적 방법

효율적인 얼굴 검출 및 추적을 위해서 많은 연구들이 수행되어 왔다. 최근에 연산량이 적은 알고리즘이 Viola 와 Jones^[1]에 의해서 제안되었다. Adaboost라 불리는 이 알고리즘은 얼굴을 실시간으로 빠르게 검출하기 위해 연속된 약 분류기를 사용한다. 그러나 이 알고리즘은 오직 정면 얼굴만 검출 가능하여 다측면 얼굴을 검

출 할 수 없다.

Huang^[3]은 다측면 얼굴 검출을 위해 벡터 부스팅 방법을 제안했으나 계산량과 필요 메모리가 너무 커서 움직이는 타겟에 대한 검출과 추적을 실시간으로 수행하기 어려웠다. Choi^[4]는 비디오에서 다측면 얼굴을 추적하기 위해 얼굴 검출의 결과를 추적 시에 이용하여 효과적인 가중치를 둔 형상 질감(shaped-texture weighted) 알고리즘을 만들었으나, 하드웨어로 구현 시 면적이 너무 커져서 사용하기가 불가능하다. [2]에서는 커널 기반 물체 추적 방법을 사용했는데, 이 알고리즘은 평균이동 알고리즘에 기반을 둔 것이다. 이 추적기의 핵심은 구역 매칭(region matching)이고, 매칭 기준은 색상 히스토그램을 기본으로 한 유사도이다. 얼굴 후보의 색상 분포는 추적을 위해 피부색 모델의 색상 분포와 비교되는데, 색상 기반 방법은 계산량 면에서 효율적이다, 그러나 이 방법은 커널 계산 복잡도가 상당히 증가하는 단점이 있어서 실시간으로 동작가능하도록 커널 기반 객체 추적 알고리즘을 발전시켰다^[5]. 제안하는 논문은 [5]의 방법의 연장선에 있다. 이 알고리즘의 구현은 실시간으로 다측면 얼굴 추적을 가능하게 함으로써, Adaboost 얼굴 검출 알고리즘의 단점을 보완하고, false negative를 제거하는 성능 향상이 가능하다.

2. 다측면 얼굴 추적

다측면 얼굴 추적은 동영상상을 기반으로 한 휴대용 멀티미디어 기기에서 다양한 얼굴의 움직임을 추적해야 하기 때문에 필수적이나, 계산량 및 메모리 필요량이 급격히 증가하는 문제가 있다. Adaboost 방법을 이용해 하드웨어로 구현된 정면 얼굴 검출기조차 학습된 얼굴의 패턴 매칭을 위한 시간이 많이 걸려서, 다른 응용 프로그램을 위해 한 프레임 내에 검출을 완료하기 쉽지 않다^[6].

따라서, 본 논문은 안정적인 다측면 얼굴 추적과 계산량 감소를 위해 얼굴 검출과 추적을 통합 구현하는 전략을 취했다. 초기 얼굴 좌표의 획득을 위해 일정 간격의 첫 프레임에만 얼굴 검출기를 사용하여 초기 얼굴 좌표를 얻고, 뒤에 이어지는 프레임에는 검출 결과를 바탕으로 얼굴 추적을 실시하는 방법을 사용한다. 이 방법을 통해서, 기존의 독립적인 방법에 비해 세 가지 이득을 얻을 수 있다.

- 얼굴 검출에 비해 얼굴 추적은 빠른 시간 내에 결과를 도출하므로, 병합된 추적 결과를 바탕으로

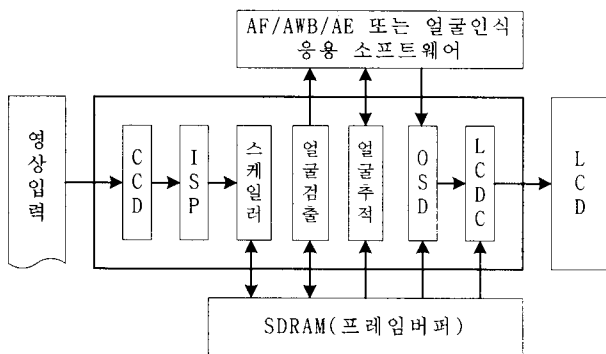


그림 1. 멀티미디어 얼굴 추적 시스템
Fig. 1. Multimedia Face Tracking System.

로 AE, AF, AWB와 같은 응용 프로그램이 동작할 수 있는 충분한 처리 시간 이득을 줄 수 있다.

- 얼굴 검출의 false negative(miss) 횟수를 줄일 수 있다. 선 검출 방법을 사용한 얼굴 검출 방법은 조명의 변화와 같은 환경에서 정확한 얼굴의 검출 어려워져 얼굴인 부분을 얼굴이 아닌 것으로 검출 실패할 수가 있다. 하지만, 현재의 얼굴 검출 결과에 이전 검출 좌표와 현재의 추적 결과를 병합함으로써, 이와 같은 단점을 보상해 준다.
- 색상 정보를 같이 사용함으로써, 명암 정보만을 사용하는 정면 얼굴 검출기인 Adaboost 방법의 한계를 보완하여 다측면 얼굴 추적이 가능하다.

그림 2는 얼굴 검출기와 추적기를 병합하여 동작하는 전체 얼굴 추적 시스템의 순서도와 시간적 관계를 보여 준다. 연속적으로 들어오는 비디오 영상에서 일정 프레임 간격마다 얼굴 검출기가 동작해서 얼굴 검출 좌표들을 얼굴 추적기에 전달하면 얼굴 검출기가 동작하지 않는 나머지 프레임에서 얼굴 추적기가 동작하고, 두 결과를 토대로 최종 얼굴 좌표를 병합하는 방식이다.

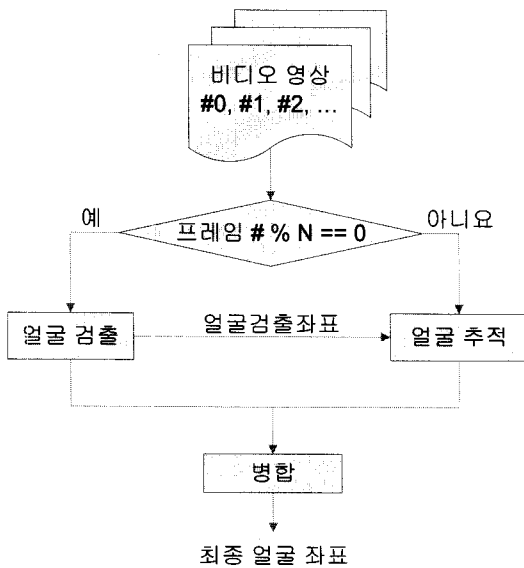


그림 2. 얼굴 검출과 얼굴 추적의 결합
Fig. 2. Integration of Face Detection and Face Tracking.

3. 커널 기반 객체 추적 알고리즘

커널 기반 객체 추적은 타겟을 특성화시키기 위해 특징 영역을 선택한다. 우리는 특징 영역으로서 YCbCr 색상 공간을 선택하였다. 색상 히스토그램은 식 (1), (2) 각각에 의해 생성된다.

$$q_u = \sum_{l=1}^n \delta[b(x_l) - u] \quad (1)$$

$$p_u = \sum_{l=1}^n \delta[b(x_l) - u] \quad (2)$$

여기서 q_u 는 각 색상에 대한 얼굴 모델 색상 히스토그램, p_u 는 얼굴 후보 색상 히스토그램, x_l 는 모델 및 후보를 구성하는 픽셀 위치, $b(x_l)$ 는 각 픽셀에 대한 bin 값, u 는 픽셀 색상을 각각 의미한다. 이 때, 식 (1), (2)는 식 (3)과 같이 간략화된 형태로 나타낼 수 있다.

$$\{q_u\}_{u=1..m} = \text{histogram}(Cb \gg 4, Cr \gg 4) \quad (3)$$

$Cb \gg 4$ 와 $Cr \gg 4$ 는 Cb 와 Cr 값을 오른쪽으로 4번 쉬프트 시키는 것을 의미하며, m 은 0에서 255까지의 나 타낸다. Cb, Cr 의 2차원이었던 색상을 새로운 1차원의 값(u)으로 표현함으로써, 계산량을 낮추게 된다. $\{q_u\}_{u=1..m}$ 와 얼굴 중심 좌표, y_0 가 주어졌을 때, 이동된 얼굴의 중심 좌표, y_1 에 대한 추적 과정은 다음과 같다.

for $k = 1$: 반복횟수

Step 1. 이전 좌표의 색상 히스토그램, $\{P_u(y_0)\}_{u=1..m}$ 과 얼굴 모델과의 L1 거리인 $\rho(y_0)$ 를 계산한다. L1 거리는 얼굴 모델 색상 히스토그램과 얼굴 후보 색상 히스토그램 간의 유사도를 나타내고, 식 (4) 같이 정의된다.

$$\begin{aligned} \rho(y_0) &= \sum_{u=1}^m \left| \frac{p_u(y)}{N_{p(y)}} - \frac{q_u}{N_q} \right| \\ &= \frac{1}{N_{p(y)} N_q} \sum_{u=1}^m |N_q p_u(y) - N_{p(y)} q_u| \end{aligned} \quad (4)$$

$\rho(y_0)$ 는 얼굴 모델과 얼굴 후보와의 거리, N_q 는 얼굴 모델의 픽셀 수, $N_{p(y)}$ 는 얼굴 후보의 픽셀 수를 각각 의미한다. 스케일링 다운되어 모델과 후보 간의 얼굴 크기가 다르기 때문에 각각의 얼굴 분포를 픽셀 수로 나눠 줌으로써

정규화된 크기로 상이한 얼굴 크기간의 거리를 구할 수 있다.

Step 2. 각 픽셀에 대한 가중치, $\{w_i\}_{i=1..m}$ 를 계산한다. 각 픽셀의 가중치는 현재 프레임에 얼굴 모델의 색상 bin이 존재하면 커지고, 존재하지 않으면 0이 된다.

Step 3. 후보 얼굴의 새로운 중심좌표(y_1)를 가중치 평균(Weighted average)을 구하는 식 (5)에서 예측하고, 새로운 중심 좌표로 얼굴 후보 히스토그램, $\{P_u(y_1)\}_{u=1..m}$ 을 업데이트한 뒤, 얼굴 모델과의 거리, $\rho(y_1)$ 을 계산한다.

$$y_1 = \frac{\sum_{i=1}^{n_{h0}} w_i x_i}{\sum_{i=1}^{n_{h0}} w_i} \quad (5)$$

Step 4. $\rho(y_0) < \rho(y_1)$ 면 추적 루틴을 중단하고, 그렇지 않으면, 새로운 중심이동 값인 y_1 을 y_0 에 대입한 뒤, step 1으로 반복한다.

그림 3은 얼굴 추적 순서도를 나타낸다. 얼굴 검출이 이루어진 후에, 얼굴 추적이 동작하는 부분이며, 최종 얼굴 추적 결과는 그림 2에서 보았듯이, 얼굴 검출 결과와 병합하여 판단하게 된다.

III. 얼굴 추적 시스템의 하드웨어 구현

얼굴 추적 성능 개선을 위해서 에지를 사용하는 얼굴 검출기와 연동하여 기존의 에지만이나 또는 색상만으로 검출 및 추적하던 알고리즘을 상호 보완하였으며, 저전력을 위해 메모리 액세스를 줄이고자 했고, 메모리와 로직의 동작의 분석을 통해 면적을 감소시키는데 주안점을 두었다.

영상 크기는 디지털 영상 프로토콜인 ITU-R 656 포맷을 지원해야 하므로, 인터레이스(Interlace) 방식으로 들어오는 PAL(720x576)과 NTSC(720x480)의 입력 영상에서, 실제 추적기 입력은 720x288 또는 720x240 크기의 60Hz 필드(field)가 된다. 여기서, 짝수나 홀수 필드의 선택은 둘 중 처음 추적을 시작한 필드만 계속 선택되게 된다. 하드웨어는 최대 입력을 기준으로 PAL을 수평으로 1/4, 수직으로 1/2 스케일링 다운하여 180x144로 정하고, 최대 추적 가능한 얼굴 수는 10명으로 설계

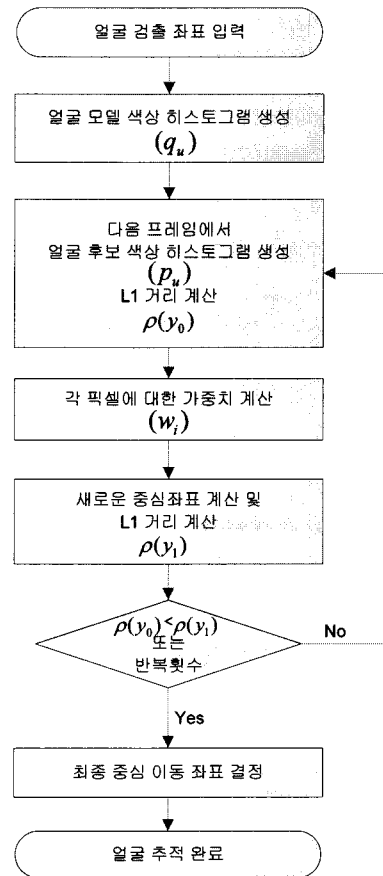


그림 3. 얼굴 추적 순서도
Fig. 3. Flow Chart of Face Tracking.

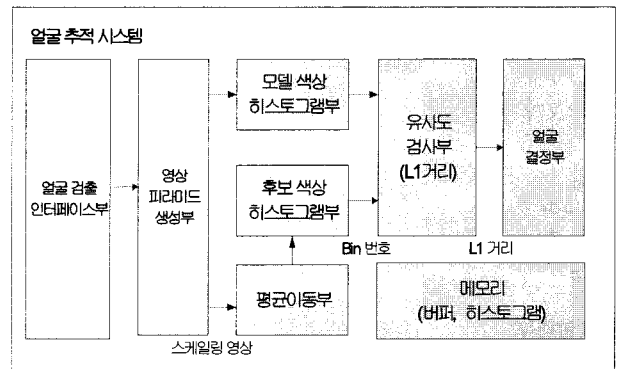


그림 4. 기본 얼굴 추적 시스템 구조
Fig. 4. Baseline Face Tracking System Architecture.

되었다.

추적 가능 얼굴의 수는 성능 및 하드웨어 크기에 직접적인 영향을 미친다. 응용 프로그램은 화면 내에 있는 모든 얼굴을 추적하는 것이 당연하지만, 추적 가능한 얼굴 수가 클수록 전체 추적속도는 느려지고, 하드웨어 면적은 커지게 된다. 따라서, 일상적인 디지털캠코더의 사용 환경을 고려하여 최대 10명으로 사양을 제한하였다. 추적 가능 얼굴의 크기는 너무 작을 경우 스케

일링을 여러 번 해야 해서 성능이 감소하고 너무 클 경우 얼굴 색상 히스토그램 메모리가 많이 필요하기 때문에, 실험을 통해서 47x47 픽셀로 크기를 정하였다.

본 논문에서 제안하는 얼굴추적 구조는 그림 4와 같다. 얼굴 검출 인터페이스부는 얼굴 검출 부분에서 넘어오는 얼굴 검출 좌표를 받아들이는 부분이다. 최대 10명까지의 결과를 받아 들여, 그 좌표를 기준으로 얼굴 추적을 실시한다. 영상 피라미드 생성부는 크게 두 가지 기능을 한다. 첫째, 센서로부터 들어오는 입력 영상을 메모리 크기와 얼굴 정보를 잃지 않는 범위 내에서 휴대용 멀티미디어 크기에 비례해서 작은 크기로 스케일링하고, 둘째, 얼굴 크기에 따른 얼굴 추적을 용이하게 하기 위해, 입력 영상을 3가지 피라미드 이미지로 스케일링 하는 역할을 하게 된다. 피라미드 이미지를 이용함으로써, 전체적인 픽셀 연산수가 줄어드는 효과를 볼 수 있다. 모델 색상 히스토그램부는 얼굴 검출부에서 넘어온 얼굴 좌표로부터 색상 히스토그램을 생성하는 부분이다. 모델 히스토그램은 얼굴 추적의 기본이 되는 기준 히스토그램이다. 후보 색상 히스토그램부는 얼굴 추적을 위해 일정 부분 커진 탐색창의 색상 히스토그램으로 기준 히스토그램인 모델 색상 히스토그램과의 비교를 위한 후보 색상 히스토그램이다. 평균 이동부는 움직인 위치를 찾기 위한 커널 기반 객체 이 구현된 부분이다. 유사도 검사부는 식 (4)의 모델 색상 히스토그램과 후보 색상 히스토그램간의 L1 거리를 구하는 부분이다. 얼굴 결정부는 유사도 검사부에서 구한 L1 거리를 기반으로 후보의 얼굴이 진짜 모델의 얼굴과 유사한 얼굴인지 판단하는 부분이다. 내부 메모리는 10명의 얼굴을 추적하기 위한 모델과 후보 각각 10개의 히스토그램 메모리와 이미지 스케일링을 위한 2-라인의 버퍼 메모리를 포함한다.

1. 메모리 연산 제어 및 절감

유사도 검사부는 모델 히스토그램과 후보 히스토그램 간의 L1 거리를 계산하게 되는데, 히스토그램은 256개의 bin을 가지고 있으므로, 두 메모리에 모두 256x2번의 메모리 액세스가 일어나야 한다. 하지만, 얼굴 검출 모듈에 의해 생성된 얼굴을 히스토그램으로 만들어 보면 특정 bin에 얼굴색이 분포되어 있는 것을 알 수 있다.

그림 5는 얼굴 모델 히스토그램과 얼굴 후보 히스토그램의 분포를 보여준다. [A]는 얼굴 검출된 프레임으

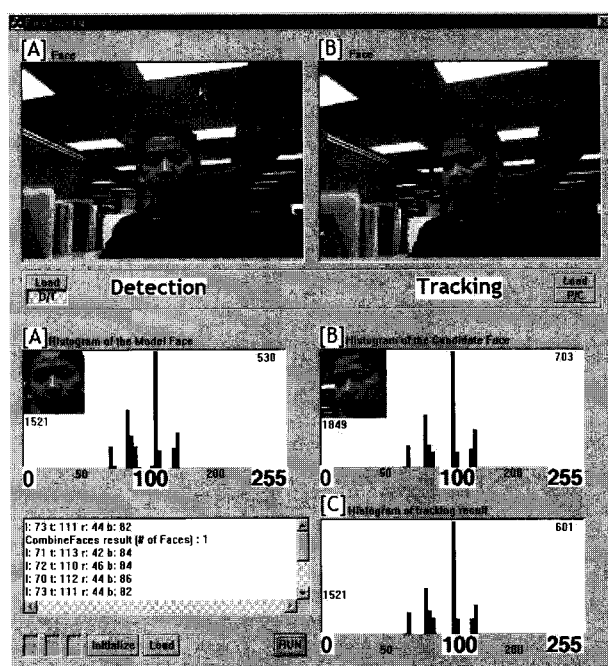


그림 5. 얼굴 색상 히스토그램
Fig. 5. Face Color Histogram.

로 색상 모델 히스토그램을 나타내고, [B]는 얼굴을 추적하기 위한 색상 모델 히스토그램, [C]는 중심 이동된 최종 추적 좌표의 색상 히스토그램을 보여준다. 전체 256-bin 중에서 얼굴 색상이 분포하는 히스토그램의 높이가 0이 아닌 색상 bin은 전체의 1/10 보다 작음을 볼 수 있다. 이것은 얼굴 검출기에서 이미 검출된 얼굴 좌표가 존재하기 때문에 얼굴을 제외한 배경색의 간섭을 최소한으로 줄일 수 있기 때문에 가능하다. 색상 bin이 없는 부분은 L1 거리를 구할 필요가 없으므로, 색상 bin이 있는 부분만 메모리를 액세스 한다면 많은 전력이 소요되는 메모리 액세스 수를 줄일 수 있다.

본 논문에서는 색상 bin이 존재하는 부분의 정보를 얻기 위해, 색상 히스토그램 생성부에서 색상 히스토그램 생성 시에 색상의 범위를 8개 부분으로 나누어 존재하는 Cb, Cr 각각의 색상 정보를 8-비트 플래그로 생성해서 유사도 검사부로 전달하게 하였다. 그림 6에서 8-비트 플래그는 4-비트인 Cb, Cr 값의 상위 2-비트(Cb[3:2],Cr[3:2])를 모니터링하여, 한번이라도 값이 존재하면 'high'로 세팅되도록 만들었다. 유사도 검사부는 색상 히스토그램으로부터 Cb, Cr 각각 4-비트의 색상 플래그를 받아 히스토그램 메모리 제어부에서 발생하는 메모리 주소와 OEN(read enable)과 제어신호를 멀티플렉싱하여 필요 없는 메모리 액세스를 막는다. 이 방법을 적용하면, L1 거리 계산 시 약 75% 정도의 메

모리 액세스 감소를 볼 수 있어 저전력 및 추적 시간 감소에 상당히 유리하지만, 필요한 하드웨어는 8-비트 레지스터, 2-입력 멀티플렉서 등 전체 면적의 0.2%에 불과하다.

얼굴 색상 히스토그램을 생성하기 위해서는 10명의

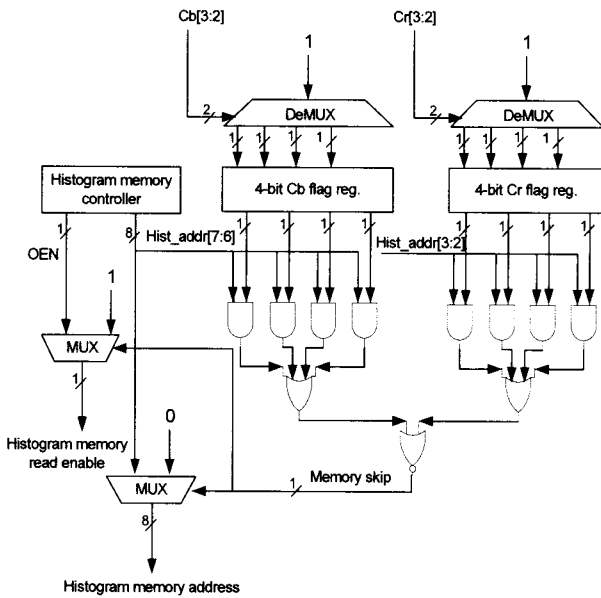


그림 6. 메모리 연산 제어 블록 다이어그램
Fig. 6. Block diagram for memory operation control.

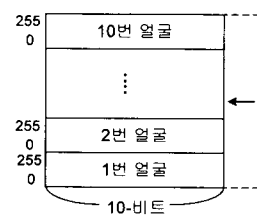
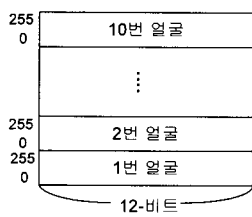
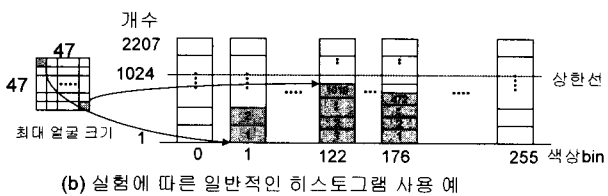
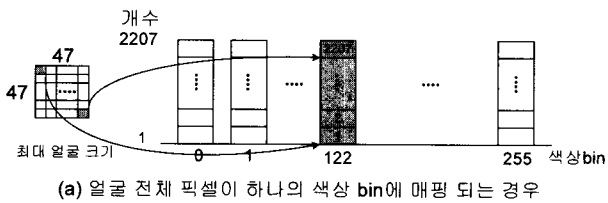


그림 7. 메모리 절감 효과
Fig. 7. Memory Reduction Effect.

모델과 후보 각각에 대한 메모리가 필요하게 되어 전체 얼굴 추적 시스템 면적에 큰 부분을 차지한다. 이론적으로 계산하면, 한 명에 대한 색상 히스토그램을 위한 메모리 크기는 다음과 같다.

47 x 47 (= 2209)개의 픽셀에 대한 정보를 저장하려면 12-비트가 필요하므로, 10명의 얼굴을 256 bin으로 표현하기 위해서는 그림 7의 (c)처럼 메모리 구조를 가지게 된다. 이 같은 경우는 그림 7의 (a)와 같이 사람 얼굴색이 하나의 얼굴색만으로 모두 이루어져서 256 색상 bin중 하나의 bin만으로 사람의 얼굴 특징을 나타낸 것이므로 현실적이지 않을 뿐만 아니라, 얼굴 검출도 불가능한 상태이다. 따라서 실험을 통해 그림 7의 (b)에서 보듯이 최소 5가지 이상의 bin으로 사람의 얼굴 특징을 기술해야 함을 알 수 있었다.

실험을 통해 얻어진 결과를 적용하면, 그림 7의 (d)의 메모리 구조를 가지게 되어 약 18% 정도의 히스토그램 메모리 면적 감소 효과를 볼 수 있고, 얼굴 추적 시스템 전체 면적 대비 2.8%의 면적이 감소한다.

■ 기존의 필요 메모리 양:

$$256\text{-bin} \times 12\text{-비트} \times 10\text{명} \times 2\text{-set} = 7.5\text{KB}$$

■ 부가 메모리 제거 시 필요 메모리 양:

$$256\text{-bin} \times 10\text{-비트} \times 10\text{명} \times 2\text{-set} = 6.2\text{KB}$$

2. 색상 히스토그램 생성의 파이프라이닝

색상 히스토그램 생성부는 모델과 후보의 색상 히스토그램을 생성하는 모듈로 메모리 액세스가 순차적으로 일어나게 된다. 추적 가능한 최대 얼굴의 크기가 47x47이므로 한 명의 얼굴 색상 히스토그램을 만드는 최대 필요 사이클은 47x47(memory read cycles 수)가 된다. 따라서 색상 히스토그램을 만드는 시간이 전체 추적 시간의 12% 정도 차지하게 되므로 성능 향상을 위해 파

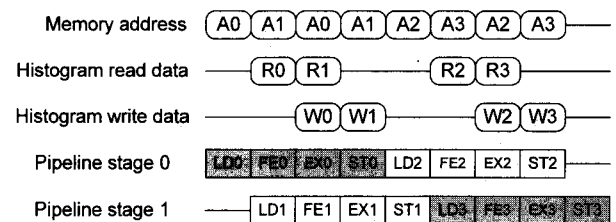


그림 8. 색상 히스토그램 파이프라이닝
Fig. 8. Color Histogram Pipelining.

이프라이닝을 적용하였다.

그림 8은 색상 히스토그램을 만들기 위한 파이프라이닝을 나타낸다. 새로운 색상 bin을 만들기 위해 Cb, Cr 메모리로부터 픽셀 데이터를 읽고, 식 (3)처럼 색상 히스토그램 주소를 생성한다. 생성된 히스토그램 주소에 대해 히스토그램 메모리를 읽어서 그 값에 1을 더한 뒤, 다시 쓰기를 하면 명령이 끝나게 된다. 이 때, 주의할 점은 첫 번째 주소에 이어지는 주소가 동일할 경우, 첫 번째 주소에 대한 쓰기 데이터와 두 번째 주소에 대한 읽기 데이터의 데이터 의존성(data dependency)이 발생하므로, 이를 회피하기 위해 같은 주소에 대해서는 휴지 사이클을 추가하고 다음 사이클에 2를 더해주는 전략을 취했다.

파이프라이닝을 적용하기 전에는 하나의 픽셀에 대한 색상 히스토그램을 생성하는데 3-사이클이 필요했지만, 적용 후에는 두 픽셀에 대해 4-사이클이 소요되어 이론적으로는 50%의 성능 향상이 가능하다. 하지만, 하나의 수평 라인이 끝나게 되면 다시 메모리 주소를 계산해야 할 뿐만 아니라, 전체적으로 기본적으로 필요한 천이상대 사이클이 필요하게 되어 결과적으로는, 한 명의 색상 히스토그램을 생성하는 시간이 적용 이전 대비 평균 23% 정도 감소하였다. 반면, 파이프라이닝을 구현하기 위해 로직이 추가적으로 필요해서 면적 측면에서는 trade-off가 존재했으나, 전체 얼굴 추적 시간에서 컬러 히스토그램을 생성하는 시간이 큰 부분을 차지하므로 파이프라이닝 적용을 선택하게 됐다.

3. 내부 스케일러의 최적화

본 시스템의 영상 피라미드부에서는 얼굴 추적 시 색상 히스토그램 생성 시 필요한 메모리 요구량과 메모리 연산량을 줄이기 위해, 얼굴 크기에 따른 영상 계층피라미드를 구성하는데, 원영상으로부터 수평과 수직으로 각각 2:1로 스케일링을 하여 먼저 180x120 크기의 원영상을 90x60 크기의 영상으로 구성한 후, 마지막으로 45x30 크기의 영상 계층 피라미드를 생성한다. 각 영상 피라미드마다 추적 가능한 얼굴의 크기를 제한해서 색상 히스토그램이 최대 47x47 픽셀이 넘는 것을 제한한다. 따라서 얼굴의 크기가 95 픽셀 이상인 얼굴은 두 번의 스케일링을 거쳐 3단계의 피라미드 계층에서야 추적이 가능해 지고, 얼굴 크기가 47보다는 크고 95보다 작은 경우는 한 번의 스케일링이 필요하며, 얼굴 크기가 47 픽셀보다 작은 얼굴은 원 영상에서 스케일링 없이

표 1. 계층 피라미드 영상의 추적 얼굴 크기

Table 1. Face Size of Tracking per Hierarchical Image Pyramid.

피라미드 계층	피라미드 크기	추적 얼굴 크기 (Width)
1 단계	180x144(PAL) 180x120(NTSC)	Width<=47
2 단계	90x72(PAL) 90x60(NTSC)	47<Width<=95
3 단계	45x36(PAL) 45x30(NTSC)	95<Width

바로 추적이 가능하다. 각 피라미드 단계에서 추적하는 얼굴 크기는 표 1과 같다.

스케일러는 각 영상 피라미드의 제한된 범위에서 다양한 크기의 얼굴을 추적하기 위해 필요하다. 하지만, 얼굴 추적 시스템 내부에서 스케일된 영상은 외부 디스플레이용으로 사용되지 않을 뿐만 아니라, 식 (3)에서와 같이 색상 히스토그램 생성을 위해 검출된 얼굴 좌표내의 간략화되고, 유사한 얼굴 색상 데이터만 필요하므로 B-Splines과 같은 하드웨어 사이즈가 크고, 복잡한 알고리즘이 필요하지 않았다. 입력 영상을 B-Splines 알고리즘과 bi-linear 알고리즘을 각각 사용하여 다운 스케일링을 하고 얼굴 색상 히스토그램을 생성하는 실험을 한 결과, 두 히스토그램 간에 0.972의 강한 양의 상관도와 0.21 정도의 L1 거리차만 보여 주어 전체적인 추적 성능에 큰 차이를 보이지 않았다.

따라서 본 시스템에서는 bi-linear 타입의 스케일러를 특징화된 영상 피라미드 크기 및 스케일링 비율에 맞추어 수정하였다. 우선, 스케일링 비율이 1/2로 고정되어 있으므로 각 픽셀에 대한 가중치가 필요 없기 때문에 가중치 처리를 위한 곱셈기 및 뺄셈기의 제거가 가능하였고, 스케일링 비율이 1/2보다 작은 경우에 화소가 필터를 거치지 않아 발생할 수 있는 엘리어싱을 막기 위한 프리스케일러 (pres 맞추케일)의 구현도 필요가 없어져, 덧셈기와 쉬프트 연산만으로 로직 구현이 가능하였다. 또한, 얼굴 추적 시스템 내부에 캐시RAM 타입의 프레임 버퍼를 사용함으로써, 수직 방향으로의 스케일링 시 필요한 라인 버퍼도 스케일러 내부에서 제거하였다.

이와 같이 본 얼굴 추적 시스템에 적합한 최적화 작업을 통해 범용 bi-linear 타입의 스케일러 대비 76% 면적이 감소하였다.

IV. 실험 결과

1. FPGA 검증

구현된 얼굴 추적 시스템을 검증하기 위해 그림 11과 같은 검증 환경을 사용하였다. FPGA는 Xilinx사의 Virtex-4를 사용하였고, Trace32를 통해 ARM7을 제어하여 얼굴 검출기 및 추적기를 동작시켰다. 입력 영상 데이터는 검증 보드 안에 내장되어 있는 DRAM 메모리칩에 테스트 영상을 미리 저장하였다.

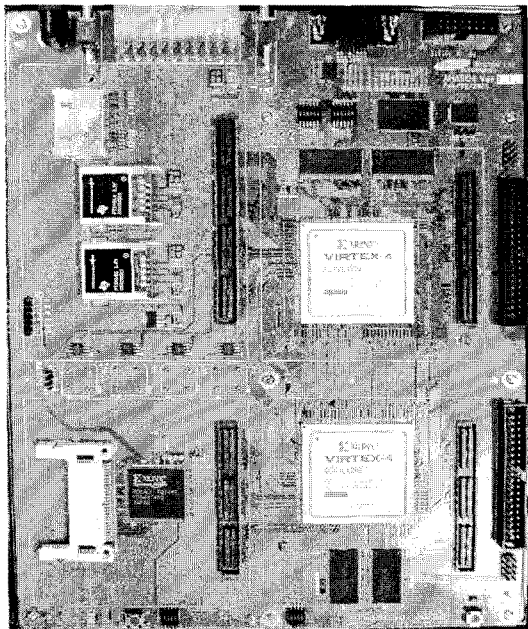


그림 9. FPGA 프로토타이핑 보드
Fig. 9. FPGA Prototyping Board.

2. 합성 결과

얼굴 검출 시스템은 Verilog HDL로 기술되었고, 삼성 65nm CMOS 라이브러리로 합성하였다. 표 2는 전체 얼굴 검출 시스템에 대한 결과 나타낸다. 최대 클럭 주파수는 210MHz이다. 일반적인 카메라 시스템이 133MHz 정도로 동작한다고 봤을 때, 부가적인 기능 처리를 위한 성능 마진을 고려해서 210MHz를 사용했다.

표 2. 얼굴 추적 시스템 합성 결과
Table 2. Face Tracking System Synthesis Result.

얼굴 추적 시스템 합성 결과	
클럭 주파수(MHz)	210
전체 면적(um ²)	860832
로직 게이트(Kgates)	157
메모리(KBytes)	57

전체 메모리 사이즈는 57KB이고, 이 중 50KB는 PAL 기준 180x144의 얼굴 추적기 입력 영상을 저장하기 위한 프레임 버퍼이다. 프레임 버퍼 메모리가 전체 칩 사이즈의 55%를 차지하고 있지만, 칩 사이즈를 줄이기 위해 외부 SDRAM 메모리로 프레임 버퍼를 옮길 경우, 외부 메모리 이용으로 인한 전력소모 증가 및 시스템 버스의 대역폭 공유로 발생하는 동작 성능 감소 등의 trade-off가 발생할 수 있어 내장 메모리를 사용하였다.

3. 메모리 전력 예측

메모리 연산 제어를 통한 메모리 소모 전력을 예측하기 위해 전력 예측기^[7]를 사용하였다. 시뮬레이션은 게이트 레벨 worst-case에서 수행되었다.

표 3은 저전력을 위한 메모리 연산 제어 방법을 적용하기 전과 적용 후의 전력을 비교하여 나타낸다. 시뮬레이션 결과는 메모리 연산 제어 방법을 적용하기 전보다 약 70% 정도의 전력 감소 효과가 있었다. 이것은 얼굴 색상 히스토그램에 존재하는 색상 bin이 III장 1절에서 예측한 것처럼 25%이하임을 나타낸다.

표 3. 메모리 소모 전력 예측
Table 3. Memory Power Estimation.

히스토그램 메모리 소모 전력 예측		
Baseline(적용 전)	1342uW	100%
메모리 연산 제어 적용 후	414uW	30%

4. 동작 성능

본 논문에서는 디지털 캠코더의 응용 프로그램에 맞추어 하드웨어의 성능, 크기 등을 고려하였다. 표 4는 설계된 추적기의 성능을 나타내었다. 얼굴 추적기 단독으로만 동작시켰을 때, 하나의 얼굴을 찾는 데 최대 1.9 msec가 필요하여, 10명의 얼굴을 모두 찾을 경우, 약 8.4 msec만 소요되므로, 휴대용 캠코더가 동작하는 30

표 4. 성능 결과
Table 4. Performance results.

성능 결과	
입력 영상 크기	180x144(PAL) 180x120(NTSC)
최대 추적 가능 얼굴 수	10명
최대 추적 가능 얼굴 크기	47x47 픽셀
최대 추적 시간(10명)	8.4msec



그림 10. 다측면 및 다중 얼굴 추적 결과
Fig. 10. Results of Multi-view and Multiple Face Tracking.

fps (프레임당 33.3 msec) 스펙 내에서 마진 있게 동작할 수 있게 된다. 이것은 얼굴 추적된 결과를 사용하는 응용 프로그램 뿐만 아니라, 같은 프레임 내에서 다른 어플리케이션도 구동 가능하게 함으로써, 시스템에게 대역폭에 대한 부담을 줄여준다.

그림 10은 다측면 얼굴의 다양한 추적 결과를 나타낸다. 각 실험은 초기 프레임에 얼굴 검출이 동작하여 초기 얼굴의 좌표를 획득한 뒤, 연속되는 후속 프레임들은 얼굴 추적 시스템이 동작하는 본 논문의 알고리즘을 적용시켰다. 실험 결과에 의하면, 정면 얼굴뿐만 아니라 그림 10의 (a), (b), (c), (d), (e)와 같이 다측면 및 다중의 얼굴이 실시간으로 추적이 가능했다.

그림 11은 급격한 조명 변화에 대한 얼굴 추적 시스템의 추적 성능을 보여준다. 그림 11의 (a)는 얼굴 추적기 없이 얼굴 검출기만 동작했을 경우에, 환경이 급격히 저조도로 변화했을 때, 얼굴 검출기가 명암 정보의 부족으로 얼굴 검출에 실패했을 경우를 나타낸다. 그림 11의 (b)는 얼굴 검출기와 얼굴 추적기가 동시에 동작하여, 저조도의 환경 변화에 대해 얼굴 검출의 한계를 보완하여, 색상 정보를 이용해서 얼굴이 추적 가능함을



(a) 급격히 조명이 어두워졌을 때, 얼굴 검출만 동작 시 (false negative)



(b) 급격히 조명이 어두워 졌을 때, 얼굴 검출과 얼굴 추적이 동시에 이루어 졌을 경우(false negative 제거)

그림 11. False negative 제거 효과
Fig. 11. False negative removal effect.

보여준다. 이 같은 결과는 본 시스템이 명암 정보가 부족한 급격한 조도의 변화에서도 false negative rate을 개선해 주어, AdaBoost 얼굴 검출 알고리즘을 보완해 조명 변화에 강인함을 보여준다.

V. 결 론

본 논문에서는 하드웨어 기반의 실시간으로 동작하는 저전력 얼굴 추적 시스템을 제안하였다. 제안하는 하드웨어는 얼굴 검출기와 함께 동작해서 실시간으로 다측면 얼굴 추적이 가능하고 조명 변화에 강인함을 보였으며, 저전력으로 구현되어 휴대용 멀티미디어 기기 적용에 적합함을 보였다. 또한, 매 프레임마다 얼굴 추적 정보 제공이 가능하여 얼굴 기반 응용 소프트웨어에 대해 성능 향상 및 확장성을 제공할 수 있었다.

참 고 문 헌

- [1] Viola and Jones, Robust real-time object detection, International Journal of Computer Vision, Volume 57, issue 2, pp. 137-154, May 2004.
- [2] Dorin Comaniciu, Kernel-Based Object Tracking, IEEE Transactions on pattern analysis and machine intelligence, Volume 25, issue 5, pp.

- 564-577, MAY 2003.
- [3] Chang Huang, Vector boosting for rotation invariant multi-view face detection, The Tenth IEEE International Conference on Computer Vision (ICCV'05), Volume 1, pp. 446-453, 17-21 Oct. 2005.
- [4] Wing-pong Choi, Kini-Man Lam, An Effective Shape-Texture Weighted Algorithm for Multi-view Face Tracking in Videos, Congress on Image and Signal Processing (CISP'08), Volume 4, pp. 156-160, May 2008.
- [5] Kim jung-bae, 비디오 영상 트래킹 방법 및 장치, 대한민국특허청, 10-200700011122, Mar, 2008.
- [6] 이수현, 얼굴 검출을 위한 SoC 하드웨어 구현 및 검증, 전자공학회 논문지 제 44권 SD편 제4호, pp. 43-54, April 2007.
- [7] http://www.sequencedesign.com/image/files/powertheater_datasheet.pdf

 저 자 소 개



윤 석 기(학생회원)
 2000년 2월 충남대학교 정보통신공학과 학사 졸업.
 2000년 2월~현재 삼성전자 책임연구원.
 2008년 3월~현재 성균관대학교 반도체디스플레이공학과 석사과정.

<주관심분야 : SoC설계, 저전력, 영상처리>



한 태 희(평생회원)-교신저자
 1992년 KAIST 전기 및 전자공학과 학사 졸업.
 1994년 KAIST 전기 및 전자공학과 석사 졸업.
 1999년 KAIST 전기 및 전자공학과 박사 졸업.

1999년 3월~2006년 8월 삼성전자 통신연구소 책임 연구원.

2006년 9월~2008년 2월 한국산업기술대학교 전자공학과 조교수.

2008년 3월~현재 성균관대학교 정보통신공학부 반도체시스템공학 전공 조교수.

<주관심분야 : IT SoC 설계, VLSI 설계 기술 및 방법론, 단말 시스템>