

멀티코어 SoC에서 피부색상 정보와 병렬처리를 이용한 효율적인 얼굴 검출 방법

An Efficient Face Detection Method using Skin Color Information and Parallel Processing in Multi-Core SoC

김 홍 희*, 이 재 흥**

Hong-Hee Kim*, Jae-Heung Lee**

Abstract

In this paper, we present an implementation of Viola-Jones algorithm in a multi-core SoC by using skin color information and a parallel processing method. In order to reduce unnecessary operations and improve the detection speed, we adopted a face detection algorithm based on skin color and deleted background image. The algorithm is functionally divided into several parts taking account of the size and the dependency so that the divided functions can be proceeded in parallel. Experiment results in SoC with built-in Cortex-A9 multi core show that it is about 1.8 times faster than the existing algorithm which is not divided.

요 약

본 연구에서는 멀티코어에서 피부색 정보와 병렬처리를 이용하여 Viola-Jones 알고리즘을 설계하였다. 불필요한 연산을 줄이고 빠른 검출 속도를 위해 기존의 얼굴 검출 방법 중 피부색을 이용한 얼굴 검출 알고리즘을 사용하여 배경영상을 삭제하였다. 병렬처리를 위해 알고리즘을 기능별로 분할하였고, 멀티코어에서 성능을 향상시키기 위해 의존성과 크기를 고려하여 독립적으로 동작되도록 설계하였다. Cortex-A9 멀티코어가 내장된 SoC에서 실험 결과 알고리즘을 분할하기 전에 비해 약 1.8배 빠른 검출 결과를 나타내었다.

Key words : Embedded System, Face Detection, Multi-Core, Parallel processing, Haar-like feature

1. 서론

* Dept. of Computer Engineering, Hanbat University (freemoora@naver.com)

Corresponding author

※ Acknowledgment

This research was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation

Manuscript received Dec. 9, 2012; revised Dec. 17, 2012; accepted Dec 17, 2012

최근 각종범죄가 사건사고 뉴스에 끊이지 않고, 범죄 예방에 관련된 인식이 사회 전반적으로 높아지면서 범죄를 예방하기 위한 방편으로 영상 저장 장치에 대한 수요가 증가하고 있다. 영상 저장 장치는 단순히 범죄를 예방하는 것으로만 그치지 않고 실제로 촬영된 영상 기록을 기반으로 범죄 해결하는 경우도 빈번하다. 영상 저장 장치는 CCTV에서 DVR, Network Camera 등으로 변천해가고 카메라 또한 지능화되고 있어 일상생활 속에서 감시 기능도 하고 있다. 현재 생활 곳곳에서 볼 수 있는 번호판 인식 카메라, 지문

인식 카메라 등과 같은 고성능 인식 알고리즘이 탑재된 시스템의 경우 PC를 활용하여 처리하는 경우가 많아 가격이 비싸고 설치범위가 제한적이다.

생체 인식 중 얼굴 인식은 다른 지문인식 또는 홍채 인식 등과 다르게 사용자가 직접 인식 장치에 접촉하지 않고 카메라 스스로 얼굴을 검출하여 인식하므로 사용자에게 편리함을 제공한다. 이러한 이유로 얼굴 인식 기술은 다른 인식 기술에 비해 주목을 받고 있다. 얼굴 검출과 얼굴 인식은 서로 다른 분야이다. 얼굴 검출은 얼굴 인식의 전 단계라 할 수 있다. 예를 들어 영상 장치로부터 입력받은 데이터를 분석하여 사람의 얼굴과 기타 유사 객체를 구분하는 알고리즘은 얼굴 알고리즘이라 할 수 있다. 얼굴 검출을 통해 얼굴 데이터를 배경으로부터 분리하고 이목구비(耳目口鼻)를 기반으로 특정 인물을 찾아낸다면 얼굴 인식이 된다. 성공률이 높은 얼굴 인식을 하기 위해서는 얼굴 검출이 중요한 전처리과정이지만 현재는 그 중요성에 따라 얼굴 검출이 얼굴 인식과 독립적인 분야로 연구되고 있다.

이와 더불어 최근 ARM 프로세서를 기반으로 임베디드 모듈의 성능이 향상되고 다양해짐에 따라 임베디드 기반의 영상처리 알고리즘에 대한 연구가 활발히 진행되고 있다. FPGA(Field-Programmable Gate Array)를 기반으로 얼굴검출 전용 알고리즘 하드웨어 설계가 가장 대표적인 방법이다[1]. ARM 프로세서는 최근 싱글코어 기반에서 멀티코어로 발전해가고 있다. 싱글코어에서 클럭을 높이는 것보다 멀티코어로 발전함이 다양한 확장성과 시스템 성능향상, 전력 소비의 감소 등의 효과를 얻을 수 있다[2]. 이러한 장점을 기반으로 본 연구에서는 임베디드 시스템에서의 효율적인 얼굴 검출을 목적으로 하고 있다. 멀티코어가 내장된 임베디드 하드웨어 플랫폼을 이해하고 얼굴 검출을 구성하는 알고리즘을 스레드를 활용하여 병렬 처리하였다. 기존의 Viola-Jones 알고리즘을 멀티코어에 맞게 변형하여 성능을 개선하였으며 그 결과를 ARM Cortex-A9 MPCore에서 측정하였다.

관련연구에서는 기존의 얼굴 검출 연구를 알아보고 본 연구에 활용되는 Viola-Jones 알고리즘[3]에 대해 설명하였다. 또한 3절에서는 효율적인 병렬처리를 위해 ARM Cortex-A9 MPCore를 분석하였다. 3장의 제안하는 방법 1절에서는 멀티코어에 맞는 병렬처리 방법에 대해 기술하였고 2절에서는 피부색을 통한 얼굴 후보 추출과 POSIX Thread를 활용하여 얼굴 검출 알고리즘을 병렬 처리하는 방법에 대해 설명한다. 그리고 3장에서는 멀티코어가 탑재된 H/W에서 테스트한 결과를 수치로 표현하였으며 그 결과를 종합하

여 결론을 맺는다.

II. 관련연구

1. 기존의 얼굴 검출 연구

기존의 얼굴검출 방법은 크게 지식 기반 방법(Knowledge-Based Top-Down Methods), 특징 기반 방법(Bottom-Up Feature-Based Methods), 형판 정합 방법(Template Matching), 외형 기반 방법(Appearance-Based Methods)으로 나눌 수 있다[4]. 지식 기반 방법은 사람의 이목구비(耳目口鼻)와 같이 얼굴의 형태를 구성하는 특성들을 기하학적인 방법으로 검출하는 방법이다. 특징 기반의 방법의 가장 대표적인 방법은 피부색을 이용하는 방법이다. RGB, YCbCr, HSI 등과 같은 영상 데이터 포맷을 기반으로 픽셀 데이터 값의 일정범위를 정하여 탐색을 하게 된다. 피부색을 이용하여 얼굴을 검출하는 방법은 고전적인 방법으로 검출 속도가 빠르지만 조명과 주변 환경, 피부색 등으로 인해 오류가 발생할 가능성이 크다.



Fig 1. Face Candidate Detector using Skin Color
그림 1. 피부색을 이용한 얼굴 후보 검출

형판 정합 방법은 미리 정해진 형판을 입력 영상과 비교하여 검출하는 방법이다. 고정된 영상에서는 성공률이 높지만 입력 형태가 달라지면 성공률이 낮아진다. 마지막으로 외형 기반 방법은 신경망, Support Vector Machine, HMM 등과 같이 많은 영상을 입력받아 학습한 후 학습된 데이터와 입력 영상을 비교하여 검출하는 알고리즘이다. 학습된 데이터의 양과 질에 따라 인식률이 달라지고 계산량이 많아 속도가 느리지만 다른 방법들이 비해 신뢰성이 높고 검출 성공률 또한 높다.

2. Viola-Jones 알고리즘

Viola-Jones 알고리즘은 특징 기반 방법에 속하고 현재 널리 사용되고 있는 얼굴검출 알고리즘이다. 다른 물체 검출 방법들과는 달리 물체가 위치할 범위를

초기화하는 작업이 필요 없으며 높은 얼굴 검출율과 동시에 매우 빠른 이미지 프로세싱이 가능한 얼굴 검출 프레임 워크 기술이다[3].

Viola-Jones 알고리즘은 크게 3가지 방법을 제시하고 있다. 첫 번째는 검출기를 매우 빠르게 계산되도록 해주는 적분이미지(Integral Image), 두 번째는 간단하고 효율적인 AdaBoost 학습 알고리즘을 사용하여 만든 분류기이다. 이 알고리즘은 잠재적인 얼굴 후보 중에서 강력한 외모의 특징을 나타내는 영역을 선택한다. 세 번째는 "Cascade" 분류기의 결합의 방법이다. Cascade는 많은 약분류기를 단계별로 구분하여 빠르게 얼굴을 검출한다[2]. 위 방법을 토대로 Viola-Jones 알고리즘은 실시간으로 물체를 검출할 수 있게 하였으며 Adaboost 알고리즘은 하나의 특징만을 이용하여 단일 약분류기 여러 개를 중요도에 따라 가중치를 두어 조합한 강분류기를 만들어냄으로써 Viola-Jones 알고리즘이 물체를 더욱 정확하게 검출할 수 있도록 하였다.

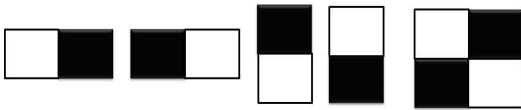


Fig 2. Example of Haar-like feature
그림 2. Haar-like feature 예시

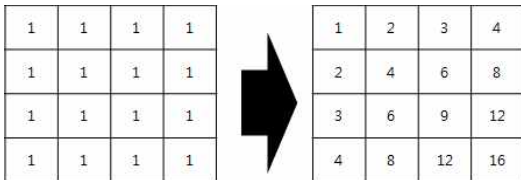


Fig 3. Convert Original image to Integral image
그림 3. 원본 이미지를 적분 이미지로 변환

3. ARM Cortex-A9 MPCore

Cortex-A9 MPCore 멀티코어 프로세서는 효율적인 전력 관리 기술과 높은 성능을 지원하며, 확장성이 뛰어나 현재 모바일 기기에서 많이 사용되는 최신 프로세서이다.

Cortex-A9 MPCore는 1-4개의 프로세서를 지원하며 각각의 프로세서는 1GHz 클럭 주파수 이상이 가능하다. 멀티코어와 SCU(Snoop Control Unit)를 사용하면 멀티코어 애플리케이션 프로그래밍에 매우 효과적이다. 또한 NEON 미디어 처리 엔진을 통해 애플리케이션 성능을 개선하고 소프트웨어 개발의 편리성을 제공한다. 얼굴검출 알고리즘을 설계하다보면 부동소

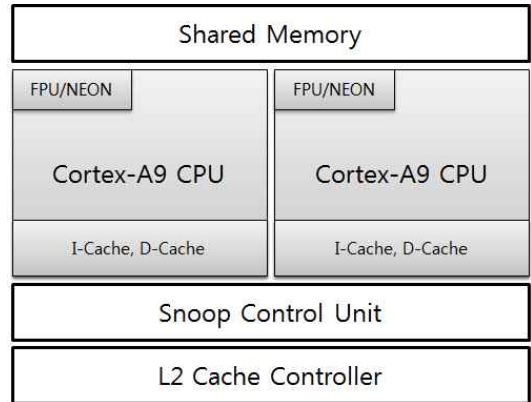


Fig 4. Cortex-A9 Dual-Core Block Diagram
그림 4. Cortex-A9 Dual-Core 블록 다이어그램

수점 연산은 필연적으로 사용할 수밖에 없다.

Cortex-A9는 이전 ARM 프로세서의 FPU(Floating Point Unit)보다 성능이 두 배 개선되었다[5]. 저비용과 효율적인 성능을 요구하는 애플리케이션의 예는 무수히 많다. 멀티코어 프로세서 아키텍처를 사용하는 것은 최대 성능 수요를 충족하면서도 전력 소비가 매우 적은 디자인을 유지할 수 있는 한 방법이다. 멀티코어 기기는 확장성이 매우 우수한 성능과 저전력 애플리케이션을 구현하여 디자인상의 유연성을 높은 수준으로 제공할 수 있다[5][6].

III. 제안하는 방법

1. 멀티코어 프로그래밍 방법

병렬처리에 관한 연구는 오래전부터 지속되어 왔으며 현재 많은 방법들과 라이브러리들이 사용되고 있다. OpenMP[10] 라이브러리는 Windows 환경에서는 아주 손쉽게 병렬처리를 가능하게 한다. OpenMP는 리눅스 환경에서도 사용이 가능하지만 아직까진 ARM 프로세서에서는 완전히 적용되어 사용되지 않는다. ARM 프로세서와 같은 임베디드 플랫폼에서 가장 많이 사용되는 병렬처리 방법은 POSIX Thread를 사용하여 병렬 처리를 하는 것이다.

본 연구에서는 효율적인 병렬처리 방법을 위해 그

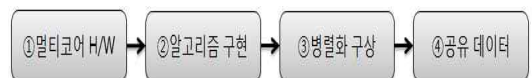


Fig 5. Parallel processing Flow
그림 5. 병렬처리 흐름도

림 5과 같이 4단계로 나누어 시스템을 설계하였다.

(1) 멀티코어 H/W : 기존의 싱글코어에서 멀티 스레드를 활용하여 설계하면 일정 시간마다 소프트웨어 스레드를 번갈아 가며 실행하는 방법을 사용하였다. 처리 결과만 보면 병렬처리라 할 수 있지만 그 과정은 완벽한 병렬처리라 할 수 없다. 멀티코어에서 다중 스레드를 활용하여 프로그래밍을 하면 완벽한 병렬처리가 가능해진다. 멀티코어가 내장된 H/W를 확실히 이해하고 그에 맞는 프로그래밍을 하여야 한다. 예를 들면 SIMD(Single Instruction Multiple Data)[11] 아키텍처를 지원하는 ARM 프로세서의 경우 그에 맞는 명령어를 사용함으로써 동작시간을 줄일 수 있고, 2장에 소개한 SCU와 같은 H/W 모듈을 확실히 이해하면 더욱 효율적인 병렬처리가 가능해진다.

(2) 알고리즘 구현 : 모든 알고리즘을 병렬처리 방법을 활용하여 구현한다고 속도가 개선되는 건 아니다. 우선 자신이 구현하고자하는 알고리즘을 일반적인 방법으로 구현 한다. 알고리즘을 구현할 때는 공통된 동작을 하는 모듈별로 나누어 설계함이 추후 스레드를 활용하여 병렬처리를 할 때 편리하다. 또한 병렬처리로 구현한 알고리즘이 일반적으로 구현한 알고리즘 비해 성능이 저하되는 경우도 있다.

(3) 병렬화 구성 : 모듈별로 구현된 알고리즘을 구현하면 다음단계로 병렬 처리 가능한 모듈을 선택해야 한다. 병렬처리를 하기 위해서는 그에 맞는 스레드 생성 함수가 필요하고 또한 결과 값을 도출하기 위한 함수도 새로 필요하다. 즉, 기존의 알고리즘에 비해 새롭게 추가되는 함수가 존재하게 되어 시간이 좀 더 소요된다. 이러한 속도적인 측면을 고려할 때 모든 모듈을 병렬 처리하는 것보다 모듈의 처리시간과 사용빈도 등을 고려하여 적합한 모듈만을 병렬 처리해야 효율적인 알고리즘을 구현할 수 있다.

(4) 공유 데이터 : 멀티프로그래밍 방법으로 알고리즘을 구현하였다면 병렬 처리되는 모듈간의 의존성을 확인해야 한다. 하나의 결과를 위해 동작하는 과정들은 다양한 모듈로 구성되어 있고 각각의 모듈은 병렬 처리로 병렬처리 되고 있다. 즉, 다른 코어에서 각각의 데이터를 처리하고 있고 처리되는 데이터는 다른 코어에 영향을 주어서는 안 된다. 또한 독립적인 동작이 이루어질 수 있도록 데이터 또한 독립성을 가져야한다. 만일 독립적이지 못한 데이터들이 존재하면 예상치 못한 결과를 도출할 것이고 병렬처리는 의미가 없어진다.

2. 병렬처리 방법을 이용한 얼굴 검출

멀티코어를 가진 Cortex-A9 MPCore는 기존의 단

일코어 방식과 동일하게 설계하면 효율적이지 못하다. 컴파일러 또는 SCU를 통한 CPU의 업무 분담은 한계가 있으며 그 성능 또한 미비하다. 본 연구에서는 멀티코어에 효과적인 방법으로 다음의 2가지 단계를 제안한다.

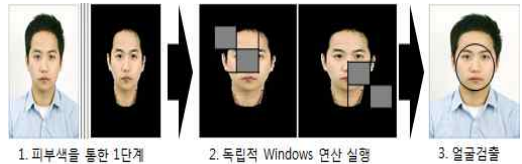


Fig 6. Face Detection parallel processing Flow
그림 6. 얼굴검출 병렬처리 흐름도

Viola-Jones 알고리즘의 경우는 얼굴의 위치와 상관없이 탐색을 하기 때문에 얼굴영역 이외의 공간을 탐색해야하는 번거로움이 존재한다. 이러한 과정은 얼굴 위치를 정해두지 않고 탐색하기 위해선 필수적인 단계로 모든 영역을 탐색하다보면 시간이 많이 소요된다. 이러한 문제점을 어느 정도 보완하기 위해 피부색을 통해 얼굴 영역을 검출한다. 입력되는 데이터를 YCbCr 형식으로 맞추고 그 중 Cb, Cr의 컬러영상 픽셀 값을 기준으로 피부색을 판단한다. 피부색을 이용하여 탐색을 하면 피부색과 유사한 객체에는 오류를 범하지만 얼굴 후보만을 검출하기 때문에 큰 문제가 되지 않는다. 또한 한 번의 탐색으로 얼굴 후보를 추출할 수 있기 때문에 시간적으로 많이 절약이 된다.

다중 멀티코어에서는 코어별로 최적의 자원을 분배하여 효율적으로 동작하게 하는 것이 중요하다. 그러기 위해서는 서로 독립적인 상태가 유지되어야 하며 결과 값이 다른 코어에 영향을 주어서는 안 된다. Viola-Jones 알고리즘의 경우 독립적으로 동작하는 부분이 많은 곳에 존재한다. 그 중 Windows 영역은 서로 독립적으로 동작하며 그 결과 값 또한 영향을 주지 않는다.

2.1 피부색을 통한 얼굴 후보검출

피부색을 이용하여 얼굴을 검출하는 방법은 고전적인 방법으로 얼굴과 비슷한 색이 배경 영상에 있으면 심각한 오류를 범하게 된다. 하지만 얼굴 후보를 추출함에는 효과적이고 계산과정이 단순하여 ARM 프로세서에 적합하다. 컬러 영상 포맷 중 YCbCr의 CbCr 영역만으로 후보 얼굴 영역을 검출할 수 있다 [8].

$$(x, y) = \begin{cases} 1 & \text{if } [100 \leq b(x, y) \leq 128] \\ & \text{and } [135 \leq Cr(x, y) \leq 164] \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

식(1)에 만족하는 Y영역의 범위만을 따로 저장하고 그 외의 범위에는 '0x00'으로 값을 채운다. 얼굴 후보 이외지역에 '0x00'를 채우게 되면 동일한 값에 대한 계산이 반복적으로 실행되므로 전체적인 연산이 빨라지고, 분류기가 동작할 때 확실한 결과 값을 전달하여 최대 65%까지 시스템 속도가 빨라진다.



Fig 7. Face Region Detection using Skin Color
그림 7. 피부색을 이용한 얼굴 영역 검출

2.2 Windows 병렬 처리

피부색을 통한 1단계가 거치면 2단계는 POSIX Thread를 활용하여 Windows연산을 독립적으로 병렬 처리한다. 입력 영상에서 얼굴이 크기와 위치는 항상 다르기 때문에 Windows를 설정하여 모든 영상 범위를 차례대로 탐색해야한다.

Viola-Jones의 전체 알고리즘 순서는 2.2절에서 언급한 것과 마찬가지로 다양한 Haar-like feature가 모여서 Adaboost 알고리즘을 구성한다. Adaboost 알고리즘을 통해 높은 검출 성공률을 얻기 위해서는 매우 많은 수의 약분류기를 사용해야 한다. 많은 약분류기를 사용하게 되면 속도가 문제되기 때문에 Stage를 직렬로 배치하고 단계별로 모두 만족해야 얼굴을 검출하도록 되어있다[2]. 이러한 과정은 서로 의존성이 강하기 때문에 병렬 처리를 할 수가 없다. 또한 약분류기를 병렬처리하기에는 너무 많은 Thread를 생성해야 하므로 시간을 더욱 증가시킨다. 얼굴 탐색과정은 픽셀 처음부터 끝까지 모든 영역에 거쳐 이루어져야한다. 이러한 동작은 서로 독립적이고 의존성이 없다. 이러한 사실에 기반을 두어 탐색의 범위를 지정하는 Windows를 병렬 처리하도록 설계하였다.

그림 8에서의 빨간색 테두리가 Windows이다. Windows의 위치는 데이터의 크기에 따라 개수가 달라지며 상단 좌측부터 하단 우측까지 움직이게 된다. 본 연구에서는 코어 수에 맞게 POSIX Thread를 생성하여 Windows 탐색과정을 병렬처리 하였다.



Fig 8. Example of Windows Position
그림 8. Windows 위치 예시

```
for all WZ // WZ: window size
Resize image;
Integral image;
for all WP // WP: window position
Detect {
for all SC // SC: strong classifier
for all WC // WC: weak classifier
If failed, label the position as negative; jump to the next WP;
If passed all the SCs, label the position as positive;
}
```

Fig 9. Viola-Jones Pseudo Code
그림 9. Viola-Jones Pseudo Code

```
Make Thread(CORE)
for all/core WZ // WZ: window size
Thread[0] = Viola-Jones Face Detection; // Detecting face
Thread[1] = Viola-Jones Face Detection; // Detecting face
}
```

Fig 10. Proposed Viola-Jones Pseudo Code
그림 10. 제안하는 Viola-Jones Pseudo Code

그림 9는 Viola-Jones 알고리즘의 Pseudo Code[7]로 모든 Windows 사이즈를 탐색하지만, 제안하는 방법은 Multi-Core의 개수만큼 병렬처리를 하게 때문에 반복 횟수가 줄어들어 얼굴검출 성능을 개선할 수 있다.

IV 실험 및 고찰

본 연구에서는 멀티코어에 효율적인 얼굴검출 알고리즘을 구현한 후 싱글코어와 멀티코어에서 성능을 측정하여 비교하였다. 얼굴검출에 필요한 학습데이터는 OpenCV Library에서 제공하는 "haarcascade_frontalface_alt2.xml"를 사용하였고 입력영상은 얼굴 정면이 보이는 이미지 파일을 100장을 사용하였고 대표적인 5개의 이미지 파일의 결과를 표에 나타내었다. 실험에 사용되는 SoC는 Cortex-A8 Single-Core가 내장된 S5PV210와 Cortex-A9 Dual-Core가 내장된 Exynos4210[6]를 사용하였다. SoC에 리눅스 커널 4.6버전을 설치하고 Arm-linux-g++ 4.4.1 컴파일러를 사용하여 성능을 측

정하였다.

Table 1. Face Detection Performance in Single-Core
표 1. 싱글코어에서 얼굴검출 성능 (단위:ms)

크기	Viola-Jones	제안된 방법	검출 유무
177x236	7,186	7,226	O
236x315	12,354	12,701	O
225x225	13,346	13,611	O
160x200	6,076	6,165	O
179x282	11,154	11,215	X
평균	10,023	10,183	-

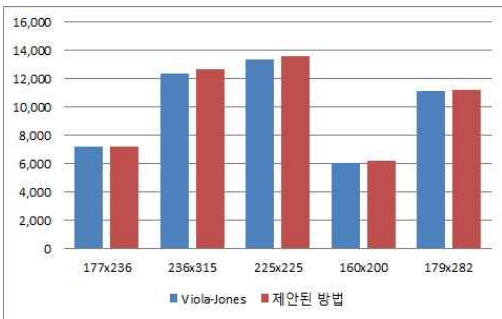


Fig 11. Result Graph in Single-Core
그림 11. Single-Core에서 결과 그래프

Table 2. Face Detection Performance in Dual-Core
표 2. 듀얼코어에서 얼굴검출 성능 (단위:ms)

크기	Viola-Jones	제안된 방법	검출 유무
177x236	4,235	2,361	O
236x315	7,362	3,904	O
225x225	7,341	4,125	O
160x200	3,610	1,977	O
179x282	6,651	3,539	X
평균	5,840	3,181	-

표 1과 같이 싱글코어에서는 제안한 병렬처리 얼굴 검출보다 기존의 Viola-Jones 알고리즘이 미세하게 더 빠른 성능을 보였다. 그 이유는 제안하는 방법은

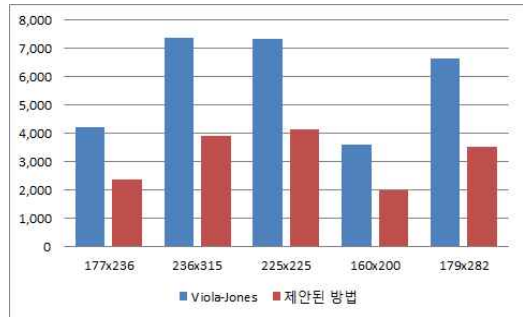


Fig 12. Result Graph in Dual-Core
그림 12. Dual-Core에서 결과 그래프

POSIX Thread를 사용하여 병렬처리 연산을 하기 때문에 Single-Core에서는 의미가 없고 또한 POSIX Thread를 생성하고 관리하는데 시간이 소요되어 효과가 없다. 하지만 표2의 수치와 같이 Dual-Core를 가진 Exynos4210 SoC칩에서 성능을 측정하면 기존의 Viola-Jones 알고리즘에 비해 평균 1.8배 증가하는 결과를 보였다. 인식률의 경우 동일한 학습데이터 과일을 사용하여 기존의 인식률과 변함이 없었다.

V 결론

본 연구에서는 기존의 PC기반의 인식 시스템이 아닌 SoC를 이용한 임베디드 시스템에 OS 환경을 구축하고 피부색정보와 병렬처리 방법을 이용하여 얼굴 검출 시스템을 구현하였다.

기존의 얼굴검출 알고리즘으로 널리 사용되는 Viola-Jones 알고리즘을 멀티코어가 내장된 임베디드 시스템에 맞게 설계하였고 그 결과를 수치로 나타내었다. 그 결과 기존의 알고리즘보다 약 1.8배 빠른 검출 속도를 나타내었다. 이 결과는 코어의 클럭과 FPU의 성능개선도 영향을 미쳤지만 병렬처리 방법이 가장 큰 영향을 미쳤다고 할 수 있다.

향후 Core의 개수가 증가하고 병렬 처리량이 많아지면 현재 제안한 얼굴검출 방법은 더욱 빠른 속도가 측정될 것으로 생각된다. 얼굴검출 성공률의 경우는 모두 동일한 학습데이터 파일을 사용하였고 개발환경 또한 동일하여 기존의 소스와 동일한 검출 성공률을 보였다. 영상 데이터의 얼굴 방향이 정면을 향하고 일정수준 이상의 크기와 화질을 가진 데이터는 99%의 얼굴검출 성공률을 보이지만 그 외의 얼굴의 각도가 달라진 사진 또는 배경화면에 따라 성공률은 많이 달라진다. 앞으로 스레드를 활용한 병렬처리 외에도 SIMD 명령어를 통한 연구도 진행할 것이며 다양한

알고리즘을 멀티코어에 맞게 구현하는 연구도 진행할 것이다.

References

- [1] M. Tusch, "High-Performance Image Processing on FPGAs," Xcell Journal, Vol.57, No.2, pp.42-44, April. 2006.
- [2] Cortex-A9 MPCore Technical Reference Manual, <http://www.arm.com>
- [3] P. Viola, M.J. Jones, "Robust Real-Time Face Detection", International Journal of Computer Vision, Vol.57 No.2, pp.137-154, May. 2004.
- [4] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: a survey", In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, No.1, pp.34-54, January. 2002.
- [5] Cortex-A9 MPCore Technical Reference Manual, <http://www.arm.com>
- [6] Samsung Exynos 4210 RISC Microprocessor User's Manual, <http://www.samsung.com/sec>
- [7] Bo-Cheng Charles Lai, Chih-Hsuan Chiang, Guan-Ru Li, "Data Locality Optimization for A Parallel Object Detection on Embedded Multi-Core Systems", Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on, pp.576-579, July. 2011.
- [8] D. Chai, K.N. Ngan, "Locating Facial Region of a Head-and-Shoulders Color Image," Proc. Third Int'l Conf. Automatic Face and Gesture Recognition, pp. 124-129, 1998.
- [9] Gove. Darryl, Multi-Core Application Programming, Pearson Education, pp. 21-52, 2011.
- [10] OpenMP Application Program Interface, <http://www.openmp.org/>
- [11] RealView Compilation Tools, <http://www.arm.com>

BIOGRAPHY

Kim Hong-Hee (Member)



2011 : BS degree in Computer Engineering, Hanbat University.
2011 ~ Present : MS Course in Computer Engineering, Hanbat University.

Lee Jae-Heung (Member)



1983 : BS degree in Electrical Engineering, Hanyang University.
1985 : MS degree in Electrical Engineering, Hanyang University.
1994 : PhD degree in Electrical Engineering, Hanyang University.
1989 ~ Present : Professor in Dept. of Computer Engineering, Hanbat University