韓國 컴퓨터情報學會 論文誌
第11卷 第2號, 2006. 5.

2006-11-2-1-18

# HMM(Hidden Markov Model) 기반의 견고한 실시간 립리딩을 위한 효율적인 VLSI 구조 설계 및 FPGA 구현을 이용한 검증

이 지 근*, 김 명 훈**, 이 상 설***, 정 성 태****

# Design of an Efficient VLSI Architecture and Verification using FPGA-implementation for HMM(Hidden Markov Model)-based Robust and Real-time Lip Reading

Chi-Geun Lee *, Myung-Hun Kim **, Sang-Seol Lee ***, Sung-Tae Jung ****

## 요 약

립리딩은 잡음이 있는 환경에서 음성 인식 시스템의 성능 향상을 위한 한 방법으로 제안되었다. 기존의 논문들이 소프트웨어 립리딩 방법을 제안하는 것에 반하여, 본 논문에서는 실시간 립리딩을 위한 하드웨어 설계를 제안한다. 실시간 처리와 구현의 용이성을 위하여 본 논문에서는 립리딩 시스템을 이미지 획득 모듈, 특징 벡터 추출 모듈, 인식 모듈의 세 모듈로 분할하였다. 이미지 획득 모듈에서는 CMOS 이미지 센서를 사용하여 입력 영상을 획득하게 하였고, 특징 벡터 추출 모듈에서는 병렬 블록매칭 알고리즘을 이용하여 입력영상으로부터 특징벡터를 추출하도록 하였고, 이를 FPGA로 코딩하여 시뮬레이션 하였다. 인식 모듈에서는 추출된 특징 벡터에 대하여 HMM 기반 인식 알고리즘을 적용하여 발성한 단어를 인식하도록 하였고, 이를 DSP에 코딩하여 시뮬레이션 하였다. 시뮬레이션 결과 실시간 립리딩 시스템이 하드웨어로 구현 가능함을 알 수 있었다.

## Abstract

Lipreading has been suggested as one of the methods to improve the performance of speech recognition in noisy environment. However, existing methods are developed and implemented only in software. This paper suggests a hardware design for real-time lipreading. For real-time processing and feasible implementation, we decompose the lipreading system into three parts: image acquisition module, feature vector extraction module, and recognition module. Image acquisition module capture input image by using CMOS image sensor. The feature vector extraction module extracts feature vector from the input image by using parallel block matching algorithm. The parallel block matching algorithm is coded and simulated for FPGA circuit. Recognition module uses HMM based recognition algorithm. The recognition algorithm is coded and simulated by using DSP chip. The simulation results show that a real-time lipreading system can be implemented in hardware.

▸ Keyword : 립리딩(Lipreading), 음성인식(Speech recognition), 블록매칭(Block matching)

• 제1저자 : 이지근
• 접수일 : 2006.01.16, 심사완료일 : 2006.02.20
* 원광대학교 컴퓨터공학과 박사과정, ** 원광대학교 컴퓨터 공학과 석사과정
*** 원광대학교 전기전자및정보공학부 교수, **** 원광대학교 전기전자 및 정보공학부 교수
* 이 논문은 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(R05-2003-000-10770-02003)

# Ⅰ. 서 론

가정보기술의 발전에 따라 사람과 컴퓨터 사이의 상호작용에 보다 사람 중심의 새로운 개념의 인터페이스가 필요하게 되었다. 여러 가지 대안 중에서 음성 기반 인터페이스가 가장 유망한 수단으로 부각되고 있다. 음성 기반 인터페이스를 위하여 음성 인식에 대한 연구가 아주 오래 전부터 수행되어 많은 음성 인식 시스템들이 개발되었고 장난감, 컴퓨터, 핸드폰, 가전제품 등에 응용되고 있다. 근래에는 잡음에 덜 민감한 음성 인식 시스템을 개발하기 위한 연구가 널리 수행되고 있는데, 입술 모양을 보고 무슨 단어를 발성하는지를 판단하는 립리딩 기술을 보조 수단으로 음성 인식의 신뢰도를 향상시키기 쉬한 연구가 많이 수행되고 있다[1,2,3,4,5,6].

립리딩에 대한 연구는 오래전부터 수행되어 왔지만 모든 연구들이 소프트웨어를 이용한 구현 방법 개발을 다루고 있다. 소프트웨어를 이용한 립리딩은 사용하는 특징 데이터에 따라 영상에 의한 방법과 기하학적 특징에 의한 방법으로 분류할 수 있다[7]. 영상을 이용한 방법에서는 입술 영역의 영상을 그대로 사용하거나 특징 데이터 양을 줄이기 위해 주성분 분석 등의 처리를 하여 사용한다[2,5]. 기하학적 특징을 이용한 방법에서는 영상으로부터 입술의 넓이, 높이, 모양 등을 추출하여 특징 데이터로 사용한다[4,8,9]. 기존의 모든 립리딩 방법은 학습에 기반한 인식 방법을 사용한다. 학습 및 인식을 위해 널리 사용되는 방법으로는 인공 신경망[5,10]과 HMM(Hidden Markov Model)[8,9]이 있다.

본 논문에서는 기본적인 립리딩에 대한 하드웨어 설계를 제안한다. 소프트웨어에 의한 립리딩에서 사용하는 특징 데이터 추출 과정을 하드웨어로 구현하기 위해서는 많은 양의 부동소수점 연산을 필요로 하고 병렬 처리하기에도 부적합하여 본 논문에서는 입술 영역의 영상을 작은 블록들로 나눈 다음에 연속된 프레임 사이에 각 블록의 모션 벡터를 새로운 특징 데이터를 사용하였다. 하드웨어로 구현하면 각 블록들의 모션 벡터 계산을 병렬로 처리할 수 있으므로 처리 속도를 크게 향상시킬 수 있다. 입술 영상에 대한 블록들의 모션 벡터 추출 알고리즘은 FPGA(Field Programmable Gate Array) 회로에 매핑하고 시뮬레이션을 수행하였다. 또한 소프트웨어에 의해서도 입술 영역 블록들에 대한 모션 벡터 추출 과정을 구현하고 이를 이용하여 립리딩을 수행함으로써 모션 벡터를 특징데이터로 사용하는 것에 대한 타당성을 검증하였다. 립리딩을 위한 학습과 인식은 HMM을 이용하였는데, 하드웨어의 크기를 줄이기 위하여 학습 과정은 하드웨어 구현에서 제외하고 인식 알고리즘 부분만 DSP(Digital Signal Processing) 칩을 이용하여 시뮬레이션하였다. 학습 과정은 한번만 수행하면 되므로 컴퓨터에서 소프트웨어로 처리한 다음에 학습 결과만 하드웨어로 다운로드하면 된다. DSP 칩을 이용하여 인식 알고리즘을 시뮬레이션 해본 결과 인식이 잘 수행되었다.

# Ⅱ. 립리딩 시스템의 하드웨어 설계

본 논문에서는 실시간 립리딩 시스템을 구성하기 위하여 크게 소프트웨어 구현 부분과 하드웨어 구현 부분으로 구분하여 연구를 수행하였다. 우선, 립리딩 시스템을 하드웨어로 구현하기에 앞서서 소프트웨어로 먼저 구현하고 이를 시뮬레이션 함으로써 전체 립리딩 시스템 영역에서 하드웨어 구현 범위를 결정짓고, 립리딩 시스템의 하드웨어 구현 가능성과 확실성을 확보하고자 하였다. 소프트웨어 구현 부분에서는 효율적인 실시간 립리딩 시스템을 구성하는데 필요한 기본 알고리즘과 세부 기술을 연구 분석하고 이를 C언어로 프로그램화 하여 순차적인 립리딩 시스템을 구현 하였다. 소프트웨어로 작성된 립리딩 시스템을 이용하여 인식 실험과 시뮬레이션을 통하여 인식률을 평가하였다. 또한 립리딩과 음성인식간의 상호 인식률 보완에 관한 실험도 진행하여 립리딩 시스템의 신뢰성을 확보하였다.

하드웨어 구현 부분에서는 입력 영상 획득을 위하여 CMOS 이미지 센서를 활용하고 립리딩 시스템에서 가장 많은 연산량과 빠른 처리속도를 요구하는 특징벡터 추출 단계를 하드웨어 구현 범위로 정하였다. CMOS 이미지 센서로부터 획득된 입술 영상에 병렬 블록매칭 알고리즘을 적용하여 특징벡터를 추출해냄으로써 립리딩 인식과정의 전체적인 속도를 향상 시키고자 하였다. 립리딩 인식 알고리즘은 HMM 인식 알고리즘을 DSP에 코딩하여 시뮬레이션 하였다.

## 2.1 시스템 구성

본 논문에서 설계한 립리딩 시스템의 하드웨어 구성도는 〈그림 1〉과 같다.

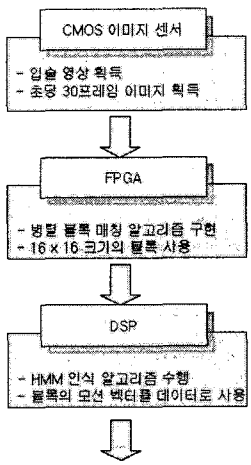


그림 1 하드웨어 구성 및 흐름도
Fig 1. Architecture & Flowchart of H/W

전체 립리딩 인식 과정중 가장 많은 연산량과 빠른 처리 속도를 요구하는 영상 특징 추출 단계를 병렬화 하였고 추출된 특징벡터를 인식하기 위한 인식 알고리즘을 DSP에 디코딩 하여 하드웨어 영역에서 인식 수행하였다. 인식 단계에서 필요한 학습 데이터는 PC 기반에서 프로그램 모듈을 수행함으로써 생성 하였다. 먼저, PC 기반에서 획득된 영상에 병렬 블록매칭 알고리즘을 프로그램 영역에서 수행하여 입술 영상의 모션 벡터를 추출하였다. HMM 학습 알고리즘을 이용하여 추출된 모션 벡터를 학습시켰다. HMM 학습 알고리즘 거쳐 생성된 학습 데이터는 하드웨어 처리부의 인식 모듈로 전달되도록 하였다.

하드웨어 처리부는 크게 영상 획득 단계와 특징 벡터 추출단계 및 인식처리 단계로 구성되어 있다. 본 논문에서는 립리딩에 필요한 입력영상 획득을 위하여 CMOS 이미지 센서를 사용하였다. 획득된 입술 영상은 특징 벡터 추출 단계로 전달된다. 입력 영상에서 크기가 작은 특징 벡터 추출을 위하여 블록매칭 알고리즘을 적용하여 모션 벡터를 추출한다. 이때 블록매칭 알고리즘을 병렬화 함으로써 특징 벡터 추출 단계의 처리속도를 향상 시키고자 하였다. 병렬 블록매칭 알고리즘을 FPGA로 구현함으로써 전체 립리딩 단계에서 가장 많은 처리시간을 요구하는 특징벡터 추출을 위한 연산 처리속도를 향상 시키고자 하였다. FPGA 모듈에서 추출된 입술영상의 모션벡터는 립리딩 인식을 위하여 DSP로 전달된다. 인식 단계에서는 DSP 모듈에서 HMM 인식 알고리즘인 Viterbi 알고리즘을 수행하였다.

## 2.2 CMOS 이미지 센서를 이용한 영상 획득

본 논문에서는 CMOS 이미지 센서를 사용하여 입술 영역을 획득 후 FPGA보드로 전달했다. CMOS 이미지 센서로는 Omnivision사의 ov8610계열을 사용하였다. CMOS 센서에서 영상이 정상적으로 획득되는지를 확인할 수 있도록 하기 위하여 VGA 모니터에 출력되도록 하였다. CMOS 센서의 출력은 디지털 데이터이므로 디지털-아날로그 컨버터를 거쳐서 모니터로 출력되도록 하였다. 이미지 센서와 VGA의 write 시간과 read 시간의 차이로 인해 하나의 램을 사용할 수 없기에 〈그림 2〉와 같이 두 개의 램을 사용하였다. 즉 이미지 센서가 하나의 램을 write 할 동안 VGA에서는 다른 램에서 read하였다. 이미지 센서가 write 하는 시간이 빠르더라도 VGA에서 read가 안 끝났으면 대기하였다가 끝나면 사용하도록 하였다.

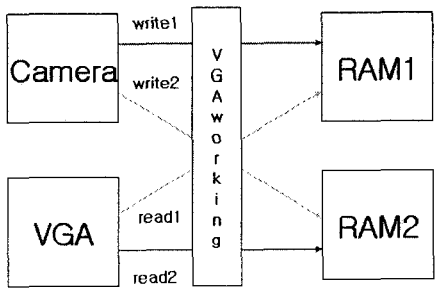


그림 2 모니터 출력을 위한 더블버퍼
Fig 2. Double Buffer for Output images

## 2.3 병렬 블록 매칭

본 논문에서는 입술의 움직임을 파악하여 립리딩에 활용하였다. 입술의 움직임은 픽셀단위로 진행하는 방법과 블록단위로 진행하는 방법을 고려할 수 있다. 픽셀단위의 분석은 각 픽셀에 대한 분석을 바탕으로 하고 있으므로 시간이 많이 소요되므로 본 논문에서는 블록단위로 움직임 정보를 추출하였다. 연속된 프레임 사이에 블록의 움직임은 두 프레임 사이의 블록을 서로 매칭시킴으로써 찾을 수 있다. 블록 매칭 알고리즘에는 여러 가지가 있지만 본 논문에서는 3단계 블록 매칭 알고리즘을 기반으로 하여 병렬화 하였다.

기존의 3-단계 탐색 블록 매칭 알고리즘의 각 단계 m에

서는 현재 블록과 가장 비슷한 블록을 찾기 위하여 아래 식과 같이 각 후보 위치 (i,j)에 대하여 후보 블록과 현재 블록 사이의 픽셀 값의 절대 차이의 합(SAD: Sum of Absolute Difference)을 구한다.

$$SAD^m(i,j) = \sum_{y=0}^{N-1}\sum_{x=0}^{N-1} | a(x,y) - b(x + d_x^m(i,j), y + d_y^m(i,j)) | \quad \cdots\cdots (2.3)$$

여기에서 a(x,y)는 크기가 N×N인 현재 블록의 픽셀 값이다. 본 논문에서는 현재 블록의 크기를 16×16으로 가정한다. $b(x + d_x^m(i,j), y + d_y^m(i,j))$는 탐색 영역의 픽셀 값을 나타낸다. $d_x^m(i,j)$와 $d_y^m(i,j)$는 각 단계 m에서 현재 블록과 후보 위치 (i,j) 사이의 누적된 변위를 나타내는 것으로서 $d_x^m(i,j) = d_x^{m-1}(i,j) + i \times \delta_m, d_y^m(i,j) = d_y^{m-1}(i,j) + j \times \delta_m$과 같이 정의된다. 이들의 초기값은 $d_x^0(i,j) = 0$, $d_y^0(i,j) = 0$이다. (i,j)는 (-1,-1), (-1, 0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)과 같은 값을 가지며 9개의 후보 위치를 나타낸다. $\delta_m$은 각 단계 m에서 각 후보 위치 사이의 거리이다($\delta_1 = 4$, $\delta_2 = 2$, $\delta_3 = 1$).

〈그림 3〉에는 3단계 탐색 블록 매칭 과정의 한 예가 나타나 있다. 첫 단계에서는 가로 세로 방향으로 4씩 떨어져 있는 9개의 위치에서 SAD를 계산한 다음에 최소의 SAD 값을 가진 위치를 선택한다. 두 번째 단계에서는 첫 번째 단계에서 선택된 위치를 중심으로 9개의 후보 위치를 설정하고 SAD 값을 계산한다. 이때에는 후보 위치 사이의 거리가 2가 된다. 마찬가지로 세 번째 단계에서도 두 번째 단계에서 최소의 SAD를 갖는 위치를 중심으로 9개의 후보 위치에서 SAD를 계산한다. 이때에는 후보 위치 사이의 거리가 1이 된다. 3단계에서 최소의 SAD 값을 가지는 후보 위치와 현재 블록 사이의 변위가 최종 모션 벡터 값이 된다. 그림에서는 각 단계의 후보 위치에 단계 번호 1,2,3을 표시하였다. 그리고 1 단계의 후보 위치에는 위치 표시자 $(i,j)$를 좌측 상단에 표시하였다. 그림에서 화살표는 후보 위치 (0,0)의 이동 경로를 표시하고 있다.

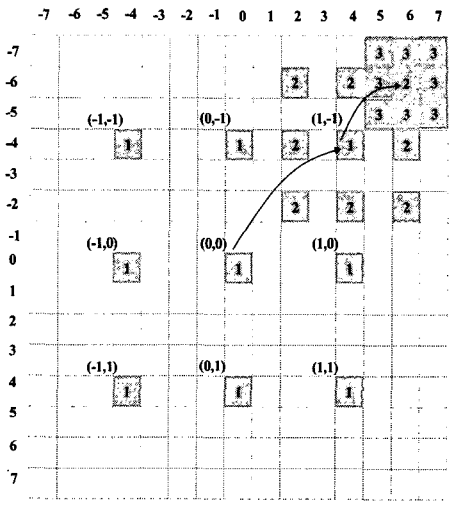


그림 3. 3-단계 탐색 블록 매칭 예
Fig 3. 3-step search block matching method.

3단계 블록 매칭의 각 단계에서는 9개의 후보 위치에 대하여 블록 사이의 SAD 값을 계산하고 그중에서 가장 작은 값을 가진 위치를 선택한다. 이들 9개의 후보 위치에 대한 SAD 값 계산은 순서에 상관없으므로 SAD 값을 계산하는 처리기를 9개 사용하여 SAD 값을 병렬로 계산할 수 있다. 병렬로 처리함에 있어서 문제가 되는 것은 각 처리기에서 SAD 계산에 필요한 픽셀 값을 읽는 데 있다. 예를 들어 9개의 처리기에서 서로 다른 픽셀 값을 필요로 할 때에 이들을 하나의 메모리로부터 동시에 읽는 것은 불가능하다. 이 문제를 해결하기 위한 한 방법은 영상을 9개의 독립된 메모리에 복사해두고 9개의 처리기가 각각 서로 다른 메모리로부터 픽셀 값을 읽게 할 수 있을 것이다. 그러나 이 방법은 너무 많은 양의 메모리를 필요로 할 뿐만 아니라 메모리 대역폭(bandwidth)이 너무 커져서 현실적으로는 구현이 불가능하다.

다른 해결 방법으로는 9개의 처리기를 일렬로 연결한 다음에 파이프라인 처리 방식을 사용하여 한 처리기에서 사용한 픽셀 값을 다음 처리기에 전달하여 사용될 수 있도록 하는 것이다. 파이프라인 처리 방식에서는 파이프라인에 데이터가 가득 차있는 경우에 모든 처리기가 병렬로 동작하게 되고 파이프라인에 데이터가 차기 시작하거나 데이터가 비워지는 경우에 일부 처리기만 동작하게 된다. 파이프라인 처리 방식은 모든 처리기가 항상 병렬로 동작하지는 않지만 적은 양의 하드웨어를 사용하면서도 병렬 처리를 가능하게 해주는 효과적인 방법이다.

본 논문에서는 참고문헌 [11]에서 제안된 4-경로 파이프라인 구조를 사용하여 블록 매칭 알고리즘을 구현한다. 이 구조에서는 파이프라인 경로가 네 개가 있어서 병렬성을 더 높일 수 있는 구조로서 9개의 처리기로 구성되어 있다. <그림 4>에는 하나의 처리기의 구조가 나타나 있다.

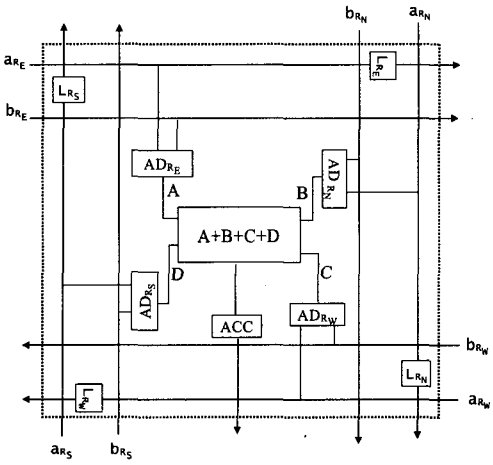


그림 4. 처리기의 내부 구조
Fig 4. Inner Architecture of processor

## 2.4 HMM을 이용한 립리딩

HMM은 다중확률 구조를 갖는 프로세스들을 모델링 하는데 매우 적합한 방법으로서 HMM 파라미터들의 정밀한 계산을 위한 효과적인 알고리즘이 존재한다는 장점 때문에 근래에 음성인식 뿐만 아니라 얼굴 인식, 제스처 인식, 영상인식 등, 패턴인식 영역에서 전반적으로 널리 사용되고 있는 방법이다.

HMM은 확률의 상태와 그들 간의 천이 확률로 정의되며, 각 천이는 상태선택에 관한 상태 천이확률과 천이가 이루어졌을 때 유한개의 관측 심볼로부터 각 출력 심볼이 나올 수 있는 조건부 확률과 관련이 되어있다. 즉, HMM은 관측이 불가능한 하나의 프로세스를 관측이 가능한 심볼로 발생시키는 프로세스를 통하여 추정하는 이중확률 프로세스로서 음성이나 영상과 같이 가변성이 많고 발생과정을 알 수 없는 프로세스를 모델링 하는데 적합하다. HMM은 상태 천이확률, 관측 심볼의 확률분포, 초기 상태확률의 3가지 확률 파라미터로 정의된다.

HMM은 forward-backward 알고리즘, Baum-Welch 재추정 알고리즘, Viterbi 알고리즘을 이용하여 학습 및 인식문제에 적용되어진다. Forward-backward 알고리즘은 관측 열 $O=[O_1, O_2, O_3, \ldots, O_T]$와 HMM $\lambda=(A, B, \Pi)$가 주어졌을 때, 관측 열을 발생시킬 확률 $P(O|\lambda)$를 계산하는 확률 계산법이다. Baum-Welch 재 추정 알고리즘은 초기 모델이 주어졌을 때, 학습 데이터를 사용하여 관측 심볼의 발생확률을 최대화하기 위해 HMM 파라미터 λ를 반복적으로 학습시키는데 사용된다. Viterbi 알고리즘은 최적의 상태 열과 그 상태 열을 통한 확률을 구함으로써 인식하는데 사용된다. 본 논문에서는 HMM 알고리즘이 적용된 HTK(Hidden Markov Toolkit)을 이용하여 학습을 수행하였다. HTK는 HMM인식 알고리즘을 수행할 수 있도록 HMM 모델 생성, 모델의 재추정과 학습 및 인식, 그리고 인식결과 분석에 필요한 라이브러리를 지원하고 있는 인식관련 툴킷이다. 인식에 필요한 특징 데이터는 블록 매칭에서 구한 모션 벡터들을 HMM 파일 형식에 맞게 변환하여 이용하였고 사용된 HMM 모델은 3-state left-to-right 전이형태의 모델을 사용하였다. 학습 과정은 계속적으로 수행할 필요 없이 한번만 수행하면 되고 학습을 하기 위해서는 많은 양의 영상 데이터가 필요하기 때문에 하드웨어에서 실행하기 보다는 영상 데이터가 저장되어 있는 컴퓨터에서 실행하는 것이 적합하다. 따라서, 본 논문에서는 학습 과정은 HTK를 이용하여 컴퓨터에서 수행하도록 하였다. HTK에서 인식은 Viterbi 알고리즘을 이용하여 구현되어 있는데, 본 논문에서는 HTK로부터 불필요한 부분을 제거하고 순수하게 Viterbi 알고리즘을 구현하는 부분만 추출해서 다시 C언어 코드로 작성하였다.

# III 실험 결과

## 3.1 소프트웨어를 이용한 립리딩 실험

실험은 본 논문에서 소프트웨어로 구현된 립리딩 시스템을 사용하여 화자 독립일 경우에 잡음 환경에서의 음성인식률 저하를 보상하는 실험과 구현된 립리딩 시스템만의 자체적인 인식 실험을 하였다. 실험에 사용된 립리딩 학습데이터는 PC 카메라를 이용하여 영상을 입력받은 다음에 블록 매칭을 수행하여 구하였다. 학습데이터 단어로는 오디오 및 CD플레이어를 작동시키기 위해 필요한 단어들(재생, 정지, 종료, 앞으로, 뒤로)을 사용하였다. 입력되는 영상은 15 Frame/sec이

고, 저장되는 입술정보는 대략 20 프레임의 이미지를 얻었다. 구현 프로그램으로는 Visual C++ 6.0을 이용하여 블록 매칭 알고리즘을 구현하였으며 학습 및 인식을 위하여 HTK 프로그램을 사용하였다. 실험에서 사용된 음성인식 보완 립리딩 시스템은 음성정보와 영상정보를 분석하여 특징 파라미터를 추출하고 학습과 인식을 한 다음 인식된 결과를 통합하게 된다.

첫 번째 인식 실험에서는 입술인식과 음성인식을 모두 화자독립으로 수행하였으며 학습데이터는 15명, 실험데이터는 2명으로 하였다. 음성인식의 경우, 5개의 단어를 연속 발음한 16비트 PCM형식의 AVI 포맷의 음성 데이터를 사용하였고, 이 음성 파일에 오디오 편집 프로그램인 쿨에디트 프로(Cool Edit Pro)를 이용하여 무 잡음의 음성 데이터에 Brown 잡음을 혼합하여 생성된 잡음-음성 데이터를 사용하였다. 음성 인식 알고리즘으로는 입술인식과 동일하게 HMM을 사용하였으며, 사용된 HMM 모델은 3 state left-to-right 전이형태의 HMM 모델을 사용하여 학습-인식 실험을 하였다. 학습과 인식 도구로는 입술인식과 음성인식 모두 HMM 알고리즘을 이용하였다. 실험결과 화자독립 립리딩만의 인식률은 40%의 결과를 보였고, 음성만을 인식했을 경우 〈표 1〉과 같이 Brown 잡음의 비율에 따라 인식률이 변화하는 음성 데이터와 합병하여 실험하였다.

표 1 잡음에 따른 음성 인식률의 변화
Table 1. Variation of recognition rate according to noises

| 음성잡음 | 5 | 10 | 20 | 30 |
|---|---|---|---|---|
| 음성인식 | 80% | 35% | 30% | 20% |

음성-입술을 합병하는 방법은 음성과 입술정보를 각각 인식한 후 인식 결과를 합병하는 방법을 사용하였다. 입술 인식률( $M_{lip}$ )과 음성인식률( $M_{speech}$ )결과에 잡음도에 따른 가변적 가중치(α)를 주어 인식률을 합병하여 식(3.1)과 같이 합병 인식결과( $M$ )를 구하였다.

$$M = \alpha M_{lip} + (1-\alpha) M_{speech} \quad \cdots\cdots (3.1)$$

$$\alpha = \frac{3 \times \text{잡음도}}{100}, \quad 0 \le \alpha \le 0.9 \ (\text{고정상수} : 3)$$

〈표 2〉는 입술인식, 음성인식 결과에 가중치를 달리하여 합병한 후 인식한 결과이다.

표 2 가변적 가중치에 따른 합병 인식률
Table 2. Recognition rate that combined speech and lip recognition results according to variable weight.

| 가중치(α) | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 음성입술합병 인식률 | 잡음 30 | 20 | 20 | 25 | 25 | 25 | 30 | 35 | 35 | 40 |
| | 잡음 20 | 30 | 30 | 30 | 30 | 30 | 25 | 25 | 35 | 40 |
| | 잡음 10 | 35 | 40 | 45 | 40 | 50 | 50 | 45 | 45 | 45 |
| | 잡음 5 | 80 | 85 | 85 | 75 | 75 | 75 | 55 | 55 | 45 |

〈표 2〉와 같이 0.1에서 0.9사이의 가변적인 가중치를 서로 다른 음성의 잡음도에 각각 적용하여 음성-입술 합병 인식률에 대한 실험을 하였다. 실험결과 값을 이용하여 각 음성의 잡음도에서 최적의 음성-합병 인식률을 결정하는 가중치를 얻을 수 있는 고정상수를 선택하였고, 각 잡음도에 고정상수 3을 곱하여 가중치를 계산하였다. 실험결과 잡음에 따른 가변적 가중치 α에 의하여 음성-입술의 합병 인식률이 40~85%를 나타내는 것을 볼 수 있다. 〈그림 5〉는 본 실험의 입술, 음성, 음성-입술 합병 인식결과를 보여준다.

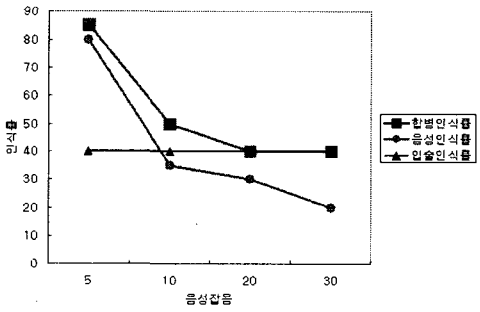


그림 5. 화자독립 입술, 음성, 음성-입술 합병 인식결과
Fig 5. Results of recognition which speaker independent lipreading, speech, combined speech-lipreading

두 번째 실험으로는 블록매칭 알고리즘으로 추출된 모션 벡터를 사용하여 화자종속 립리딩 인식 실험을 하였다. 10명의 화자를 대상으로 각각 10회씩 발성하여 영상을 획득하였다. 그 중 7회 발성한 영상 데이터를 학습 데이터로 사용하고 나머지 3회 발성한 영상은 테스트 데이터로 사용하였다. 이와 같이 동일한 조건으로 학습 데이터와 인식 데이터를 변경하면서 총 10회 립리딩 인식 실험을 하였다. 평균 인식률은 98%의 인식률을 나타냈다.

### 3.2 병렬 블록 매칭 알고리즘 시뮬레이션

병렬 블록 매칭 알고리즘은 Virtex2 XC2V1000 FPGA을 목표 시스템으로 하여 VHDL로 코딩하여 시뮬레이션하였다. 합성 툴은 Synplify, 시뮬레이션 툴은 Active HDL을 사용하였다. 〈그림 6〉에는 블록 매칭 알고리즘 수행을 위한 하드웨어 구조의 블럭도가 나타나 있다. Arbiter_Router_Interface 모듈은 Data_arbiter, Data_router, Memory_decoder, Address_generator 모듈로 구성되는데, 전체적인 제어 신호를 제공한다. Data_arbiter 모듈에서는 PE와 카메라 모듈 사이의 메모리에 대한 접근권을 제어하는 동작을 수행한다. Data_router 모듈은 메모리 모듈에서 읽혀진 픽셀 데이터를 PE에 전달하는 역할을 수행한다.이 모듈에서는 PE가 3단계 블록 매칭을 정확하게 수행하기 위해 동적으로 메모리 데이터와 파이프라인 입력 경로를 배정하는 동작을 수행한다. Address_generator 모듈은 PE에 인가되어야 할 픽셀 데이터의 주소를 생성한다. 또한 이 모듈에서는 매 단계가 종료된 후 MAD로 결정된 PE에 따라 탐색 영역에 대한 다음 단계의 기본 주소를 계산한다. Memory_decoder 모듈은 메모리 모듈에 제어신호를 인가시킨다.

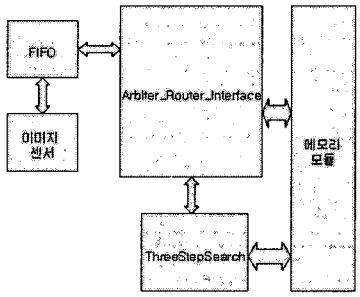


그림 6 블록 매칭을 위한 하드웨어 구조의 블록도
Fig 6. Block diagram of H/W architecture for Block matching

3-단계 블록 매칭을 수행하는 ThreeStepSearch 모듈은 〈그림 7〉과 같이 제어 모듈인 TreeStepSearchControl 모듈과 각 단계의 블록 매칭을 수행하는 SingleStepSearch 모듈로 구성된다.

TreeStepSearchControl 모듈은 SingleStepSerch 모듈에게 3 단계에 걸쳐 각 단계마다 블록 매칭을 수행하도록 지시하고 블록 매칭이 끝나면 최소 블록 매칭 값을 구하게 하는 제어 신호를 제공한다.

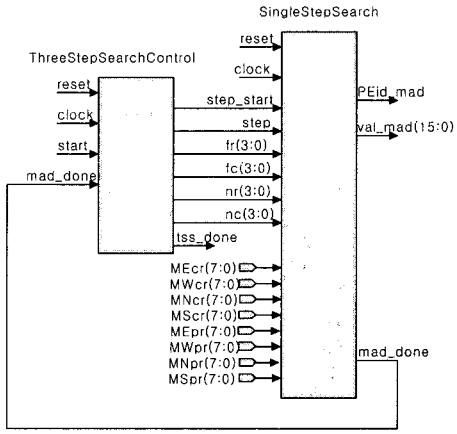


그림 7. ThreeStepSearch 모듈
Fig 7. Threestepsearch module

〈그림 8〉은 VHDL로 구현된 각 PE에서 연산된 탐색영역과 현재블록의 9개 (i,j) 천이에 대한 절대차를 누적한 값을 1단계가 종료된 시점에서 가장 적게 절대차를 기록한 일치도가 높은 PE를 결정하는 과정의 시뮬레이션 결과이다. 여기에서 가장 적은 절대차를 갖는 PE는 PE0로서 0BC0 이다.

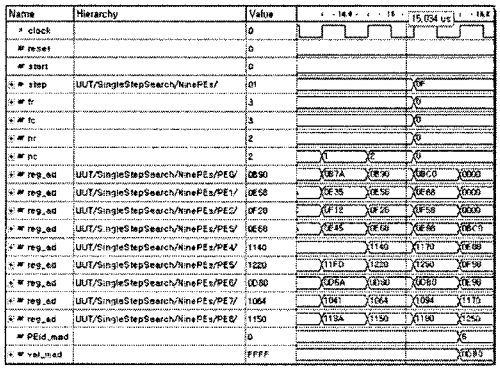


그림 8 시뮬레이션 과정
Fig 8. Precess of simulation

### 3.3 인식 알고리즘 시뮬레이션

본 논문에서는 TMS320C64x 계열의 DSP를 사용하여 Viterbi 기반 인식 알고리즘을 코딩하고 시뮬레이션 하였다. 본 논문에서는 TMS32C6416, SDRAM, SBSRAM, Flash Memory, 버퍼소자 등으로 구성된 TMS320C64xx 모듈을 사용하였다. 본 클럭은 50MHz를 사용하였으며, 이 클럭의 실제 12배 클럭이 DSP 내부에서 사용되어 진다.

DSP 프로그램 개발환경으로 CCS(Code Composer

Studio)를 사용하여 개발 하였다. DSP 프로그램을 CCS에서 구동시에는 GEL파일이 필요한데, 현 모듈에서 지원하고 있는 SDRAM이나 SBSRAM에 관한 셋팅값이 들어 있다. DSP에서 인식 알고리즘을 수행하기 위해서 먼저 DSP 보드를 PC와 연결한 다음에 인식 프로그램을 다운로드하였다. 인식 프로그램은 C언어로 코딩하였으며 되도록 프로그램 크기를 줄이기 위하여 꼭 필요한 부분만 집어넣은 결과 500라인 정도의 크기를 차지하였다. 인식에 필요한 학습 데이터와 테스트 영상은 PC에 저장한 다음에 DSP 보드에서 읽어서 사용하였고 인식 결과도 PC로 전송하여 디스플레이 하였다. 시뮬레이션 결과 인식 과정이 PC에서 소프트웨어로 실행한 경우와 동일하게 잘 수행되었다.

## Ⅳ. 결론

본 논문에서는 신뢰성 있는 립리딩(Lip Reading)을 견고하고 빠른 속도로 수행하기 위하여 립리딩 인식과정에서 핵심적인 단계인 특징벡터 추출단계를 병렬화하여 FPGA에서 실행될 수 있도록 하고 인식 알고리즘은 DSP에 수행되도록 하는 구조를 제안하였다. 입력영상으로부터 실시간으로 특징벡터를 추출하기 위하여 병렬 블록매칭 알고리즘을 적용하였다. 먼저, 블록 매칭 결과를 이용한 립리딩 시스템을 소프트웨어로 구현해서 실험해본 결과 기존의 립리딩 시스템과 유사한 인식률을 얻을 수 있었다.

소프트웨어에 의해 시뮬레이션을 수행한 다음에는 블록 매칭을 이용한 특징 벡터 추출과정을 FPGA로 코딩하고 시뮬레이션을 수행하였다. 또한 Viterbi 알고리즘을 DSP에서 코딩하고 시뮬레이션을 수행하였다. 그리고 이미지 센서를 FPGA에 연결하여 영상을 획득할 수 있도록 하였다. 각 부분별 시뮬레이션 결과는 정상적으로 동작하였으며 앞으로는 전체 시스템을 통합하여 하드웨어로 구현하는 연구를 수행할 예정이다.

## 참고문헌


[1] H. McGurk, J. MacDonald, "Hearing lips and seeing voices", Nature, vol. 264, pp. 746-748, 1976.

[2] C. Bregler, Y. Konig, "Eigenlips for Robust Speech Recognition", Processings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 669-672, 1994.

[3] P. L. Silsbee, A. C. Bovik, "Computer Lipreading for improved accuracy in automatic speech recognition", IEEE Trans. Speech Audio Processing, vol. 4, no, 5, pp. 337 351, 1996.

[4] M. Lievin, F. Luthon, "Lip features automatic extraction", Image Processing, ICIP 98. Proceedings. 1998 International Conference On, Vol.3, pp. 168-172, 1998.

[5] U. Meier, R. Stiefelhagen, J. Yang, A. Waibel, "Towards Unrestricted Lipreading", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 5, pp. 571-785, 2000.

[6] 이창윤, 이영근, "음성인식 시스템에서의 잡음제거 개선에 관한 연구" 한국컴퓨터정보학회 논문지, 2002년 7권 2호.

[7] C. Neti, G. Potamianos, J. Luettin, I. Matthews, H. Glotin, D. Vergyri, "Large-vocabulary audio-visual speech recognition: a summary of the Johns Hopkins Summer 2000 Workshop", Multimedia Signal Processing, 2001 IEEE Fourth Workshop on, pp. 619-624, 2001.

[8] G. I. Chiou, J. N. Hwang, "Lipreading from Color Video", IEEE Transactions on Image Processing, Vol. 6, No. 8, pp. 1192-1195, Aug. 1997.

[9] S.L. Wang, W.H. Lau, S.H. Leung, H.A. Yan, "Real-time Automatic Lipreading System", roceedings of the International Symposium on Circuits and Systems, pp. Vol II 101-104, 2004

[10] D.G Stork, G. Wolff, E. Levine, "Neural network lipreading system for improved speech recognition", International Joint Conference on Neural Networks, pp. Vol II 289 - 295, 1992

[11] C.-G Lee, H.-G. Lee, H.-. Shim, S.-T. Jung, S.-S. Lee, "A 4-way pipelined processing architecture for three step search block-matching motion estimation," IEEE International Conference on Multimedia and Expo, pp. 527 - 530, 20041

[12] 홍성용, "구속조건을 적용한 다이아몬드 탐색 알고리즘에 관한 고속블록정합움직임 추정" 한국컴퓨터정보학회 논문지, 2003년 8권 4호.

## 저 자 소 개

**이 지 근**

2000년 2월 : 원광대학교 컴퓨터공학과 학사
2002년 2월 : 원광대학교 컴퓨터공학과 석사
2002년 3월 ~ 현재 : 원광대학교 컴퓨터공학과 박사과정
관심분야: 영상처리, 영상인식, 컴퓨터 그래픽스

**김 명 훈**

2004년 2월 : 원광대학교 전기전자및정보공학부(공학사)
2004년 3월 ~ 현재 : 원광대학교 컴퓨터 공학과 석사과정
관심분야 : 영상인식 컴퓨터 비젼, 영상처리

**이 상 설**

1984년 2월 : 고려대학교 전자공학과 졸업
1989년 2월 : 한국과학기술원 전기및전자공학과 석사
1994년 2월 : 한국과학기술원 전기및전자공학과 박사
1994년~현재 : 원광대학교 전기전자및정보공학부 교수
관심분야 : 병렬컴퓨터구조, SoC, 영상및통신VLSI, 임베디드시스템

**정 성 태**

1989년 2월 : 서울대학교 공학 석사
1994년 8월 : 서울대학교 공학 박사
1995년 3월~현재 : 원광대학교 전기전자 및 정보공학부 교수
관심분야: 영상처리, 컴퓨터 그래픽스