

실시간 음성인식 및 립싱크 구현에 관한 연구

이형호*, 최두일, 조우연
공주대학교 전기전자정보공학과

A Study on the Implementation of Realtime Phonetic Recognition and Lip-synchronization

H.H. Lee*, D.I. Choi, W.Y. Cho.
Dept. of Electrical Electronic & Information Eng., Kongju National Univ.

Abstract - 본 논문에서는 실시간 음성 인식에 의한 립싱크(Lip-synchronization) 애니메이션 제공 방법에 관한 것으로서, 소정의 음성정보를 인식하여 이 음성 정보에 부합되도록 애니메이션의 입모양을 변화시켜 음성 정보를 시작적으로 전달하도록 하는 립싱크 방법에 대한 연구이다.

인간의 실제 발음 모습에 보다 유사한 립싱크와 생동감 있는 캐릭터의 얼굴 형태를 실시간으로 표현할 수 있도록 마이크 등의 입력을 받고 신경망을 이용하여 실시간으로 음성을 인식하고 인식된 결과에 따라 2차원 애니메이션을 모평하도록 모델을 삼고 있다.

1. 서 론

인간이 서로의 의사소통의 수단으로 사용하는 것 중에서 가장 기본적이고 가장 많이 사용하는 것은 음성이다. 인간에 의한 음성 처리는 크게 음성 생성(speech production)과 음성인지(speech perception)의 두 가지 측면으로 나누어 볼 수 있다. 이러한 음성의 두 가지 측면과 관련된 연구들이 각기 개별적으로 이루어져 왔으며, 그러한 분야들은 언어학, 음성학, 음운학, 생리학, 해부학 등 다양한 학문적인 배경하에서 진행되어 왔다. 1960년대부터 음성의 발성과 이해에 관해 많은 기초적 연구가 수행되어온 이래 기계에 의한 연속 음성인식, 합성에는 아직 많은 과제가 남아있지만 최근 30~40 여년간 연구결과로 고립단어 인식에 있어서는 많은 발전이 있었으며 그러한 결과들이 신호처리 기술과 고속의 컴퓨터 처리 기술의 발달로 인해서 단순히 실험적인 결과가 아닌 실용적인 측면에서 활용하는 연구가 활발히 진행되고 있다.

이러한 예로 철도 또는 항공편 안내 및 예약, 통역전화, 자동통역 시스템, 여행정보 안내 시스템, 관광안내 시스템, 음성구동 퍼스널 컴퓨터, 음성 다이얼링 휴대폰, 증권정보 안내 시스템 등이 개발되어 상품화되어 있다. 그러나 요즘 급속한 멀티미디어 환경의 발달로 음성을 인식하여 음성합성이라는 기술을 이용한 청각적인 서비스 외에 캐릭터나 애니메이션을 이용하여 시각적인 서비스를 제공하려는 연구가 활발히 진행되고 있다.

그러나 이러한 립싱크에 대하여 종래에는 임의의 텍스트를 입력하고 텍스트로부터의 음성 합성 결과를 립싱크하는 방식을 취하고 있거나, 인간의 실제 음성을 입력하고 이에 맞추어 고도로 숙련된 컴퓨터 그래픽 엔지니어들의 작업에 의해 립싱크 동작 화면을 완성하는 방식으로 이것들은 많은 시간이 소요되어 실시간으로 이루어지지 못한다는 단점이 있다.

이에 음성인식 시스템과 자기 생성 및 구조화 신경 회로망(SCONN)을 이용하여 음성신호 처리를 분석하였으며 입 모양이 변하는 자음의 입술소리(4개)와 기본모음(7개)을 애니메이션으로 구현하는데 목적이 있다.

2. 실시간 음성인식 및 립싱크 구현

2.1 실시간 립싱크 시스템 개요

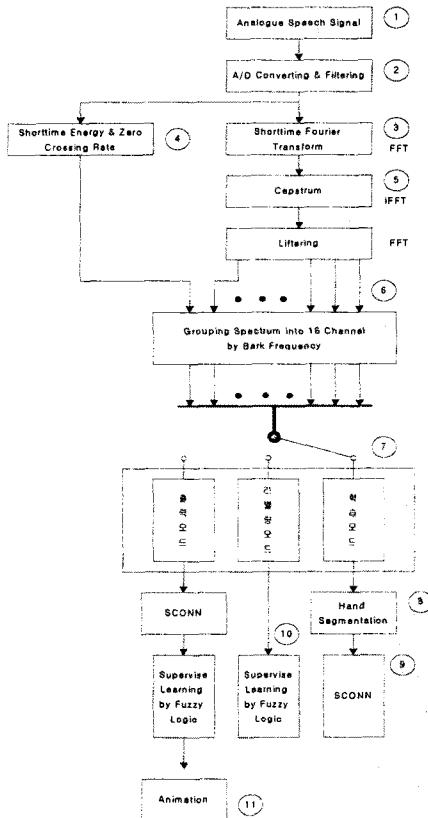


그림 2.1 실시간 립싱크 시스템의 전체 블록도

실시간으로 음성 데이터를 취득하고 음성 파형을 실시간으로 시간-주파수 영역으로 변환하기 위해서 Double Buffering 기법을 사용하였다.[1]

2.1.1 Double Buffering

Double Buffering 기법은 마이크와 sound card에 대해서 샘플링 된 음성 데이터는 Buffer의 개수가 2개로 구성된 Buffer Bank의 첫 번째 Buffer에 저장이 되기 시작한다. 발성을 시작하게 되면 음성 데이터가 그림 2.2에 보는 바와 같이 실행순서 1번과 같이 Buffer 1에 저장이 되고 다음 실행순서 2로 넘어가면 Buffer 1에서는 Data Block으로 Copy 되고, Buffer 2는 음성 데이터를 저장한다. 다음 실행순서 3으로 넘어가면 Buffer 1은 다음 데이터를 받기 위해 Free가 되고

Buffer 2는 Data Block으로 Copy 되고, Buffer 1은 음성 데이터를 저장한다. 이와 같은 과정을 순서 4, 5, ..., 이후로 음성 데이터 입력이 종료 될 때까지 반복 한다.

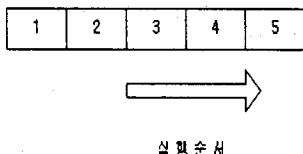
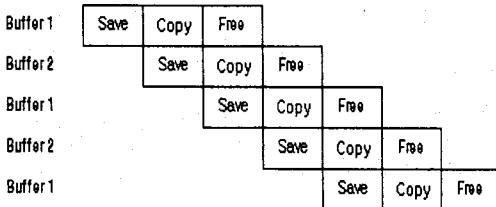


그림 2.2 Double Buffering 기법

2.1.2 음성 인식

일반적으로 푸리에 변환을 하게 되면 크기와 위상으로 분리되어 변환하게 된다. 그중 음성 인식에 사용되는 성분은 주로 크기 성분이 이용된다. 따라서 본 시스템은 위상 성분을 제외한 크기 성분인 스펙트럼 파워만 추출하여 음성인식에 이용하였다.

스펙트럼의 파워를 계산하기 위하여 그 전처리 과정으로 pre-emphasis와 석 2-1과 같이 main lobe의 대역 폭이 좁아서 해상도가 좋고, main lobe와 side lobe의 차이가 -40 dB 이하로 친볼루션에 의해 생기는 왜곡이 작아서 두가지 특성을 가장 적절하게 만족하는 창합수 특성을 보이고 있는 해밍 창 과정을 거친 뒤 FFT로 스펙트럼 파워를 계산하였다.

$$W(n) = 0.54 - 0.46 \cos\left(2\pi \frac{n}{N-1}\right), \quad 0 \leq n \leq N-1 \quad (2-1)$$

영교차율(ZCR)이라 함은 보통 기준점에서 데이터가 양에서 음으로 바뀌거나 음에서 양으로 바뀔 때의 비율을 말한다. 단구간 에너지와 마찬가지로 음성과 잡음은 영교차율(ZCR)에서 확실히 구분되는 특성이 보여므로 음성여부를 판별할 때 중요한 자료로 사용된다. 각 frame의 영교차율(ZCR)은 석 2-2에 보여지는 알고리즘으로 구할 수 있다.

$$ZCR_i = \frac{\sum_{k=0}^{N-1} count(x_{i,k}, x_{i,k+1})}{N}$$

$$\text{if } [(x_{i,k} - M_i)(x_{i,k+1} - M_i) < 0] \quad (2-2)$$

$$\text{then } count(x_{i,k}, x_{i,k+1}) = 1$$

$$\text{else } count(x_{i,k}, x_{i,k+1}) = 0$$

i 는 각 프레임, N 은 Window Size, M_i 은 평균값을 나타낸다.

2.1.3 SCONN을 이용한 학습과정

1994년 최두일이 제안한 SCONN의 알고리즘은 표 2-1과 같다. [2]

- Step 1. Initialize Weights
- Step 2. Present New Input
- Step 3. Calculate Distance to All Node(s)
- Step 4. Find Active Node(s) and a Winner Node
- Step 5. If Active Node Does not Exist, then go to step 8
- Step 6. Decrease Response Ranges of Active Node(s) Increase Response Ranges of Inactive Node(s)
- Step 7. Adapt Weights of Winner Node (or Winner node and its family nodes) go to Step 2
- Step 8. Create a Son Node from an Inactive Winner (Mother) Node go to Step 2

표 2-1 SCONN의 알고리즘

2.1.4 라벨링 과정

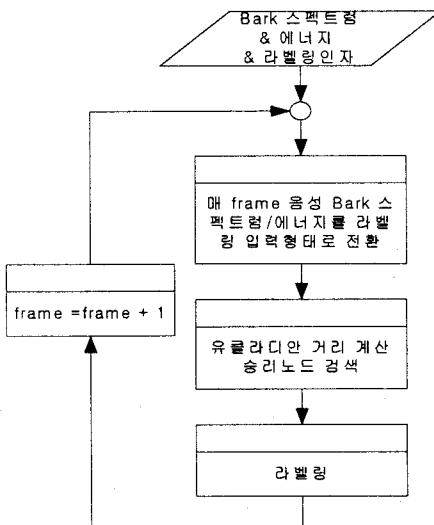


그림 2.3 라벨링의 플로차트

지금 까지 학습된 각 노드는 입력되는 Bark 스펙트럼에만 적용이 되었을 뿐 text(자음, 모음)에는 적용이 되지 못한 상태이다. 따라서 그림 2.3에 보이는 플로차트 순으로 라벨링 과정을 통해, 학습된 노드와 text와 적용시키는 과정이 요구되는데 이를 라벨링이라 한다.

2.1.5 학습된 결과를 애니메이션으로 구현

학습과 라벨링을 통해 생성된 데이터를 가지고, 음성 인식을 실시하면 학습에 의하여 라벨링 된 데이터와 가장 근접한 결과를 각 프레임마다 확률을 계산하여 매 프레임 별로 가장 높은 확률을 기록한 글자의 애니메이션으로 결과를 볼 수 있게 출력하였다.

2.2 실험 및 결과 고찰

위 실험은 자음 13개와 모음 7개의 인식 결과를 애니메이션으로 출력하기 위한 것이다. 우리나라의 자음은 표 2-2에 나타나 있듯이 입술소리를 발음할 경우 입술이 움직이고 나머지는 혀의 위치에 의해 발생되는 경우이다.

자음은 71%, 모음은 90%의 인식 결과를 나타냈으며 자음에서 예사소리([ㄱ, ㄷ, ㅂ, ㅅ, ㅈ]) 계열이 전반적으로 인식률이 낮았고 나머지는 평균 이상의 인식률을 나타냈다.

소리를 내는 자리		두입술	윗잇몸, 허끌	경구개, 혀바닥	연구개, 혀뒤	목청사이	
소리를 내는 방법		명 칭	입술소리	혀끌소리	구개음	연구개음	목청소리
안 울 림 소 리	파열음	예사소리 된소리 거센소리	ㅂ ㅃ ㅍ	ㄷ ㄸ ㅌ		ㄱ ㄲ ㅋ	
	파찰음	예사소리 된소리 거센소리			ㅈ ㅉ ㅊ		
	마찰음	예사소리 된소리		ㅅ ㅆ			ㅎ
울림 소리	비음		ㅁ	ㄴ		ㅇ	
	유음			ㄹ			

표 2-2 자음의 분류

애니메이션의 구현에서 기본 모음(7개)과 입력이 없는 경우의 애니메이션은 입술의 기본 좌표를 계산하여 데이터화했으며 입술모양이 변할 경우 자연스러운 입술모양을 얻기 위해서 중간 과정을 보간 하는 모핑 기법을 사용했다.

다음 그림은 각 모음의 음성인식에 대한 결과의 애니메이션을 출력한 그림이다.



그림 2.4 “ㅏ”의 애니메이션



그림 2.5 “ㅓ”의 애니메이션



그림 2.6 “ㅣ”의 애니메이션



그림 2.7 “ㅗ”의 애니메이션

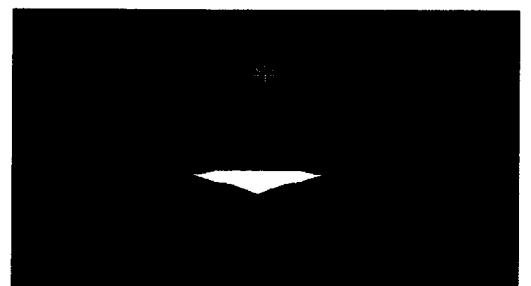


그림 2.8 “ㅜ”의 애니메이션

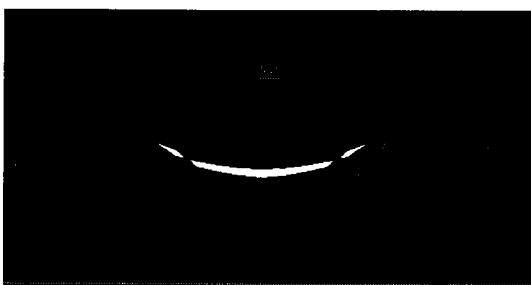


그림 2.9 “ㅡ”의 애니메이션



그림 2.10 입력이 없을 때의 애니메이션

3. 결 론

본 연구에서 자음과 모음의 인식에 의해 2차원 애니메이션으로 구현했지만 이것을 3차원의 애니메이션으로 구현하고 연속단어의 인식에 관한 연구가 이루어진다면 많은 분야에서 활용될 것으로 기대 된다.

(참 고 문 헌)

- [1] Richard j. Simon, "멀티미디어&ODBC", 대림, 617-618, 1997
- [2] D. Chio and S. Park, "Self-Creating and Organizing Neural Networks", IEEE Trans. on Neural Networks, Vol. 5, No. 4, pp.561-575, July 1994