논문 2005-42SP-4-12

고품질 내장형 음성합성 시스템을 위한 음성합성 DB 구현

(The implementation of database for high quality Embedded Text-to-speech system)

권 오 일*

(Ohil Kwon)

요 약

음성 데이터베이스는 TTS 시스템에서 가장 중요한 요소 중의 하나이다. 특히, 내장형 TTS 시스템에서는 서버형 TTS 시스템에서보다 좀 더 작은 데이터베이스를 필요로 한다. 이러한 이유로, 음성합성 데이터의 압축과 통계적 축소과정의 비중은 내장형 TTS 시스템에서 아주 중요한 항목이라고 말할 수 있다. 그러나 이러한 압축과 통계적 축소과정은 합성음질의 저하를 유발시킨다. 본 논문에서는 고품질 내장형 TTS 시스템에서의 데이터 구축방법을 제안하며, MOS 테스트를 통한 합성음질을 검증한다.

Abstract

Speech Database is one of the most important part of Text-to-speech(TTS) system. Especially, the embedded TTS system needs more small size of database than that of the server TTS system. So, the compression and statistical reduction of database is a very important factor in the embedded TTS system. But this compression and statistical reduction of database always rise a loss of quality of the synthesised speech. In this paper, we propose a method of constructing database for high quality embedded TTS system and verify the quality of synthesised speech with MOS(Mean Opinion Score) test.

Keywords: TTS, Embedded System, MOS, Database, Speech

I. 서 론

본 논문의 목적은 코퍼스 기반 고품질 내장형 한국어 음성합성 (TTS : Text to Speech) 시스템을 위한 음성합성 DB 구현 방안을 제안한다.

최근 서버형 음성합성 시스템의 개발 경향을 보면 음성 DB의 크기가 3GB에 달하는 제품이 있을 정도로 대용량 음성 합성 DB를 선호하는 추세이다. 코퍼스를 기반으로 하는 음성합성 시스템(이하 TTS 시스템) 의 특성상 보다 많은 트라이폰을 확보하면 할수록 합성음의 더 높은 품질을 기대할 수 있기 때문이다[1][3]. 그러나 이러한 대용량 TTS 시스템의 경우 합성의 정확성과 자연성 그리고 명료도라는 측면에서 상당한 장점이 있음에도 불구하고 기타 소형 응용제품에 내장하기 위해서는 전체 용량의 제한이 따른다. 따라서 내장형 코퍼스 기반 TTS 시스템의 경우, 서버 형 TTS 시스템이 음성합성 DB를 대규모로 확보하려고 노력하는 것과는 반대로 음성합성 DB의 크기를 최소한으로 줄여야 할 필요가 있다. 이때 필요한 기술이 음성합성 DB의 압축 기술과 축소 기술이다. 음성합성 DB의 압축과 축소는 필연적으로 전반적인 합성음 품질의 저하를 수반하게 되는데, 이러한 합성 음질의 저하를 최소한으로 줄이는 것이 내장형 TTS 시스템의 발전 방향이 될 것이다.

본 논문에서 구현하고자 하는 시스템은 여성 TTS와 남성 TTS 시스템으로 1,500Mbyte 용량의 서버형 TTS시스템을 기반으로 하여 음성합성 DB를 압축한 64Mbyte형과 32Mbyte형 등 2 가지를 기본으로 한 내장형 TTS 시스템이다. 임의의 문장을 선정한 후 합성했을 때 성공률과 정확도 95% 이상, 그리고 MOS Test (Mean Opinion Score Test) 결과 3.0 이상이 본 논문의 목표이다.

* 평생회원, 현대오토넷
(Hyundai Autonet Co., LTd)
접수일자: 2005년3월14일, 수정완료일: 2005년6월8일

Ⅱ. 본 론

1. TTS 시스템 일반

본 논문에서 구현할 TTS 시스템의 기본 구조는 <그림 1>과 같다.

텍스트 분석 단계는 TTS 시스템에 입력으로 들어온 문장을 최종적으로 발음 음소열로 변환하는 단계이다. 전처리 과정은 형태소 분석 및 발음변환 이전에 미리 문장을 정제하는 과정으로, 예외 사전을 이용하여 발음을 변환해주거나 괄호 및 특수기호, 한자 등을 처리하는 과정이다. 형태소 분석은 문자로 표기된 자연언어 문장에 대해 그 문장을 구성하는 형태소 간의 접속 관계를 분석하는 단계이며, 자연언어 문장에 관한 여러 가지 언어 분석을 함에 있어서 가장 먼저 이루어지는 작업이다. 운율생성 단계에서는 1984년에 Breiman에 의해 제안된 CART(Classification and Regression Tree)[1][5]와 데이터 마이닝의 결정트리 방법을 사용하여 얻은 28,163개의 여성 샘플과 30,637개의 남성 샘플을 이용해 강세구와 억양구를 예측하고 이들의 특징 변수를 추출한 뒤 음의 높이(pitch), 강도(intensity), 길이(duration)를 예측하였다.

음성합성 DB 구축 단계는 크게 문장 선정 단계와 음성 녹음 및 분석 단계, 그리고 데이터 추출 단계로 나누어 살펴볼 수 있다. 코퍼스 기반의 TTS 시스템[2]에서 DB 구축의 중요성은 가장 크게 고려되고 있는 부분이다. 서버형 TTS시스템의 경우 가능한 모든 트라이폰을 포함하는 것이 음질 향상에 보탬이 되기 때문에 대용량 음성합성 DB 구축 시스템을 설계하고, 또 지속적인 음성합성 DB 추가를 통한 품질 개선에 힘쓰고 있다. 내장형 TTS 시스템의 음성합성 DB 구조는 서버형 TTS 시스템의 구조와 동일하다. 단지 음성합성 DB의 용량을 줄이는 압축과 축소의 과정이 한 단계 더 부가될 뿐이다. 따라서 <그림 1>의 음성합성 DB구축 다음 단계에 음성합성 DB 압축과 축소의 과정이 필요하다. 이렇게 생성된 음성합성 DB는 운율생성부, 신호합성부와 연동하면서 합성음을 산출한다.

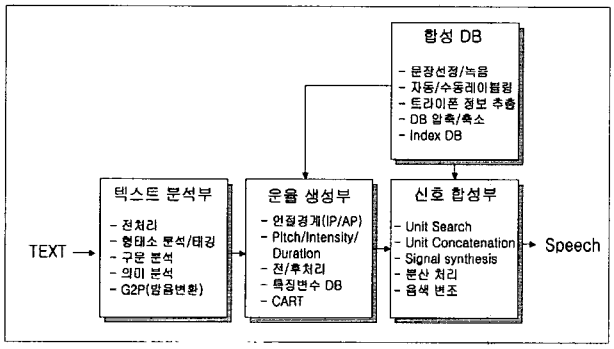

그림 1. TTS 시스템 기본 구조
Fig. 1. The basic structure of the TTS system

이러한 운율 예측 정보[3][9][10][11]를 바탕으로 합성단위 후보를 선정하였으며, 음소적 거리와 운율적 거리 그리고 위치정보 거리를 고려하여 비터비 탐색에 의한 최적 단위열을 찾아 음성 신호를 합성하였다.

2. 음성합성 DB 구축

본 논문에서 구현한 TTS 시스템의 음성합성 DB의 구축 및 순서는 <그림 2>와 같다.

가. 문장선정

TTS 시스템에서 매우 중요한 역할을 담당하고 있는 음성합성 DB를 구축하기 위해서는 무엇보다 먼저 트라이폰 합성단위가 적절히 포함되고, 다양한 운율이 반영된 문장을 선정하는 것이 필요하다. 출현 가능한 모든 트라이폰이 적절히 포함된 문장셋을 선택하기 위해서는 되도록 용량이 큰 텍스트 코퍼스를 사용해야 하며, 주어진 텍스트를 분석하여 발음변환(G2P : Graphic to Phoneme)을 해줄 수 있는 툴을 사용하여야 한다. 만약, 텍스트 코퍼스의 크기가 작다면, 한국어 음성에서 출현하는 모든 트라이폰을 포함하지 못할 우려가 있다. 또한 G2P의 성능이 좋지 못하다면, 추출된 트라이폰 결과도 믿지 못한 결과가 되고 말 것이다. 본 연구에서는 세종 텍스트 코퍼스와 자체적으로 개발한 G2P 툴을 사용

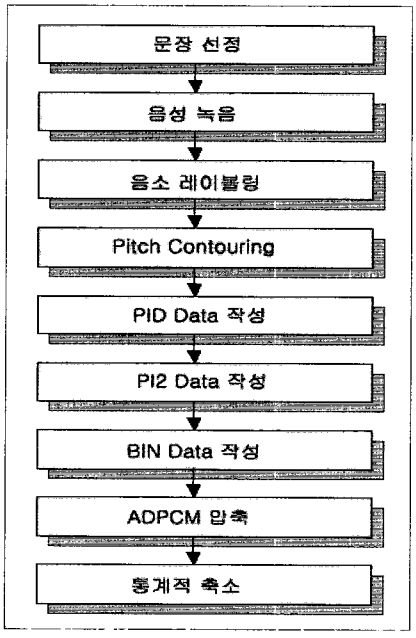

그림 2. 음성합성 DB 구축 절차
Fig. 2. The collection procedure of the DB

하여 트라이폰을 추출하였다.

본 연구에서 트라이폰 분석을 위하여 사용한 문장은 총 1,345,101 문장*이다.

(1) 발음변환(G2P)

발음변환(G2P)은 표시된 기호와 발성되는 기호의 상이함으로부터 기인되는 문제를 해결하기 위한 것으로 음운의 변동에서 규칙 변동은 발음 변환표를 따로 작성한 뒤 이를 이용하였고 불규칙 변동은 한국어 발음사전을 형성한 후 이 사전을 검색하여 음운기호를 생성하도록 하였다.** 트라이폰은 주어진 음소와 그 음소 앞뒤에 오는 음소를 보는 단위로서 음성인식과 음성합성에서 많이 사용되는 단위이다.

(2) 트라이폰 추출

본 연구에서는 트라이폰을 추출하기 위해서 모든 문장을 위와 같은 트라이폰열로 변환시킨 후 유일하게 정의되는 트라이폰을 결정하였다. 즉, 그 방법으로는 첫째, 모든 텍스트 코퍼스의 문장을 G2P 툴을 사용하여 발음기호로 변환하고 둘째, 발음기호로 표현된 문장을 모두 트라이폰 열로 변환하며 셋째, 모든 문장을 탐색하여 유일한 트라이폰을 추출한다. 이와 같은 방법을 사용하여 유일한 트라이폰을 추출하였는데, 그 개수는 모두 18,233개를 얻었다. 그리고, 위의 과정에서 임의의 유일한 트라이폰에 대하여 총 발생 개수를 세어 저장하도록 하는데, 이는 문장셋 선정에 사용된다.

(3) 트라이폰 문장셋 선정

이미 위에서 언급한 바와 같이 코퍼스 기반 음성합성기는 다양한 음가와 운율을 포함하고 있는 음성합성 DB에서 적절한 합성단위를 찾아 이를 연쇄시키는 방식이다. 따라서, 음성합성 DB는 가능한 모든 합성단위가 포함되어 있어야 하며, 더욱 가능하다면 다양한 운율현상이 반영되어 있는 문장을 선택하면 바람직할 것이다. 따라서, 녹음할 문장을 선택함에 있는 우리는 다음의 조건을 만족하도록 문장셋을 선정해야 한다. 첫째, 모든 합성단위가 포함되어야 하며 둘째, 다양한 운율이 반영되어 있어야 한다. 이 두 조건은 문장셋이 가져야할 기본적인 조건이며, 합성기의 용량을 생각할 때 다음과 같은 세 번째의 제약이 주어진다. 셋째, 가능한한 선택문장의 크기가 작아야 한다. 그러나, 위의 세 가지 모든

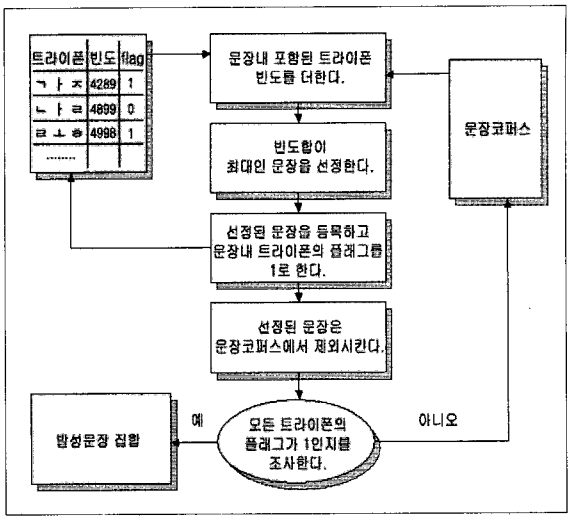

그림 3. 발성 문장셋 선정방법

Fig. 3. The selection method of the target sentence

조건을 만족하는 문장셋을 구하는 것은 어려울 뿐 아니라, 문장셋을 결정한다고 하더라고 그 용량은 굉장히 크게 될 것이다. 왜냐하면, 각 합성단위인 트라이폰에 존재하는 운율변화는 무궁무진하여 운율을 고려한 합성단위가 될 때 그 수는 기본단위 수의 약 수십 배에 해당하게 될 것이기 때문이다. 우리는 모든 출현 트라이폰이 포함되는 문장셋을 선정하게 될 것이며, 이 문장셋의 용량은 되도록 작아야 할 것이다. 물론, 문장셋의 규모가 최적이면 좋겠으나, 그 알고리듬은 구현하기도 힘들고 계산량도 많이 필요할 것이므로 본 연구에서는 논리적으로 타당한 알고리듬을 적절히 고안하여 사용하였다.

<그림 3>과 같은 방법으로 문장이 선정되는데, 이를 위하여 우선적으로 정의된 모든 유일한 트라이폰은 그 출현빈도의 역순으로 정렬된다. 다음 텍스트 코퍼스 내의 문장을 하나씩 가져와 아직 찾아지지 않은 트라이폰의 개수가 가장 많은 문장을 찾는다. 단, 선택될 문장은 현재 찾고자 하는 트라이폰이 꼭 포함되어 있어야 한다. 일단, 최적 문장이 선택되면, 이 문장에 출현된 트라이폰의 플래그를 1로 한다. 다음, 역순으로 정돈된 테이블에 따라 찾아야할 트라이폰을 결정하고 위의 과정을 되풀이하여 다음 문장을 선정하다. 이러한 과정을 모든 트라이폰이 선택될 때까지 반복하면, 발화문장을 찾을 수 있다.

<표 1>에서 트라이폰의 발생빈도는 임의의 트라이폰이 선택된 5,760 문장 속에서 몇 번이 출현하는지를 보여주는 통계이다. 즉, 전체 22,194개의 트라이폰중 33.74%인 7,488개는 선택문장에서 단 한번만 출현되었다는 것을 의미한다. 그런데, 단 1회 출현되었다는 것은

* 어절수 10,246,179 음절수 33,275,895

** 구개음화, 끝소리규칙, 자음동화, 경음화, 연음법칙, 자음축약, 음운첨가 등을 고려해 자동 변환

표 1. 문장 선정 결과
Table 1. The result of the target sentences

전체 트라이폰	22,194 개	
문장규모	5,760 문장	
트라이폰 발생 빈도 통계	6회이상	8,063(36.33%)
	5회	821(3.70%)
	4회	1,227(5.53%)
	3회	1,622(7.31%)
	2회	2,973(13.40%)
	1회	7,488(33.74%)

발화시에 1회 출현 트라이폰에 대해 한가지 종류의 운율만을 갖는다는 것을 의미하므로, 다양한 문장의 합성시에 음질저하의 요인이 될 수 있는 것이다. 이미 기술한 바와 같이, 운율을 고려하면 좋겠으나, 문장의 수가 기하급수적으로 증가할 것이므로, 이는 불가능한 것이라고 할 수 있다.

나. 음성녹음 및 분석

본 절에서는 선정된 문장을 녹음하고, 녹음된 음성을 분석하여 잘못 발음된 음성을 수정하는 과정에 대하여 설명하고자 한다.

(1) 음성녹음

문장의 발성은 적절한 아나운서를 통해 해야 하는데, 합성기를 실생활 서비스에 사용한다고 할 때, 그 음질이나 음색이 매우 중요한 요소일 것이다. 본 연구에서는 현재 방송에서 활동하고 있는 여자 아나운서 1명과 남자 아나운서 1명을 선정하여 녹음을 실시하였다. 녹음은 2-채널 녹음을 실시하였다. 즉, 채널 1은 음성을 녹음하며 채널 2는 라링고그래프 신호를 녹음한다.

여기서, 라링고그래프는 성대의 떨림을 잡아주는 기기로서 그 출력은 성대의 떨림, 즉 성대가 열리고 닫히는 과정을 보여주는 신호로서 음성의 피치값을 정확하게 결정하는데 도움을 주는 신호이다. 위의 그림에서 보인 바와 같이 음성과 라링고그래프신호는 2-채널 DAT에 녹음이 되며, 후에 PC 사운드카드의 DAT link를 거쳐 하드디스크에 저장되었다. 본 논문에서 AD 변환은 16kHz 표본추출률 그리고 16bit의 양자화로 이루어졌다.

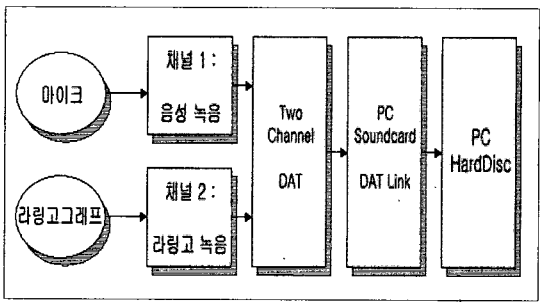

그림 4. 음성 녹음 및 저장 과정
Fig. 4. The procedure of recording and saving

(2) 음성분석

음성은 음가라는 기본정보 외에도 길이, 피치 그리고 세기의 부가적인 운율정보가 첨가된 신호라고 할 수 있다. 물론, 같은 음가라고 할지라도 좌우에 붙은 음운이 무엇인지에 따라 다양한 변이음이 존재한다. 그런데 우리는 이미 발화문장에 대한 음가열을 G2P를 통하여 가지고 있으므로 음성분석은 주로 운율정보의 분석이 주가 된다. 단, G2P의 결과가 항상 정확한 것은 아니므로 이에 대한 수정이 필요한데, 이는 수작업으로 진행한다. 따라서, 음성분석에서 수행되어야 할 내용은 다음과 같은 것들이다. 첫째, 발화문장에 대한 정확한 음가를 G2P결과를 기반으로 하여 수정한다. 둘째, 라링고그래프 신호를 분석하여 발화문장의 피치의 변화를 구한다. 셋째, 발화문장을 음소별로 분할하여 음소의 시간정보를 구한다. 넷째, 음소의 시간정보를 이용하여 주어진 음소의 운율정보를 구한다. 위에서 이미 언급한 바와 같이 발화문장에 대한 음가는 G2P 결과로 출력되는 음소열을 발화문장을 청취하면서 일일이 수작업으로 수정하였다. 선택문장의 발화문장에 대하여 음성합성 DB를 구축하기 위해서는, 우리가 선택한 합성단위가 트라이폰이므로, 음소단위의 분할(레이블링)이 필요하다. 바람직한 것은 음소분할 및 레이블링 작업을 사람이 직접 수작업으로 진행하는 것이다.

그러나, 수작업에 의한 음소 분할 및 레이블링은 다음과 같은 문제점을 갖는다. 첫째, 음소분할 및 레이블링 작업은 스펙트로그램 판독 및 반복되는 듣기 평가를 통해 이루어지므로 매우 지루한 작업일 뿐 아니라 상당한 시간이 소요된다. 둘째, 수작업에 의한 음소분할작업은 높은 수준의 음성학적 지식을 요구하며, 이는 소수의 음성학 전문가에게 의존할 수밖에 없다. 셋째, 음소경계선정을 위한 구체적 기준을 미리 정해 놓더라도 상당부분의 경우 주관적인 판단을 전혀 피할 수 없으며 이에 따르는 음소경계 선정과정에서 일관성이 보장되지 못한다. 전체적으로 수작업에 의한 음소 레이블링은 자동 음소 레이블링에 비하여 보다 정확하다고는 하지만 여러 사람이 작업을 하거나 작업량이 대용량이 될 경우 일관성이 떨어진다. 물론, 수작업을 하기 전 음소 레이

블링의 기준이 정확하게 설정되어 있다면, 이 문제는 다소 줄어들 것이다. 이에 반하여 자동 음소 레이블링은 일관성이 있으며, 물론 그 정확도는 수동 음소 레이블링에 비하여 떨어지게 될 것이다. 이와 같은 어려움을 극복하고 시간의 단축을 위하여 본 연구에서는 음성인식 툴킷은 HTK(hidden Markov model tool kit)을 이용하여 자동 레이블링을 하였다. 한편, HTK를 이용한 자동 레이블링 방법으로 레이블한 음소의 경계는 부정확한 부분들이 종종 발생한다. 따라서, 전문가에 의한 수정이 필요하다. 수동 레이블링은 전문가에 의해 음성의 스펙트로그램 분석을 통하여 수동으로 레이블링 수정 작업을 진행하였다.

다. 음성합성용 트라이폰 DB 구축

트라이폰(Triphone) 음성합성 DB는 합성 단위를 트라이폰으로 하는 기본 음성합성 DB 구조로, 녹음용 문장 150여만 문장으로 이루어진 텍스트 코퍼스에서 추출된 것이다. 모든 트라이폰이 최소 1회 이상 출현한 문장을 자동 검색한 것으로, 긴 문장(80음절 이상)은 80음절 이하로 잘라주고 또 단음절 문장을 따로 추가하는 등의 방법으로 만들어낸 것들이다. 그 후 녹음된 파형을 자동으로 음소 구분하고, 또 다시 수작업을 통해 음소 구분을 수행한 결과를 이용하여 음성 합성 DB를 구현하였다.

(1) 트라이폰 DB의 구조

본 음성 합성 시스템과 음성합성 DB에서 사용되는 음소와 그 표기, 인덱스는 아래와 같다. 코퍼스 기반 음성합성은 '목표단위'와 가장 일치하는 합성 단위를 음성합성 DB로부터 불러들여 이를 접합함으로써 합성음을 만들어 내는 합성방식이다. 이 합성 방식을 사용함에 있어서 고려해야 하는 두 가지 사항은 우선 최상의 '목표단위'를 설정하는 것과, 그 '목표단위'와 가장 잘 일치하는 합성 단위를 음성합성 DB에서 골라내는 일이다.

최상의 목표 단위를 설정하기 위해서는 먼저 음성합성 DB 문장을 발성한 화자의 발성특징을 분석하여 운율 모델로 구축하고, 이를 언어처리 및 운율예측 단계에 적용하여 음의 높낮이 및 음소지속길이 등이 가장 발화자의 발성과 유사하게 처리하는 것이 그 핵심이다.

목표단위를 설정하고 나면 음성합성 DB내에서 각 합성단위에 기록된 운율데이터를 기반으로 하여 목표단위와 가장 유사한 단위를 찾아 접합을 수행하게 된다. 따라서 음성합성 DB에 적재될 각 합성단위는 각각의 운율데이터를 내재하고 있어야 한다.

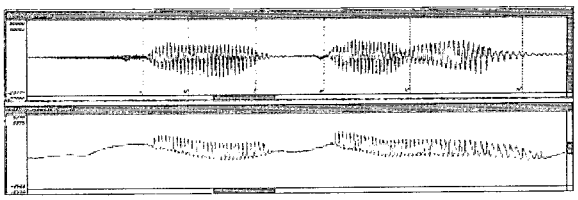

그림 5. 음성파형과 라링고그래프
Fig. 5. The waveform of the speech and the laryngograph

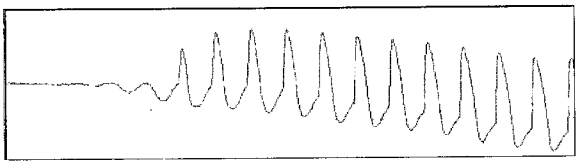

그림 6. 확대한 라링고 신호
Fig. 6. The magnified signal of the laryngograph

운율데이터는 톤의 높낮이, 음소지속길이, 세기의 운율요소를 기본으로 하여 그 밖에 음운환경정보, 음소경계정보, 위치정보 등을 포함한다.

(2) Pitch Contouring

Pitch contouring이란 음성 신호의 Pitch주파수를 음성 신호와 동일한 시간적 그래프로 표현한 것을 말한다. Pitch를 추출하기 위해서는 음성 신호를 그냥 분석할 수도 있으나 보다 정확성을 기하기 위해 라링고 신호를 사용한다. 라링고 신호는 음성을 녹음할 당시 발화자의 성대의 떨림 주기를 측정한 신호로, 역시 음성 신호와 동일한 시간적 그래프이다.

<그림 5>는 '슬픈'이라는 단어를 음성 파형과 라링고 신호를 녹음하여 음성 파일로 만든 것이고, <그림 6>은 그 중에서 일부 라링고 신호를 확대한 것이다. 신호가 위 아래로 복잡하게 올라갔다 내려갔다 하는 부분이 바로 성대가 진동한 부분이며, 피치(Pitch:음높이)가 발생하는 구간이다. 이 신호의 주기를 측정하면 그것이 곧 피치 주기이며, 피치 주기의 역수가 곧 피치 주파수가 되므로, 우리는 이 신호의 주기를 측정해서 이 값을 시간 연속적으로 기록해서 Pitch Contour를 구할 수 있다.

(3) 피치 추출 방법

본 논문에서는 신호의 잡음을 제거하고 라링고 신호의 최고 정점(peak)을 탐색하는 방법을 통해 피치를 검출하였다. 피치를 검출하는 방법으로는 AMDF 및 Autocorelation[1][7][8] 을 이용하는 등의 여러가지 방법이 있을 수 있으나, 라링고 신호는 그 자체가 바로 성대의 떨림을 의미하는 신호이기 때문에 라링고 신호의 진동

주기를 측정하면 곧 이것이 피치로 나타나게 되므로 여기서는 진동 주기를 측정하는 방법에 중점을 두었다. 저주파 잡음을 제거하고 진동의 최고 정점을 두드러지게 하기 위해 고주파 필터(High pass filter)로 먼저 저주파 신호를 걸러준 다음, 미세한 백색 잡음을 제거하기 위해 저주파 필터(low pass filter)를 통과시켰다. 그 후, 진동의 최고 정점을 최대한 도드라지게 한 다음, 진동의 주기를 시간적으로 측정하는 방법으로 Pitch Contour를 완성하게 된다.

(4) PID data 작성

PID는 Pitch, Intensity, Duration을 의미한다. PID data는 정식 트라이폰 DB를 만드는 1차 기초 DB로, 음성 파형과 labelled data, 그리고 앞서 만들어진 pitch data를 이용하여 작성하는데, 그 구성은 <표 2>와 같다.

구성물은 정식 트라이폰 DB의 그것과 비교하여 단지 magnitude 정보, cepstrum 정보만 없을 뿐 모든 것이 동일하다.

(5) 경계 Cepstrum 추출과 VQ

PI2 Data는 PID data에 cepstrum과 magnitude 등의 음소경계 data를 추가하는 data로 정의한다. PI2 Data를 만들기 전에 음소 경계에서 Cepstrum을 추출하여 256개의 대표 벡터를 만드는 과정이 있다. 이것은 음소간 거리값을 추출하는데 Cepstrum거리값을 사용하기 때문이다. 이 값은 PI2 Data의 cepst 구조체에 적용된다.

먼저 모든 음소 경계의 좌우 256 sample로 LPC Cepstrum을 추출하여 Vector Pool에 집어 넣는다. 그 다음 이 Vector Pool을 VQ하여 256개의 대표 Vector를 검출해 낸다. 검출된 대표 Vector들의 각각의 상호 거리값을 계산하여 거리값 테이블을 만들어 저장하면 완성된다.

Vector는 12차 LPC Cepstrum을 사용하고, VQ는 LBG 알고리즘을 사용하였다.[4]

표 2. PID Data의 구성
Table.2. The table of the PID data

변수	Size	내용
name	char [10]	triphone name
sentence_index	short	문장번호
begin_point	int	문장 내 시작 위치
sound_voiced	BOOL	유/무성음 유무
IAP	char	IP, AP 여부
phone_environment	char [4]	앞뒤 음운 환경
duration	struct	음소 지속 구간
intensity	struct	음소 세기
pitch	struct	음소 앞 뒤 및 중간 피치

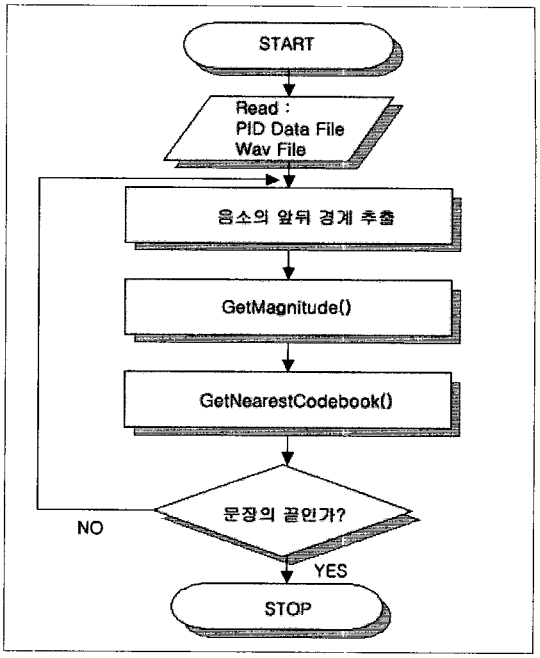

그림 7. PI2 Data 구축 절차
Fig. 7. The collection procedure of PI2 Data

(6) PI2 data 작성

PI2 data를 만드는 절차는 <그림 7>과 같다. PI2 data는 앞서 언급한 트라이폰 DB의 구조와 완전히 동일하다. 즉, PI2 data는 PID data에 Cepst 구조체, Magnitude 구조체 값을 추가하여 만들어내는 데이터 형식이다.

(7) BIN data 작성

BIN data는 최종 합성 DB이다. PID data와 PI2 data는 모두 문장별로 정리된 트라이폰 DB이며, 이것을 트라이폰 name별로 정리한 것이 곧 BIN data이다. PI2 data를 BIN data로 정리하는 데에는 4번의 단계를 거친다. 1단계는 트라이폰 정보 수집 단계로 문장별로 정리된 PI2 data를 모두 읽어서 트라이폰 정보를 수집한다. 2단계는 트라이폰을 정렬하는 단계로 수집된 트라이폰 정보를 트라이폰의 출현 빈도별로 정렬한다. 3단계는 비어있는 BIN파일 생성하는 단계로 트라이폰 수집 후 각 트라이폰의 출현 빈도에 맞게 BIN 파일을 만든다. 이 과정은 처음에 파일을 일괄적으로 만들어서 파일의 조각화 현상을 방지하기 위한 것이다. 4 단계는 BIN Data를 채워 넣는 단계로 PI2 data를 읽어서 해당 트라

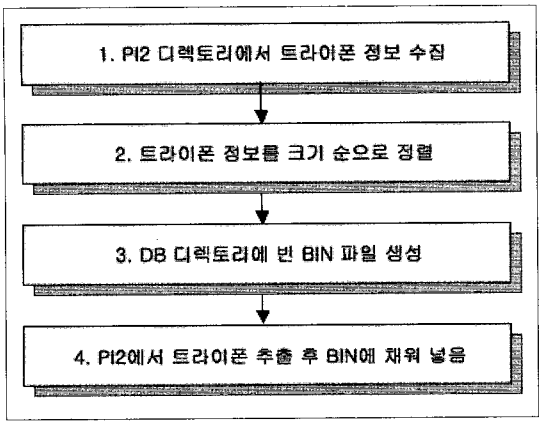

그림 8. BIN Data 구축 절차
Fig. 8. The collection procedure of BIN Data

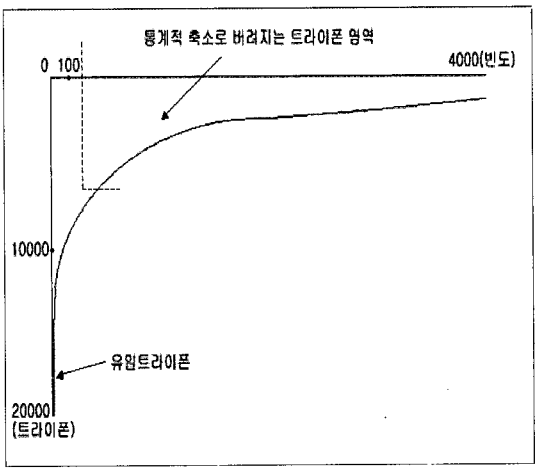

그림 9. 통계적인 트라이폰 축소 영역
Fig. 9. The reduced area of the triphones

이폰을 빈 BIN 파일에 하나씩 채워 넣는 과정이다. 이 단계들을 모두 거쳐야 비로소 음성합성 DB가 완성된다.

3. 음성합성 DB 압축 및 축소

본 논문에서는 음성합성 DB의 용량을 줄이기 위해 음성합성 DB 를 압축 및 축소하였다. 음성합성 DB 압축에 사용된 방식은 범용 압축 방식인 ADPCM 방식이며 압축 후에 통계적 축소를 통해 DB의 크기를 상당히 많이 줄일 수 있었다.

본 논문에서는 우선 1차적으로 ADPCM 방식으로 압축한 트라이폰의 사용 빈도를 기준으로 하여 많은 수의 트라이폰을 제거 하였다.

그림 9와 같이 본 논문에서 사용된 총 트라이폰의 개수는 19,862개이고 하나의 트라이폰이 가장 많이 나타난 빈도는 4,734회였다. 같은 운율 환경을 갖는 트라이폰의 수가 많을 수록 좋지만 동일한 트라이폰으로 나타난 4,734개 중에서 실제로 사용되는 트라이폰의 경우는 수 십 개에 불과하므로 사용되지 않는 트라이폰은 모두 삭제하여도 큰 영향이 없다.

그리고 단 1개만으로 이루어진 유일 트라이폰은 5,834개였다. 1개만으로 이루어진 트라이폰은 삭제할 경우 심각한 합성 음질의 저하를 불러올 수 있으므로 축소의 대상으로 선정하는 것에 주의를 기하였다. 각 트라이폰은 빈도 정보에 의해 정렬한 뒤 기준이 되는 대표 트라이폰과의 운율 거리를 측정하여 가장 유사한 트라이폰부터 삭제해 나가는 방식을 사용하였다. 그리하여 ADPCM 방식으로 1차적으로 압축한 270Mbyte의 11.8%인 32Mbyte와 23.7%인 64Mbyte로 음성합성 DB 크기를 축소했다. 이렇게 축소된 64Mbyte와 32Mbyte 두 종류의 내장형 음성합성 시스템으로 음성 파일을 생성한 뒤 MOS Test를 하였다.

Ⅲ. 실 험

본 논문에서는 100%용량의 서버형 TTS와 약 4%로 용량을 줄인 내장형 A형 TTS, 그리고 약 2%로 용량을 줄인 내장형 B형 TTS를 비교하였다. 20대~30대인 일반인 남녀 각 10명씩 총 20명에게 각각 8문장씩의 합성음을 들려주고 MOS TEST(Mean Opinion Score Test)를 하였다. 이때 피실험자들에게 요구한 평가 기준은 자연성과 명확도의 판단이다.

서버형 TTS의 경우 합성음을 음성 파일 형태로 만들어 들려주었고, 내장형 TTS 시스템의 경우에는 PDA에 포팅을 한 후 PDA 상에서 직접 들려주었다.

MOS Test의 기준은 표 3과 같다.

표 3. MOS Test 기준
Table. 3. The criteria of MOS test

점수	음질	기준 (실제 사람의 음성과 비교했을 때)	
		자연성	명료도
5	Excellent (아주 훌륭함)	자연 음성과 차이 없음	자연 음성과 차이 없음
4	Good (좋음)	자연스러움	명확함
3	Fair (보통)	보통	보통
2	Poor (안 좋음)	부자연스러움	불명확함
1	Unsatisfactory (아주 불만족스러움)	매우 부자연스러움	매우 불명확함

Ⅳ. 결 론

전체 시스템 음질 평가를 위하여 남녀 각각 10명씩

표 4. MOS Test 결과
Table 4. The result of MOS test

서버형 / 내장형	남성 TTS			여성 TTS		
	1,500MB	64MB	32MB	1,500MB	64MB	32MB
MOS	3.8	3.7	3.3	3.25	3.2	2.75

MOS 테스트를 한 결과 다음표와 같은 결과값을 얻을 수 있었다.

<표 4>는 음성합성 DB의 압축과 축소를 통해 각각 전체 1,500Mbyte의 4.2%와 2.1%로 용량을 감축한 내장형 TTS의 MOS Test 결과이다.

남성 TTS의 경우 서버형 TTS의 MOS는 3.8로 양호한 수준이었으며 64Mbyte형은 서버형 보다 0.1 떨어진 3.7, 32Mbyte형은 서버형에 비해 0.5 떨어진 3.3으로 나타났다. 남성 TTS는 32Mbyte형까지 모두 3.0 이상으로 합성 목표를 상회하는 MOS를 받아 압축 후 성능도 보통 이상의 수준으로 나타났다.

여성 TTS의 경우 서버형 TTS의 MOS는 3.25로 나타났으며 이는 남성 TTS에 비해 음질이 많이 떨어진다는 것을 보여주고 있다. 64Mbyte형은 3.2로 합성 목표로 삼은 3.0 이상으로 나타났으나 32Mbyte형은 2.75로 합성 목표에 미달하였다. 여기에서 남성 내장형 TTS가 여성 내장형 TTS에 비해 높은 MOS를 받은 이유는 서버형 자체의 MOS TEST 결과와 관련이 있는 것으로 보인다. 서버형 MOS 결과와 내장형 MOS의 결과는 그 MOS 값의 감소 폭이 유사하게 나타나고 있다.

이상에서 남성 TTS와 여성 TTS에서 공통적으로 관찰된 사항은 서버형 TTS를 64Mbyte로 줄인 경우 서버형 TTS와 내장형 TTS 사이에 MOS의 차이가 그리 크지 않다는 것이다. 64Mbyte형에서 남성 TTS는 3.8에서 3.7로 0.1이 줄었고 여성 TTS의 경우 3.25에서 3.2로 0.05가 줄었다. 즉, 1,500Mbyte를 64Mbyte로 줄인 경우 서버형 TTS에 비해 음질의 저하가 있기는 하지만 그 저하의 폭이 적다는 점을 알 수 있다. 그러나 1,500Mbyte를 32Mbyte로 줄인 경우 남성 TTS는 3.8에서 3.3으로 0.5가 줄었고 여성 TTS는 3.25에서 2.75로 역시 0.5가 줄어들어 음질 저하의 폭이 컸다는 것을 보여준다. 이는 32Mbyte로 감축되는 과정에서 트라이폰의 손실이 급격히 커진다는 것을 의미하는 것이며 이를 보완하기 위한 방안이 필요하다는 것을 말해주는 것이라고 할 수 있다. 따라서 32Mbyte형이나 또는 그 이하의 용량으로 TTS의 용량을 감축하기 위해서는 음성합성 DB에 포함되지 않은 트라이폰이 호출되었을 경우 이를 처리할 수 있는 대체 트라이폰 선택 기술 등 보완 기술에 대한 연구가 이루어져야할 것으로 보인다.

참 고 문 헌

[1] X. Huang, A. Acero and H. Hon, *Spoken Language Processing,* Prentice Hall PTR, pp. 763, 2001.

[2] 김병창, 이근배, "자연어 처리 기반 한국어 TTS 시스템 구현". 말소리 46호, 대한음성학회.

[3] 신지영, *말소리의 이해,* 한국문화사, 2001.

[4] 김장한, VQ의 코드북 생성을 위한 LBG 알고리즘의 개선에 관한 연구, 한국통신학회 논문지 제25권 1호, pp.48~55, 2000.

[5] 이상호, "미등록어를 고려한 한국어 품사 태깅 시스템 구현", 한국과학기술원 석사논문, 1995.

[6] 장경애, 정민화, 김재인, 구명완(2002), "코퍼스기반 음성합성기의 데이터베이스 감축 방안", 말소리 44호, 대한음성학회.

[7] L. R. Rabiner, R. W. Schafer, *Digital Processing Of Speech Signals,* Prentice Hall PTR, 1978.

[8] L. R. Rabiner, B. H. Jung, *Fundamentals Of Speech Recognition,* Prentice Hall PTR, 1993.

[9] Richard Sproat, *Multilingual Text-to-Speech Synthesis,* Kluwer Academic Publishers, 1998.

[10] Manfred R. Schroeder, Computer Speech, Springer, 1999.

[11] Jonathan Allen, M. Sharon Hunnicutt, Dennis Klatt, *From text to speech,* Cambridge University Press, 1987.

저 자 소 개

권 오 일(평생회원)
1987년 3월~1991년 2월 고려대학교 전자공학과 학사
1991년 3월~1993년 2월 고려대학교 전자공학과 석사
1993년 3월~1996년 8월 고려대학교 전자공학과 박사
2000년 3월~현재 현대오토넷(주) 차장
1996년 8월~2000년 3월 현대전자산업(주) 차장
<주관심분야 : 음성인식,합성, 임베디드시스템>