OCR Based Speech Synthesis System Using LabVIEW

A thesis
Submitted towards the partial fulfillment of
the requirements of the degree of

Master of Engineering
In
Electronic Instrumentation and Control Engineering

Submitted By

Rajiv Kumar Yadav
Roll No-80751017

Under the esteemed guidance of

Sunil Kumar Singla
Sr.Lecturer,EIED

DEPARTMENT OF ELECTRICAL AND INSTRUMENTATION ENGINEERING
THAPAR UNIVERSITY
PATIALA –147004.

July - 2009

DEDICATED

TO

MY PARENTS

CERTIFICATE

This is to certify that my work presented in this thesis entitled ***"OCR Based Speech Synthesis System Using LabVIEW"*** *submitted in partial fulfillment of the requirement for the award of the degree of* ***Master of Engineering in Electronic Instrumentation and Control Engineering*** *at* ***Thapar University, Patiala****, is an original record under supervision and guidance of* ***Mr. Sunil Kumar Singla (Sr. Lecturer).*** *The matter embodied in this report has not been submitted anywhere for the award of any degree.*

Date: 14-07-2009

Yadav

(Rajiv Kumar Yadav)

Roll No - 80751017

It is certified that the above statement made by the student is correct to the best of our knowledge and belief.

(Sunil Kumar Singla)
Sr. Lecturer, EIED
(Supervisor)
Thapar University, Patiala

Countersigned By:

S. Ghosh 14/7/09

(Dr. Smarajit Ghosh)
Professor & Head, EIED
Thapar University, Patiala

16/7

(Dr. R. K. Sharma)
Dean of Academic Affairs
Thapar University, Patiala

ACKNOWLEDGEMENT

The real spirit of achieving a goal is through the way of excellence and austerous discipline. I would have never succeeded in completing my task without the cooperation, encouragement and help provided to me by various personalities.

First of all, I render my gratitude to the ALMIGHTY who bestowed self-confidence, ability and strength in me to complete this work. Without his grace this would never come to be today's reality.

With deep sense of gratitude I express my sincere thanks to my esteemed and worthy Supervisor **Mr. Sunil Kumar Singla** in the Department of Electrical and Instrumentation Engineering for his valuable guidance in carrying out this work under his effective supervision, encouragement, enlightenment and cooperation. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. His feedback and editorial comments were also invaluable for writing of this thesis.

I shall be failing in my duties if I do not express my deep sense of gratitude towards **Dr. Smarajit Ghosh,** Professor and Head of Electrical and Instrumentation Department who has been a constant source of inspiration for me throughout this work.

I am grateful to **Dr. R.K. Sharma**, Dean of Academic Affair for his constant encouragement that was of great importance in the completion of the thesis.

I extend my thanks to **Dr. K.K. Raina**, Deputy Director, **Dr. Abhijit Mukherjee**, Director, Thapar University for their valuable support that made me consistent performer.

I am also thankful to all the staff members of the Department for their full cooperation and help. Share my greatest thanks are to all who wished me success especially my parents, my batch mates Manish Sharma, Ankit Sharma, Vivek Chaudhary, my juniors Rahul Dev Nigam, Yatendra Rawal & Ankit Sharma whose support and care makes me.

Place: TU, Patiala

Rajiv Kumar Yadav

Roll No. 80751017

ABSTRACT

Knowledge extraction by just listening to sounds is a distinctive property. Although text can be a medium of communication but speech signal is more effective means of communication than text. In this thesis work OCR Based Speech Synthesis System has been discussed using LabVIEW 7.1.

Although a lot of work has been done in the field OCR and Speech Synthesis individually, but it is fist OCR based Speech Synthesis System using LabVIEW. This Thesis contains two part optical character recognition and text to speech conversion. The OCR software is developed with IMAQ Vision for LabVIEW software- developing tool and it uses a commercial digital scanner as image acquisition device. IMAQ Vision OCR software is a PC-based character recognition tool for use with IMAQ Vision for LabVIEW. IMAQ Vision OCR is designed for high-speed and reliable reading performance, even with poor image quality resulting from varying lighting conditions and print quality. For speech synthesis in LabVIEW the ACTIVE X sub pallet in Communication pallet and its functions to exchange data between applications. ActiveX/COM refers to the process of controlling one program from another via ActiveX. Like networking, one program acts as the client and the other as the server LabVIEW supports ActiveX automation both as the client and the server. Both program, client and server, exist independent of each other but are able to share information. The client communicates with the ActiveX objects that the server opens to allow the sharing of information. The automation client can access the object's properties and methods. Properties are attributes of an object.

This thesis aims to study the OCR and speech synthesis technology and to develop a cost effective, user friendly OCR Based Speech Synthesis System using Laboratory virtual instruments engineering workbench (LabVIEW) graphical programming language.

LIST OF CONTENTS

LIST OF FIGURE

LIST OF TABLES

Chapter 1

Introduction

1.1 Introduction

Machine replication of human functions, like reading, is an ancient dream. However, over the last five decades, machine reading has grown from a dream to reality. Character recognition or optical character recognition (OCR), is the process of converting scanned images of machine printed or handwritten text (numerals, letters, and symbols), into a computer format text (such as ASCII). Optical character recognition has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence. Many commercial systems for performing OCR exist for a variety of applications. Speech is probably the most efficient medium for communication between humans. A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud, whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system [1].

1.1.1 OCR

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic translation of images of handwritten, typewritten or printed text (usually captured by a scanner) into machine-editable text [2]. Optical character recognition belongs to the family of techniques performing automatic identification. These different techniques are discussed below and define OCR's position among them.

1.1.2 Automatic Identification

The traditional way of entering data into a computer is through the keyboard. However this is neither always the best nor the most efficient solution. In many cases automatic identification may be an alternative. Various technologies for automatic identification

exist, and they cover needs for different areas of application [3]. Below a brief overview of the different technologies and their applications is given.

1.1.2.1 Speech recognition

In systems for speech recognition, spoken inputs from a predefined library of words are recognized. Such systems should be speaker-independent and may be used for instance for reservations or ordering of goods by telephone. Another kind of such systems are those used to recognize the speaker, rather than the words, for identification.

1.1.2.2 Radio frequency

This kind of identification is used for instance in connection with toll roads for identification of cars. Special equipment on the car emits the information. The identification is efficient, but special equipment is needed both to send and to read the information. The information is also inaccessible to humans [4].

1.1.2.3 Vision systems

By the use of a TV-camera objects may be identified by their shape or size. This approach may for instance be used in automatons for recirculation of bottles. The type of bottle must be recognized, as the amount reimbursed for a bottle depends on its type.

1.1.2.4 Magnetic stripe

Information contained in magnetic stripe is widely used on credit cards etc. Quite a large amount of information can be stored on the magnetic stripe, but specially designed readers are required and the information can not be read by humans.

1.1.2.5 Barcode Recognition

A barcode is a machine-readable representation of information. Barcodes can be read by optical scanners called barcode readers or scanned from an image using software. A 2D barcode is similar to a linear, one-dimensional barcode, but has more data representation capability. Figure of 1-D barcode and 2-D barcode are given bellow [5].

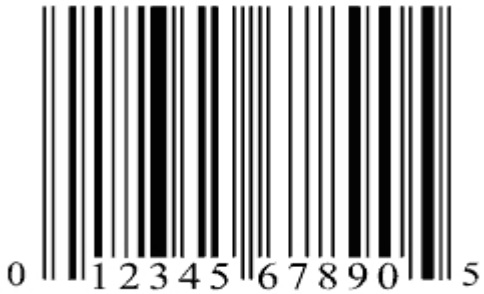


Figure 1.1 1-D barcode The Gettysburg Address (UPC).

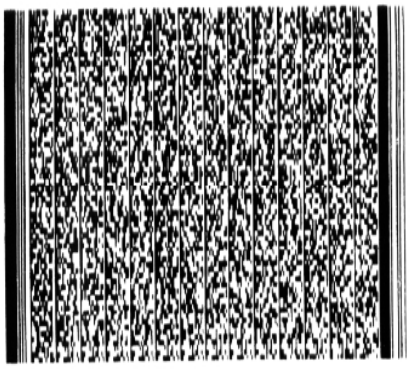


Figure 1.2 2-D barcode Universal Product Code.

1.1.2.6 Magnetic Ink Character Recognition (MICR)

Printing in magnetic ink is mainly used within bank applications. The characters are written in ink that contains finely ground magnetic material and they are written in stylized fonts which are specifically designed for the application. Before the characters are read, the ink is exposed to a magnetic field. This process accentuates each character and helps simplify the detection. The characters are read by interpreting the waveform obtained when scanning the characters horizontally. Each character is designed to have its own specific waveform. Although designed for machine reading, the characters are still readable to humans. However, the reading is dependent on the characters being printed with magnetic ink [6].

1.1.2.7 Optical Mark Recognition (OMR)

OMR technology detects the existence of a mark, not its shape. OMR forms usually contain small ovals, referred to as 'bubbles,' or check boxes that the respondent fills in. OMR cannot recognize alphabetic or numeric characters. OMR is the fastest and most accurate of the data collection technologies. It is also relatively user-friendly. The accuracy of OMR is a result of precise measurement of the darkness of a mark, and the sophisticated mark discrimination algorithms for determining whether what is detected is an erasure or a mark [7].

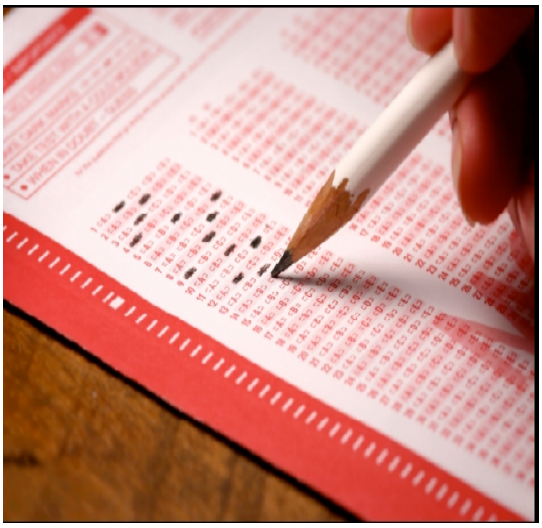


Figure 1.3 The College Board SAT uses OMR technology.

1.1.2.8 Optical Character Recognition

Optical character recognition is needed when the information should be readable both to humans and to a machine and alternative inputs cannot be predefined. In comparison with the other techniques for automatic identification, optical character recognition is unique in that it does not require control of the process that produces the information.

1.1.3 Speech Synthesis

Speech synthesis is the artificial production of human speech. Synthesizing is the very effective process of generating speech waveforms using machines based on the phonetical transcription of the message. Recent progress in speech synthesis has produced

synthesizers with very high intelligibility but the sound quality and naturalness still remains a major problem.

1.1.3.1 Phonetics and Theory of Speech Production

Speech processing and language technology contains lots of special concepts and terminology. To understand how different speech synthesis and analysis methods work one must have some knowledge of speech production, articulatory phonetics, and some other related terminology. The basic theories related to these topics are described below.

1.1.3.1.1 Representation and Analysis of Speech Signals

Continuous speech is a set of complicated audio signals which makes producing them artificially difficult. Speech signals are usually considered as voiced or unvoiced, but in some cases they are something between these two. Voiced sounds consist of fundamental frequency (F0) and its harmonic components produced by vocal cords (vocal folds). The vocal tract modifies this excitation signal causing formant (pole) and sometimes antiformant (zero) frequencies [8]. Each formant frequency has also amplitude and bandwidth and it may be sometimes difficult to define some of these parameters correctly. The fundamental frequency and formant frequencies are probably the most important concepts in speech synthesis and also in speech processing in general.

With purely unvoiced sounds, there is no fundamental frequency in excitation signal and therefore no harmonic structure either and the excitation can be considered as white noise. The airflow is forced through a vocal tract constriction which can occur in several places between glottis and mouth. Some sounds are produced with complete stoppage of airflow followed by a sudden release, producing an impulsive turbulent excitation often followed by a more protracted turbulent excitation [9]. Unvoiced sounds are also usually more silent and less steady than voiced ones.

Speech signals of the three vowels (/a/ /i/ /u/) are presented in time- and frequency domain in Figure: 1.4. The fundamental frequency is about 100 Hz in all cases and the formant frequencies F1, F2, and F3 with vowel /a/ are approximately 600 Hz, 1000 Hz, and 2500 Hz respectively. With vowel /i/ the first three formants are 200 Hz, 2300 Hz, and 3000 Hz, and with /u/ 300 Hz, 600 Hz, and 2300 Hz. The harmonic structure of the excitation is also easy to perceive from frequency domain presentation.

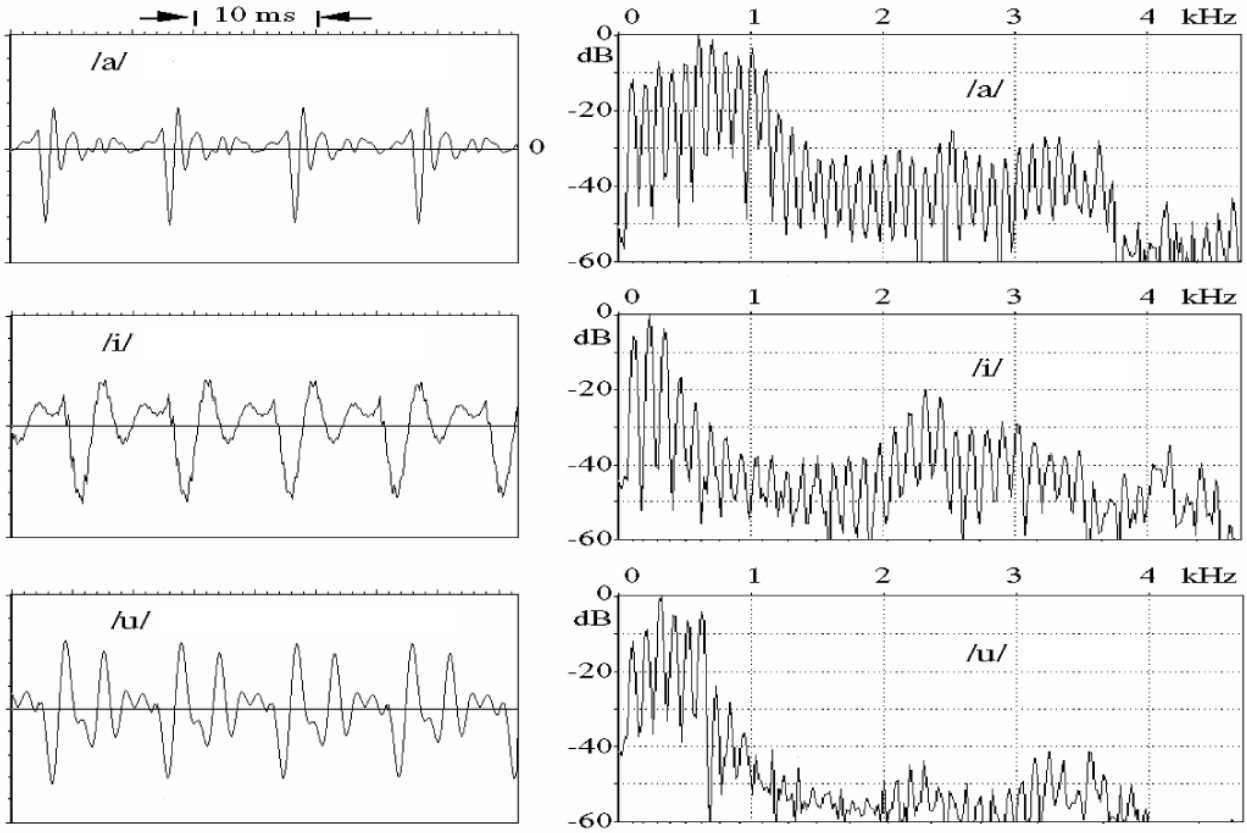


Figure 1.4 The time- and frequency-domain presentation of vowels /a/, /i/, and /u/.

For determining the fundamental frequency or pitch of speech, for example a method called cepstral analysis may be used [9]. Cepstrum is obtained by first windowing and making Discrete Fourier Transform (DFT) for the signal and then logaritmizing power spectrum and finally transforming it back to the time-domain by Inverse Discrete Fourier Transform (IDFT). The procedure is shown in Figure 1.5.

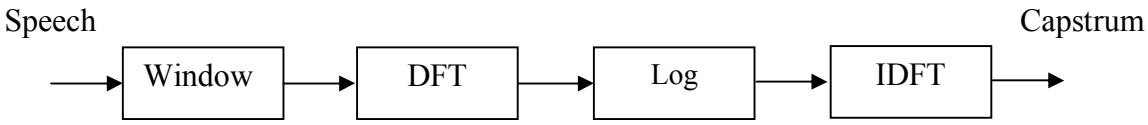


Figure 1.5 Cepstral analyses.

Fundamental frequency or intonation contour over the sentence is important for correct prosody and natural sounding speech. The different contours are usually analysed

from natural speech in specific situations and with specific speaker characteristics and then applied to rules to generate the synthetic speech. The fundamental frequency contour can be viewed as the composite set of hierarchical patterns shown in Figure 1.6. The overall contour is generated by the superposition of these patterns [10].

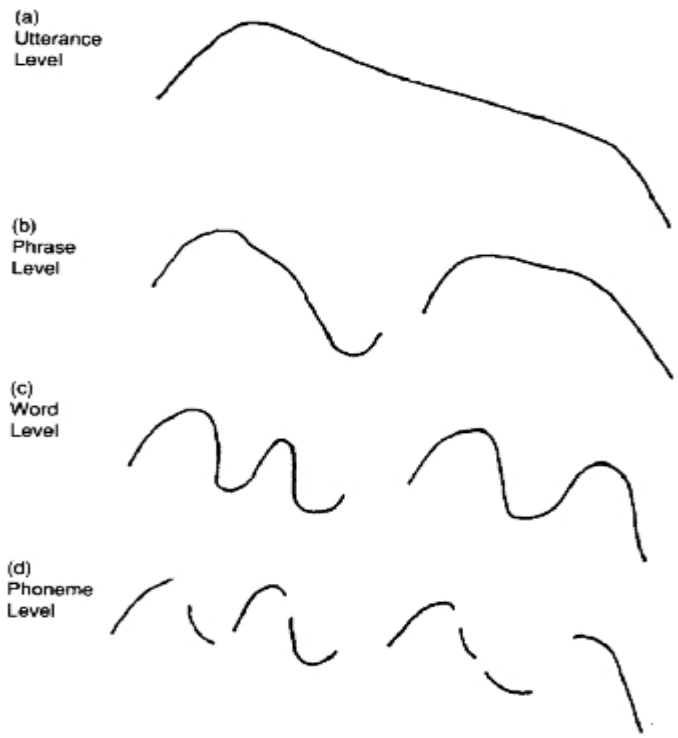


Figure 1.6 Hierarchical levels of fundamental frequency.

1.1.3.1.2 Speech Production

Human speech is produced by vocal organs presented in Figure 1.7. The main energy source is the lungs with the diaphragm. When speaking, the air flow is forced through the glottis between the vocal cords and the larynx to the three main cavities of the vocal tract, the pharynx and the oral and nasal cavities. From the oral and nasal cavities the air flow exits through the nose and mouth, respectively. The V-shaped opening between the vocal cords, called the glottis, is the most important sound source in the vocal system. The vocal cords may act in several different ways during speech. The most important function is to modulate the air flow by rapidly opening and closing, causing buzzing sound from which

vowels and voiced consonants are produced. The fundamental frequency of vibration depends on the mass and tension and is about 110 Hz, 200 Hz, and 300 Hz with men, women, and children, respectively. With stop consonants the vocal cords may act suddenly from a completely closed position, in which they cut the air flow completely, to totally open position producing a light cough or a glottal stop. On the other hand, with unvoiced consonants, such as /s/ or /f/, they may be completely open. An intermediate position may also occur with for example phonemes like /h/.

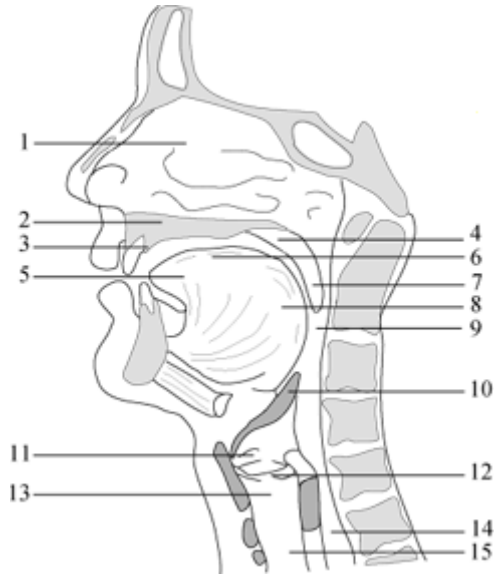


Figure 1.7 The human vocal organs. (1) Nasal cavity, (2) Hard palate, (3) Alveoral ridge, (4) Soft palate (Velum), (5) Tip of the tongue (Apex), (6) Dorsum, (7) Uvula, (8) Radix, (9) Pharynx, (10) Epiglottis, (11) False vocal cords, (12) Vocal cords, (13) Larynx, (14) Esophageus, and (15) Trachea.

The pharynx connects the larynx to the oral cavity. It has almost fixed dimensions, but its length may be changed slightly by raising or lowering the larynx at one end and the soft palate at the other end. The soft palate also isolates or connects the route from the nasal cavity to the pharynx. At the bottom of the pharynx are the epiglottis and false vocal cords to prevent food reaching the larynx and to isolate the esophagus acoustically from the vocal tract. The epiglottis, the false vocal cords and the vocal cords are closed during swallowing and open during normal breathing.

The oral cavity is one of the most important parts of the vocal tract. Its size, shape and acoustics can be varied by the movements of the palate, the tongue, the lips, the cheeks and the teeth. Especially the tongue is very flexible, the tip and the edges can be moved independently and the entire tongue can move forward, backward, up and down. The lips control the size and shape of the mouth opening through which speech sound is radiated. Unlike the oral cavity, the nasal cavity has fixed dimensions and shape. Its length is about 12 cm and volume 60 cm^3. The air stream to the nasal cavity is controlled by the soft palate.

From technical point of view, the vocal system may be considered as a single acoustic tube between the glottis and mouth. Glottal excited vocal tract may be then approximated as a straight pipe closed at the vocal cords where the acoustical impedance $Z_g = \infty$ and open at the mouth ($Z_m = 0$) [11, 12]. In this case the volume-velocity transfer function of vocal tract is

$$V(w) = \frac{Z_m}{Z_g} = \frac{U_m}{U_g} = \frac{1}{\cos\left(\frac{wl}{c}\right)} \tag{1.1}$$

Where l is the length of the tube, ω is radian frequency and c is sound velocity. The denominator is zero at frequencies, $F_i = w_i/2\pi \ (i = 1,2,3 \ldots\ldots)$ where

$$\frac{w_i l}{c} = (2i-1)\frac{\pi}{2}, \text{ and } F_i = \frac{(2i-1)}{4l} \tag{1.2}$$

If l=17 cm, $V(\omega)$ is infinite at frequencies F_i = 500, 1500, 2500, Hz which means resonances every 1 kHz starting at 500 Hz. If the length l is other than 17 cm, the frequencies F_i will be scaled by factor $17/l$ so the vocal tract may be approximated with two or three sections of tube where the areas of adjacent sections are quite different and resonances can be associated within individual cavities. Vowels can be approximated with a two-tube model presented on the left in Figure 1.8. For example, with vowel /a/ the narrower tube represents the pharynx opening into wider tube representing the oral cavity. If assumed that both tubes have an equal length of 8.5 cm, formants occur at twice the frequencies noted earlier for a single tube. Due to acoustic coupling, formants do not

approach each other by less than 200 Hz so formants F1 and F2 for /a/ are not both at 1000 Hz, but rather 900 Hz and 1100 Hz, respectively [12].

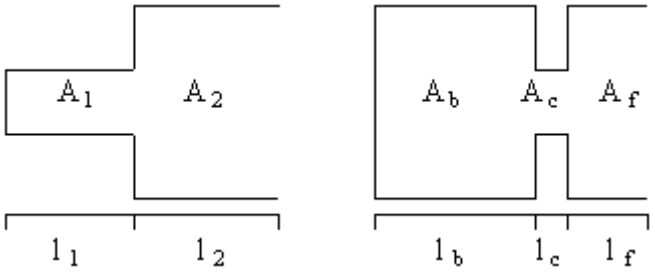


Figure 1.8 Example of two- and three-tube models for the vocal tract.

Consonants can be approximated similarly with a three-tube model shown on the right in Figure 1.8., where the narrow middle tube models the vocal tract constriction. The back and middle tubes are half-wavelength resonators and the front tube is a quarter-wavelength resonator with resonances

$$\frac{ci}{2l_b}, \frac{ci}{2l_c}, \frac{c(2i-1)}{4l_f}, \text{ for } i = 1, 2, 3, \qquad (1.3)$$

Where l_b, l_c, and l_f are the length of the back, center, and front tube, respectively with the typical constriction length of 3 cm the resonances occur at multiples of 5333 Hz and can be ignored in applications that use less than 5 kHz bandwidth [12].

The excitation signal may be modeled with a two-mass model of the vocal cords which consists of two masses coupled with a spring and connected to the larynx by strings and dampers [13].

1.1.3.1.3 Phonetics

In most languages the written text does not correspond to its pronunciation so that in order to describe correct pronunciation some kind of symbolic presentation is needed. Every language has a different phonetic alphabet and a different set of possible phonemes and their combinations. The number of phonetic symbols is between 20 and 60 in each

language [12]. A set of phonemes can be defined as the minimum number of symbols needed to describe every possible word in a language. In English there are about 40 phonemes [14, 15]. Due to complexity and different kind of definitions, the number of phonemes in English and most of the other languages cannot be defined exactly.

Phonemes are abstract units and their pronunciation depends on contextual effects, speaker's characteristics, and emotions. During continuous speech, the articulatory movements depend on the preceding and the following phonemes. The articulators are in different position depending on the preceding one and they are preparing to the following phoneme in advance. This causes some variations on how the individual phoneme is pronounced. These variations are called allophones which are the subset of phonemes and the effect is known as co-articulation. For example, a word *lice* contains a light /l/ and *small* contains a dark /l/. These l's are the same phoneme but different allophones and have different vocal tract configurations. Another reason why the phonetic representation is not perfect is that the speech signal is always continuous and phonetic notation is always discrete [8]. Different emotions and speaker characteristics are also impossible to describe with phonemes so the unit called phone is usually defined as an acoustic realization of a phoneme [15].

The phonetic alphabet is usually divided in two main categories, vowels and consonants. Vowels are always voiced sounds and they are produced with the vocal cords in vibration, while consonants may be either voiced or unvoiced. Vowels have considerably higher amplitude than consonants and they are also more stable and easier to analyze and describe acoustically. Because consonants involve very rapid changes they are more difficult to synthesize properly. The articulatory phonetics in English and Finnish are described more closely in the next section of this chapter.

Some efforts to construct language-independent phonemic alphabets were made during last decades. One of the best known is perhaps IPA (International Phonetic Alphabets) which consists of a huge set of symbols for phonemes, suprasegmentals, tones/word accent contours, and diacritics. For example, there are over twenty symbols for only fricative consonants (IPA 1998). Complexity and the use of Greek symbols makes IPA alphabet quite unsuitable for computers which usually requires standard ASCII as input. Another such kind of phonetic set is SAMPA (Speech Assessment Methods -

Phonetic Alphabet) which is designed to map IPA symbols to 7-bit printable ASCII characters. In SAMPA system, the alphabets for each language are designed individually. Originally it covered European Communities languages, but the objective is to make it possible to produce a machine-readable phonetic transcription for every known human language. Alphabet known as Worldbet is another ASCII presentation for IPA symbols which is very similar to SAMPA [16]. American linguists have developed the Arpabet phoneme alphabet to represent American English phonemes using normal ASCII characters. For example a phonetic representation in DECtalk system is based on IPA and Arpabet with some modifications and additional characters [17]. Few examples of different phonetic notations are given in Table 1.1.

Table: 1.1 Examples of different phonetic notations.

IPA	IPA-ASCII	SAMPA	DECtalk	Example
i	i	i:	iy	beet
I	I	I	ih	bit
ɛ	E	e	ey	bet
æ	&	{	ae	at
ə	@	@	ax	about
ʌ	V	V	ah	but

Several other phonetic representations and alphabets are used in present systems. For example MITalk uses a set of almost 60 two-character symbols for describing phonetic segments in it and it is quite common that synthesis systems use the alphabet of their own. There is still no single generally accepted phonetic alphabet.

1.1.3.1.3.1 English Articulatory Phonetics

Unlike in Finnish articulatory phonetics, discussed bellow, the number of phonetic symbols used in English varies by different kind of definitions. Usually there are about ten to fifteen vowels and about twenty to twenty-five consonants.

English vowels may be classified by the manner or place of articulation (front-back) and by the shape of the mouth (open-close). Main vowels in English and their classification are described in Figure 1.9 below. Sometimes also some diphthongs like /ou/ in *tone* or /ei/ in *take* are described separately.

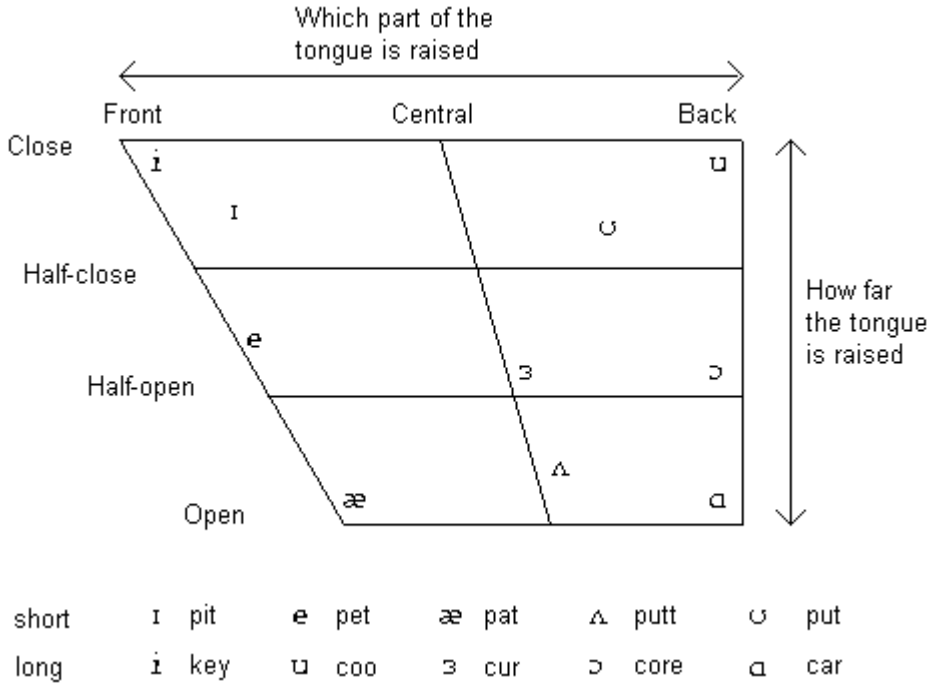


Figure 1.9 The classification of the main vowels in English.

English consonants may be classified by the manner of articulation as plosives, fricatives, nasals, liquids, and semivowels [18]. Plosives are known also as stop consonants. Liquids and semivowels are also defined in some publications as approximants and laterals. Further classification may be made by the place of articulation as labials (lips), dentals (teeth), alveolars (gums), palatals (palate), velars (soft palate), glottal (glottis), and labiodentals (lips and teeth). Classification of English consonants is summarized in Figure 1.10.

place / manner	labial	labio-dental	dental	alveolar	palate-alveoral	palatal	velar	glottal
plosive	p b			t d			k g	
fricative		f v	q ð	s z	ʒ ʃ			h
nasal	m			n			ŋ	
liquid				r l				
semivowel	w					j		

Figure 1.10 Classification of English consonants.

Finally, consonants may be classified as voiced and unvoiced. Voiced consonants are:

/b/, /d/, /g/, /v/, /z/, /ʒ/, /ð/, /l/, /r/, /j/,

Others are unvoiced.

1.1.3.1.3.2 Finnish Articulatory Phonetics

There are eight vowels in Finnish. These vowels can be divided into different categories depending how they are formulated: Front/back position of tongue, wideness/roundness of the constriction position, place of the tongue (high or low), and how open or close the mouth is during articulation. Finnish vowels and their categorization are summarized in Figure 1.11.

Vowels		front		back	
		wide	round	wide	round
Close	high	**i**	**y**		**u**
Close-mid / Open-mid	mid	**e**	**ö**		**o**
Open	low	**ä**		**a**	

Figure 1.11 Classification of Finnish vowels.

Finnish consonants can be divided into the following categories depending on the place and the manner of articulation:

1. Plosives or stop consonants: /k, p, t, g, b, d/. The vocal tract is closed causing stop or attenuated sound. When the tract reopens, it causes noise-like, impulse-like or burst sound.
2. Fricatives: /f, h, s/. The vocal tract is constricted in some place so the turbulent air flow causes noise which is modified by the vocal tract resonances. Finnish fricatives are unvoiced.
3. Nasals: /n, m, ng /. The vocal tract is closed but the velum opens a route to the nasal cavity. The generated voiced sound is affected by both vocal and nasal tract.
4. Tremulants: /r/. Top of the tongue is vibrating quickly (20-25 Hz) against the alveoral ridge causing voiced sound with an effect like amplitude modulation.
5. Laterals: /l/. The top of the tongue closes the vocal tract leaving a side route for the air flow.
6. Semivowels: /j, v/. Semivowels are almost like vowels, but they are more unstable than and not as context-free as normal vowels.

The consonant categories are summarized in Figure 1.12. For example, for phoneme /p/, the categorization will be unvoiced bilabial-plosive.

Consonants	labial		dental alveoral			palatal	velar	laryng.
	bi-lab.	labio-dent.	pro	medio	post			
plosive (tenuis)	p		t				k	
plosive (media)	b			d			g	
fricative (sibilants)			s					
fricative (spirants)		f						h
nasal	m		n				ŋ	
tremulant			r					
lateral			l					
semivowel		v				j		

Figure 1.12 Classification of Finnish consonants.

When synthesizing consonants, better results may be achieved by synthesizing these six consonant groups with separate methods because of different acoustic characteristics. Especially the tremulant /r/ needs a special attention.

1.2 Problem Formulation

Voice output of printed or hand written text produced by OCR system with Speech synthesis gives very effective medium of communication. For some application speech (voice) communication is more useful than text. So, we have chosen to develop OCR based text to speech system using LabVIEW. The main objective of this report is to:

1. Study Optical character recognition technology,
2. Study the speech synthesis technology
3. Develop Optical character recognition using LabVIEW software
4. Develop text to speech module using LabVIEW software
5. Combine OCR and Text to speech module to obtain the desired result.

Chapter 2

Literature Survey

2.1 Introduction

Methodically, character recognition is a subset of the pattern recognition area. However, it was character recognition that gave the incentives for making pattern recognition and image analysis matured fields of science. After character recognition these character are converted into speech. Speech is the vocalization form of human communication. Speech communication is more effective medium than text communication medium in many real-world applications.

2.2 History of OCR

To replicate the human functions by machines, making the machine able to perform tasks like reading is an ancient dream. The origins of character recognition can actually be found back in 1870. This was the year that C.R.Carey of Boston Massachusetts invented the retina scanner which was an image transmission system using a mosaic of photocells. Two decades later the Polish P.Nipkow invented the sequential scanner which was a major breakthrough both for modern television and reading machines [19].

During the first decades of the 19'th century several attempts were made to develop devices to aid the blind through experiments with OCR. However, the modern version of OCR did not appear until the middle of the 1940's with the development of the digital computer. The motivation for development from then on, was the possible applications within the business world.

2.2.1 The Start of OCR

By 1950 the technological revolution was moving forward at a high speed, and electronic data processing was becoming an important field. Data entry was performed through

punched cards and a cost-effective way of handling the increasing amount of data was needed. At the same time the technology for machine reading was becoming sufficiently mature for application, and by the middle of the 1950's OCR machines became commercially available [3].

The first true OCR reading machine was installed at Reader's Digest in 1954. This equipment was used to convert typewritten sales reports into punched cards for input to the computer.

2.2.2 First Generation OCR

The commercial OCR systems appearing in the period from 1960 to 1965 may be called the first generation of OCR. This generation of OCR machines were mainly characterized by the constrained letter shapes read. The symbols were specially designed for machine reading, and the first ones did not even look very natural. With time multifont machines started to appear, which could read up to ten different fonts. The number of fonts were limited by the pattern recognition method applied, template matching, which compares the character image with a library of prototype images for each character of each font [20].

2.2.3 Second Generation OCR

The reading machines of the second generation appeared in the middle of the 1960's and early 1970's. These systems were able to recognize regular machine printed characters and also had hand-printed character recognition capabilities. When hand-printed characters were considered, the character set was constrained to numerals and a few letters and symbols.

The first and famous system of this kind was the IBM 1287, which was exhibited at the World Fair in New York in 1965. Also, in this period Toshiba developed the first automatic letter sorting machine for postal code numbers and Hitachi made the first OCR machine for high performance and low cost.

In this period significant work was done in the area of standardization. In 1966, a thorough study of OCR requirements was completed and an American standard OCR character set was defined; OCR-A. This font was highly stylized and designed to facilitate optical recognition, although still readable to humans. A European font was also designed,

OCR-B, which had more natural fonts than the American standard. Some attempts were made to merge the two fonts into one standard, but instead machines being able to read both standards appeared [4].

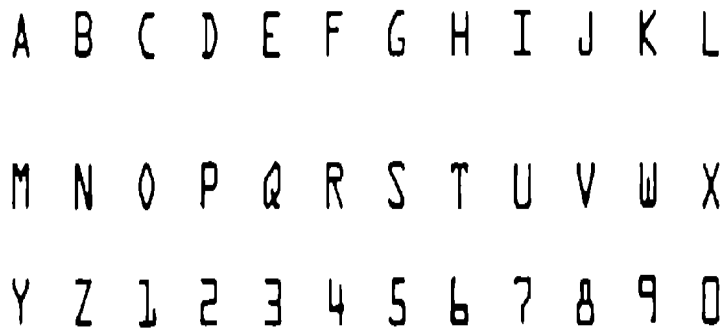


Figure 2.1 OCR-A.

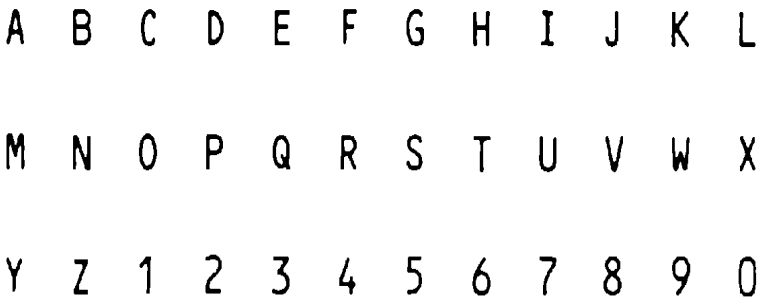


Figure 2.2 OCR-B.

2.2.4 Third Generation OCR

For the third generation of OCR systems, appearing in the middle of the 1970's, the challenge was documents of poor quality and large printed and hand-written character sets. Low cost and high performance were also important objectives, which were helped by the dramatic advances in hardware technology [21].

Although more sophisticated OCR-machines started to appear at the market simple OCR devices were still very useful. In the period before the personal computers and laser printers started to dominate the area of text production, typing was a special niche for OCR The uniform print spacing and small number of fonts made simply designed OCR devices

very useful. Rough drafts could be created on ordinary typewriters and fed into the computer through an OCR device for final editing. In these way word processors, which were an expensive resource at this time, could support several people and the costs for equipment could be cut [4].

2.2.5 OCR Today

Although, OCR machines became commercially available already in the 1950's, only a few thousand systems had been sold worldwide up to 1986. The main reason for this was the cost of the systems. However, as hardware was getting cheaper, and OCR systems started to become available as software packages, the sale increased considerably [22]. Today a few thousand is the number of systems sold every week, and the cost of an omnifont OCR has dropped with a factor of ten every other year for the last 6 years.

2.3 Components of an OCR System

A typical OCR system consists of several components. In figure 2.3 a common setup is illustrated. The first step in the process is to digitize the analog document using a digital scanner. Then extracted text will be pre-processed (binarization or thresholding), when the regions containing text are located, each symbol is extracted through a segmentation process.

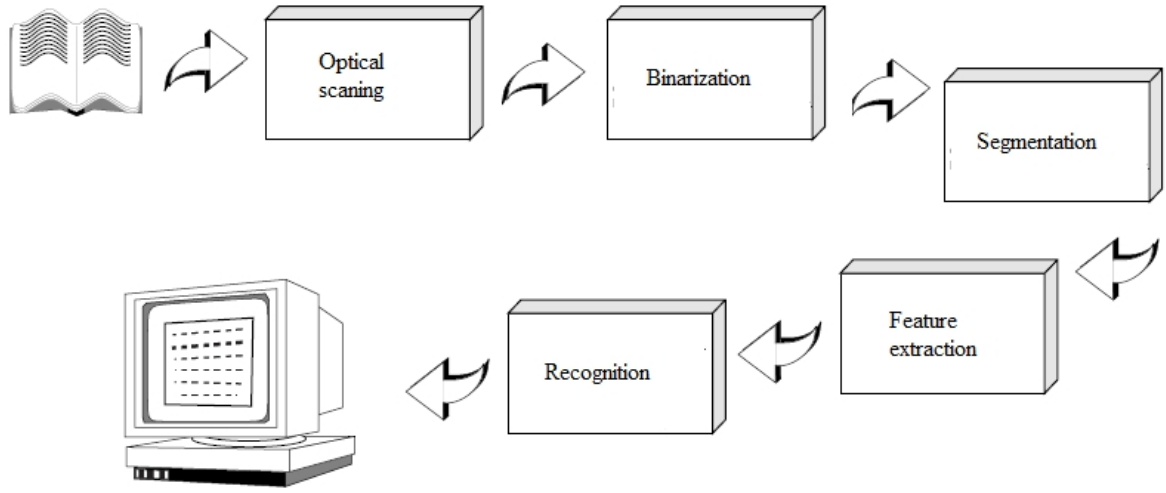


Figure 2.3 Components of an OCR-system.

The identity of each symbol is found by comparing the extracted features with descriptions of the symbol classes obtained through a previous learning phase. Finally contextual information is used to reconstruct the words and numbers of the original text. In the next sections these steps and some of the methods involved are described in more detail.

2.3.1 Image Scanning

In computing, a scanner is a device that optically scans images, printed text, handwriting, or an object, and converts it to a digital image. Common examples found in offices are variations of the desktop (or flatbed) scanner where the document is placed on a glass window for scanning. Hand-held scanners, where the device is moved by hand, have evolved from text scanning "wands" to 3D scanners used for industrial design, reverse engineering, test and measurement, orthotics, gaming and other applications. Mechanically driven scanners that move the document are typically used for large-format documents, where a flatbed design would be impractical.

Modern scanners typically use a charge-coupled device (CCD) or a Contact Image Sensor (CIS) as the image sensor, whereas older drum scanners use a photomultiplier tube as the image sensor. A rotary scanner, used for high-speed document scanning, is another type of drum scanner, using a CCD array instead of a photomultiplier. Other types of scanners are planetary scanners, which take photographs of books and documents, and 3D scanners, for producing three-dimensional models of objects.

Another category of scanner is digital camera scanners, which are based on the concept of reprographic cameras. Due to increasing resolution and new features such as anti-shake, digital cameras have become an attractive alternative to regular scanners. While they still having disadvantages compared to traditional scanners (such as distortion, reflections, shadows, low contrast), digital cameras offer advantages such as speed, portability, gentle digitizing of thick documents without damaging the book spine. New scanning technologies are combining 3D scanners with digital cameras to create full-color, photo-realistic 3D models of objects.

2.3.2 Binarization

With the advancement of technology and widespread use of colour and grayscale scanners, most images scanned now are grayscale. The reasons for not using colour images are the non-colour nature of some texts such as books, the long time needed for scanning, the large volume needed for storing color images and lack of appropriate methods for segmentation of colour images [23].

On the contrary, because of the complexity of the OCR operation, the input of the character recognition phase in most methods is binary images. Therefore, in the pre-processing phase, grayscale images are to be converted to binary images. The most common method is using a threshold. In this method, the pixels lighter than the threshold are turned to white and the remainder to black pixels. An important point to notice in here is to determine the threshold. In some methods in which the used pictures are very similar to each other, a fixed threshold is used [24].

So binarization is the process of converting a grayscale image (0 to 255 pixel values) into binary image (0 to1 pixel values) by thresholding. The binary document image allows the use of fast binary arithmetic during processing, and also requires less space to store [4].

2.3.3 Segmentation Process

Segmentation of text is a process by which the text is partitioned into its coherent parts. The text image contains a number of text lines. Each line again contains a number of words. Each word may contain a number of characters.

The following segmentation scheme is proposed where lines are segmented then words and finally characters. These are then put together to the effect of recognition of individual characters. The individual characters in a word are isolated. Spacing between the characters can be used for segmentation [2].

2.3.3.1 Line Segmentation

Line segmentation is the process of identifying lines in a given image. Steps for the line Segmentation is as follows

1. Scan the BMP image horizontally to find first ON pixel and remember that y coordinate as y1.
2. Continue scanning the BMP image then we would find lots of ON pixel since the characters would have started.
3. Finally we get the first OFF pixel and remember that y coordinate as y2.
4. y1 to y2 is the line.
5. Repeat the above steps till the end of the image.

2.3.3.2 Word Segmentation

As it's known that there is a distance between one word to another word. This concept will be use here for word segmentation. After the line segmentation scan the image vertically for word segmentation [21]. Steps for the word Segmentation is as follows

1. Scan the BMP image vertically for the recognized line segment, to find first ON pixel and remember that x coordinate as x1. Treat this as starting coordinate for the word.
2. Continue scanning the BMP image then we would find lots of ON pixel since the word would have started.
3. Finally we get the successive five (this is assumed word distance) OFF pixel column and remember that x coordinate as x2.
4. x1 to x2 is the word.
5. Repeat the above steps till the end of the line segment.
6. Repeat the above steps for all the recognized line segments.

2.3.3.3 Character Segmentation

Character segmentation is the process of separation of characters word. Steps for the line Segmentation is as follows

1. Scan the BMP image vertically for the recognized word segment, to find first ON pixel and remember that x coordinate as x1. Treat this as starting coordinate for the character.
2. Continue scanning the BMP image then we would find lots of ON pixel since the characters would have started.
3. Finally we get the OFF pixel column and remember that x coordinate as x2.

4. x1 to x2 is the character.
5. Repeat the above steps till the end of the word segment, line segment.
6. Repeat the above steps for all the recognized line segments [2, 25].

2.3.4 Feature Extraction

The objective of feature extraction is to capture the essential characteristics of the symbols, and it is generally accepted that this is one of the most difficult problems of pattern recognition. The most straight forward way of describing a character is by the actual raster image. Another approach is to extract certain features that still characterize the symbols, but leaves out the unimportant attributes. The techniques for extraction of such features are often divided into three main groups, where the features are found from [26]:

- The distribution of points.
- Transformations and series expansions.
- Structural analysis.

The different groups of features may be evaluated according to their sensitivity to noise and deformation and the ease of implementation and use. The results of such a comparison are shown in table 2.1. The criteria used in this evaluation are the following:

i. Robustness.

1. Noise.

Sensitivity to disconnected line segments, bumps, gaps, filled loops etc.

2. Distortions.

Sensitivity to local variations like rounded corners, improper protrusions dilations and shrinkage.

3. Style variation.

Sensitivity to variation in style like the use of different shapes to represent the same character or the use of serifs slants etc.

4. Translation.

Sensitivity to movement of the whole character or its components.

5. Rotation.

Sensitivity to change in orientation of the characters.

ii. Practical use.

1. *Speed of recognition.*
2. *Complexity of implementation.*
3. *Independence.*

Each of the techniques evaluated in table 2.1 are described in the next sections.

Feature extraction technique	Robustness 1 2 3 4 5	Practical use 1 2 3
Template matching	● ● ○ ○ ○	○ ● ○
Transformations	○ ● ● ● ●	○ ○ ●
Distribution of points: Zoning	○ ● ○ ○ ●	● ● ○
Moments	● ● ○ ● ●	○ ● ○
n-tuple	● ○ ● ○ ●	● ● ●
Characteristic loci	○ ● ● ● ●	● ● ○
Crossings	○ ● ● ● ●	● ● ○
Structural features	○ ● ● ● ●	● ○ ●

● High or easy ● Medium ○ Low or difficult

Table 2.1 Evaluation of feature extraction techniques.

2.3.4.1 Template-Matching and Correlation Techniques.

These techniques are different from the others in that no features are actually extracted. Instead the matrix containing the image of the input character is directly matched with a set of prototype characters representing each possible class. The distance between the pattern and each prototype is computed, and the class of the prototype giving the best match is assigned to the pattern [27].

The technique is simple and easy to implement in hardware and has been used in many commercial OCR machines. However, this technique is sensitive to noise and style variations and has no way of handling rotated characters [28].

2.3.4.2 Feature Based Techniques

In these methods, significant measurements are calculated and extracted from a character and compared to descriptions of the character classes obtained during a training phase. The

description that matches most closely provides recognition. The features are given as numbers in a feature vector, and this feature vector is used to represent the symbol [29].

2.3.4.3 Distribution of Points.

This category covers techniques that extract features based on the statistical distribution of points. These features are usually tolerant to distortions and style variations. Some of the typical techniques within this area are listed below [26].

2.3.4.3.1 Zoning.

The rectangle circumscribing the character is divided into several overlapping, or no overlapping, regions and the densities of black points within these regions are computed and used as features [30].

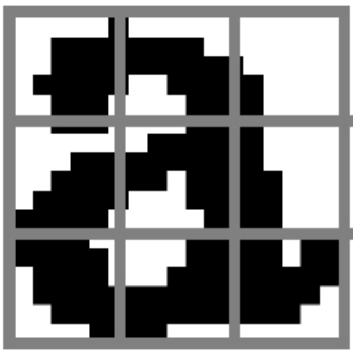


Figure 2.4. Zoning.

2.3.4.3.2 Moments

The moments of black points about a chosen centre, for example the centre of gravity, or a chosen coordinate system, are used as features [31].

2.3.4.3.3 Crossings and Distances

In the crossing technique features are found from the number of times the character shape is crossed by vectors along certain directions. This technique is often used by commercial systems because it can be performed at high speed and requires low complexity.

When using the distance technique certain lengths along the vectors crossing the character shapes are measured, for instance the length of the vectors within the boundary of the character [32].

2.3.4.3.4 N-tuples

The relative joint occurrence of black and white points (foreground and background) in certain specified orderings, are used as features [33].

2.3.4.3.5 Characteristic Loci

For each point in the background of the character, vertical and horizontal vectors are generated.

The numbers of times the line segments describing the character are intersected by these vectors are used as features [34].

2.3.4.4 Transformations and Series Expansions

These techniques help to reduce the dimensionality of the feature vector and the extracted features can be made invariant to global deformations like translation and rotation. The transformations used may be Fourier [35], Walsh, Haar, Hadamard, Karhunen-Loeve, Hough, principal axis transform etc. Many of these transformations are based on the curve describing the contour of the characters [26].

This means that these features are very sensitive to noise affecting the contour of the character like unintended gaps in the contour. In table: 2.1 these features are therefore characterized as having a low tolerance to noise. However, they are tolerant to noise affecting the inside of the character and to distortions.

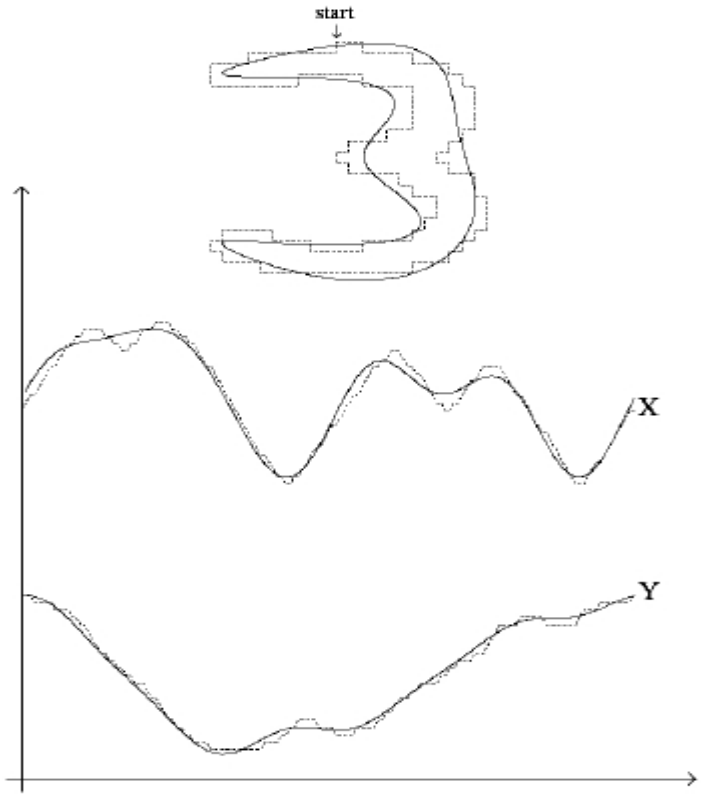


Figure 2.5 Elliptical Fourier descriptors.

2.3.4.5 Structural Analysis

During structural analysis, features that describe the geometric and topological structures of a symbol are extracted. By these features one attempt to describe the physical makeup of the character, and some of the commonly used features are strokes, bays, end-points, intersections between lines and loops. Compared to other techniques the structural analysis gives features with high tolerance to noise and style variations. However, the features are only moderately tolerant to rotation and translation. Unfortunately, the extractions of these features are not trivial, and to some extent still an area of research.

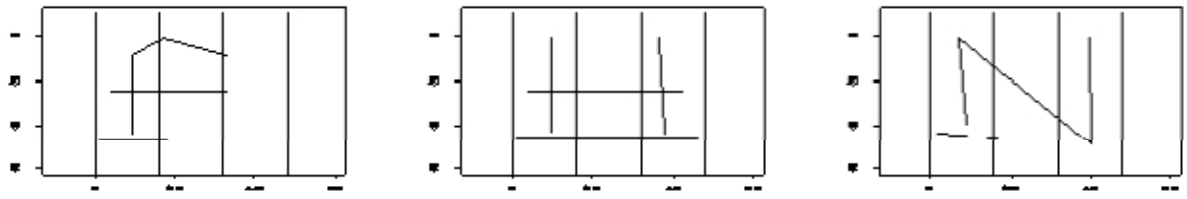


Figure 2.6 Strokes extracted from the capital letters F, H and N.

2.3.5 Recognition

After we got the character by character segmentation we store the character image in a structure. This character as to be identified for the pre defined character set.

There will be preliminary data will be stored for all characters for a identified font and size. This data contains the following information

1. Character ASCII value
2. Character name
3. Character BMP image
4. Character width and length
5. Total number of ON pixel in the image.

For every recognized Character above mentioned information will be captured. The recognized character information will be compared with the pre defined data which we have stored in the system.

As we are using the same font and size for the recognition there will be exact one unique match for the character. This will identify us the name of the character.

If the size of the character varies it will be scaled to the known standard and then recognizing process will be done.

2.4 Text to Speech Conversion System

Traditionally, Text-to-Speech (TTS) systems convert input text into voice by using a set of manually derived rules for prosody generation and/or voice synthesis .While these systems can achieve a high level of intelligibility, they typically sound unnatural. The process of deriving these rules is not only labour intensive but also difficult to8 generalize to a new language, a new voice, or a new speech style.

TTS can "read" text from a document, Web page or e-Book, generating synthesized speech .TTS programs can be useful for a variety of applications. For example, proofreading with TTS allows the author to catch awkward phrases, missing words or pacing problems.

TTS can also convert text files into audio MP3 files that can then be transferred to a portable MP3 player or CD-ROM. This can save time by allowing the user to listen to reports or background materials in bed, *en route* to a meeting, or while performing other tasks.

Even top screenwriting software includes TTS functionality so that a writer can assign different voices to characters in his or her script. The writer can then *listen* to the dialog to weed out stilted sentences. In the area of education, TTS programs provide a valuable edge, particularly for learning new languages. Speech engines are available in a variety of languages, including English, Spanish, German, French, and dozens more.

Figure 2.7 is a simple functional diagram of a general TTS synthesizer. A TTS system is composed of two main parts, the Natural Language Processing (NLP) module and the Digital Signal Processing (DSP) module.

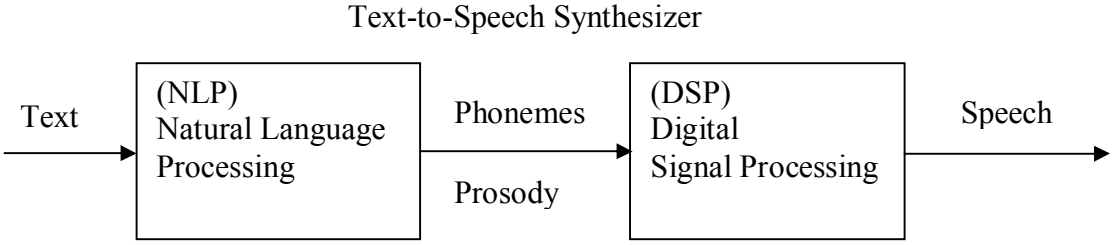


Figure 2.7 General TTS Synthesizer.

The NLP module takes a series of text input and produces a phonetic transcription together with the desired intonation and prosody (rhythm) that is ready to pass on the DSP module. There are three major components within the NLP module, the letter-to-sound component, the prosody generation component, and the morpho-syntactic analyser component [36].

The DSP module takes the phonemes and prosody that were generated by the NLP module and transforms them into speech. There are two main approaches used by DSP module: rule-based-synthesis approach and concatenative-synthesis approach [36].

2.4.1 Natural Language Processing Module

Figure 2.8 introduces the functional view of a NLP module of a general Text-to-Speech conversion system. The NLP module is composed of three major components: text-analyser, letter-to-sound (LTS), and prosody generator.

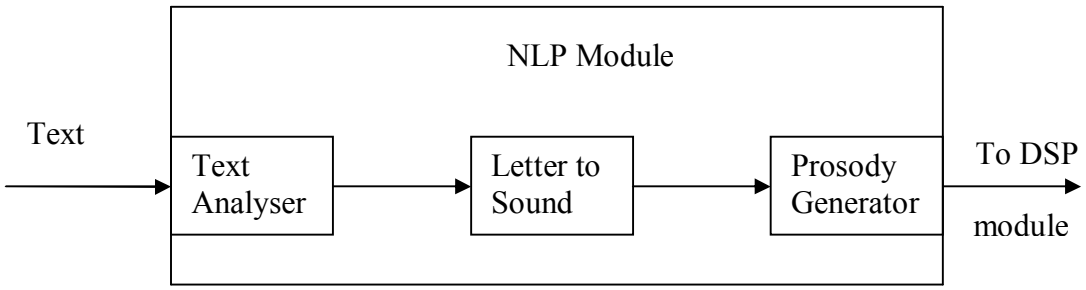


Figure 2.8 A Simple NLP Module.

Besides the expected letter-to-sound and prosody generation blocks, the NLP module comprises a morpho-syntactic analyser, underlying the need for some syntactic processing in a high quality TTS system.

2.4.1.1 Text/Linguistic Analysis

Text analysis is a language-dependent component in TTS system. It is invoked to analyse the input text [37]. This process can be divided into three major steps:

1. Pre-processing- At this stage, the main components of the input text are identified. Pre-processing also determines exact boundaries of the input, which is usually done by delimiting characters like white space, tab or carriage return. In addition, this task identifies abbreviations, numbers, and acronyms within the body of text and transfers them into a predefined format. Usually, pre-processing also segments the whole body of text into paragraphs and organizes these paragraphs into sentences. Finally, pre-processing divides the sentences into words.[36, 38]

2. Morphological analysis - The morphological analysis serves the purpose of generating pronunciations and syntactic information for every word (or lexical phrases) in the text. Most languages have a very large and ever-increasing number of words. Thus, it is impossible to produce an absolutely complete dictionary. Morphological analysis determines the 'root' form of every word, (i.e. love is a root form of loves), and allows the dictionary to store just headword entries, rather than all derived forms of a word [36, 38].

3. Contextual analysis - This task considers words in their context and determines the part-of-speech (POS) for each word in the sentence. To aid this process it have to know the corresponding possible parts of speech of neighbouring words. Context analysis is essential to solve problems like homographs (words that are spelled the same way but have different pronunciations) [36, 38]

2.4.1.2 Letter-to-Sound (LTS)

Dutoit [36] indicates the Letter-To-Sound (LTS) module is responsible for automatically determining the incoming text's phonetic transcription. There are two different types of popular modules used within this task, a dictionary-based module or a rule-based module. According to Dutoit, dictionary-based solutions are based on a large database of phonological knowledge. In order to keep the dictionary size reasonably small, engenerally restricted to morphemes. Examples of morphemes are man, walk, words ending with 'ed', etc. The pronunciation of surface forms is accounted for by inflectional, derivational, and compounding morphophonemic rules [37].

On the other hand, Dutoit goes on to indicate that a different strategy is adopted in rule- based transcription systems, which transfer most of the phonological competence of dictionaries into a set of rules. This time, only those words that are pronounced in such a particular way that they constitute a rule on their own are stored in an exceptions dictionary. Since many exceptions are found in the most frequent words, a reasonably small exceptions dictionary can account for a large fraction of the words in a running text. For instance, in English, 2000 words typically suffice to cover 70% of the words in text.

In the early days powerful dictionary-based methods, which were inherently capable of achieving higher accuracy than letter-to-sound rules, were common given the

availability of very large phonetic dictionaries on computers. Dutoit believes that recently, considerable efforts have been made towards designing sets of rules with a very coverage-start from computerized dictionaries and add rules and exceptions until all words are covered.

2.4.1.3 Prosody Generation

Finding correct intonation, stress, and duration from written text is probably the most challenging problem for years to come. These features together are called prosodic or suprasegmental features and may be considered as the melody, rhythm, and emphasis of the speech at the perceptual level. The intonation means how the pitch pattern or fundamental frequency changes during speech. The prosody of continuous speech depends on many separate aspects, such as the meaning of the sentence and the speaker characteristics and emotions. The prosodic dependencies are shown in Figure 2.9. Unfortunately, written text usually contains very little information of these features and some of them change dynamically during speech. However, with some specific control characters this information may be given to a speech synthesizer.

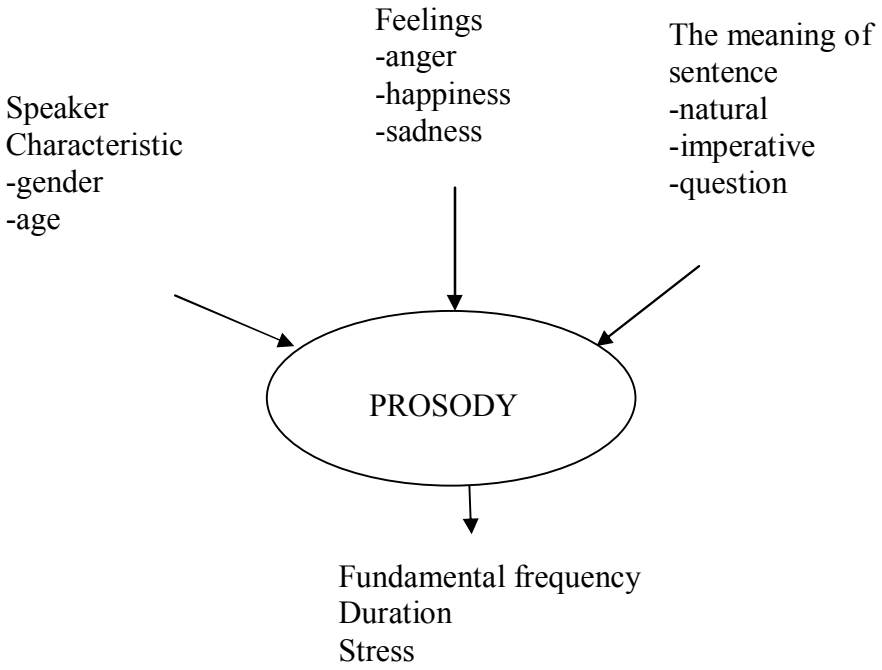


Figure 2.9 Prosodic dependencies.

Prosodic or suprasegmental features consist of pitch, duration, and stress over the time. With good controlling of these genders, age, emotions, and other features in speech can be well modeled. However, almost everything seems to have effect on prosodic features of natural speech which makes accurate modeling very difficult. Prosodic features can be divided into several levels such as syllable, word, or phrase level. For example, at word level vowels are more intense than consonants. At phrase level correct prosody is more difficult to produce than at the word level.

The pitch pattern or fundamental frequency over a sentence (intonation) in natural speech is a combination of many factors. The pitch contour depends on the meaning of the sentence. For example, in normal speech the pitch slightly decreases toward the end of the sentence and when the sentence is in a question form, the pitch pattern will raise to the end of sentence. In the end of sentence there may also be a continuation rise which indicates that there is more speech to come. A raise or fall in fundamental frequency can also indicate a stressed syllable [15, 39].Finally, the pitch contour is also affected by gender, physical and emotional state, and attitude of the speaker.

The duration or time characteristics can also be investigated at several levels from phoneme (segmental) durations to sentence level timing, speaking rate, and rhythm. The segmental duration is determined by a set of rules to determine correct timing. Usually some inherent duration for phoneme is modified by rules between maximum and minimum durations. For example, consonants in non-word-initial position are shortened, emphasized words are significantly lengthened, or a stressed vowel or sonorant preceded by a voiceless plosive is lengthened [39, 40].In general, the phoneme duration differs due to neighboring phonemes. At sentence level, the speech rate, rhythm, and correct placing of pauses for correct phrase boundaries are important. For example, a missing phrase boundary just makes speech sound rushed which is not as bad as an extra boundary which can be confusing.

The intensity pattern is perceived as a loudness of speech over the time. At syllable level vowels are usually more intense than consonants and at a phrase level syllables at the end of an utterance can become weaker in intensity. The intensity pattern in speech is highly related with fundamental frequency. The intensity of a voiced sound goes up in proportion to fundamental frequency [40].

The speaker's feelings and emotional state affect speech in many ways and the proper implementation of these features in synthesized speech may increase the quality considerably. With text-to-speech systems this is rather difficult because written text usually contains no information of these features. However, this kind of information may be provided to a synthesizer with some specific control characters or character strings. The users of speech synthesizers may also need to express their feelings in "real-time". For example, deafened people cannot express their feelings when communicating with speech synthesizer through a telephone line.

This section shortly introduces how some basic emotional states affect voice characteristics. The voice parameters affected by emotions are usually categorized in three main types [41, 42].

- Voice quality which contains largely constant voice characteristics over the spoken utterance, such as loudness and breathiness. For example, angry voice is breathy, loud, and has a tense articulation with abrupt changes while sad voice is very quiet with a decreased articulation precision.
- Pitch contour and its dynamic changes carry important emotional information, both in the general form for the whole sentence and in small fluctuations at word and phonemic levels. The most important pitch features are the general level, the dynamic range, changes in overall shape, content words, stressed phonemes, emphatic stress, and clause boundaries.
- Time characteristics contain the general rhythm, speech rate, the lengthening and shortening of the stressed syllables, the length of content words, and the duration and placing of pauses.

The number of possible emotions is very large, but there are five discrete emotional states which are commonly referred as the primary or basic emotions and the others are altered or mixed forms of these [41, 42]. These are anger, happiness, sadness, fear, and disgust. The secondary emotional states are for example whispering, shouting, grief, and tiredness.

Anger in speech causes increased intensity with dynamic changes. The voice is very breathy and has tense articulation with abrupt changes. The average pitch pattern is higher and there is a strong downward inflection at the end of the sentence. The pitch range

and its variations are also wider than in normal speech and the average speech rate is also a little bit faster.

Happiness or joy causes slightly increased intensity and articulation for content words. The voice is breathy and light without tension. Happiness also leads to increase in pitch and pitch range. The peak values of pitch and the speech rate are the highest of basic emotions.

Fear or anxiety makes the intensity of speech lower with no dynamic changes. Articulation is precise and the voice is irregular and energy at lower frequencies is reduced. The average pitch and pitch range are slightly higher than in neutral speech. The speech rate is slightly faster than in normal speech and contains pauses between words forming almost one third of the total speaking time [41, 42].

Sadness or sorrowness in speech decreases the speech intensity and its dynamic changes. The average pitch is at the same level as in neutral speech, but there are almost no dynamic changes. The articulation precision and the speech rate are also decreased. High ratio of pauses to phonation time also occurs. Grief is an extreme form of sadness where the average pitch is lowered and the pitch range is very narrow. Speech rate is very slow and pauses for almost a half of the total speaking time [42].

2.4.2 Digital Signal Processing (DSP) Module

Generally, the DSP module takes the phonemes and prosodic information that were generated by the NLP module and turns them into speech signals. Sometimes, it might not use the phonemes and prosodic information that were generated by NLP module.

There are two main techniques used in the DSP module, a rule-based synthesizer or a concatenative-based synthesizer. These techniques are described in next section of this chapter.

2.5 Methods, Techniques, and Algorithms

Synthesized speech can be produced by several different methods. The methods are usually classified into three groups:

- Articulatory synthesis, which attempts to model the human speech production system directly.
- Formant synthesis, which models the pole frequencies of speech signal or transfer function of vocal tract based on source-filter-model.
- Concatenative synthesis, which uses different length pre-recorded samples derived from natural speech.

The formant and concatenative methods are the most commonly used in present synthesis systems. The formant synthesis was dominant for long time, but today the concatenative method is becoming more and more popular. The articulatory method is still too complicated for high quality implementations, but may arise as a potential method in the future.

2.5.1 Articulatory Synthesis

Articulatory synthesis tries to model the human vocal organs as perfectly as possible, so it is potentially the most satisfying method to produce high-quality synthetic speech. On the other hand, it is also one of the most difficult methods to implement and the computational load is also considerably higher than with other common methods [43]. Thus, it has received less attention than other synthesis methods and has not yet achieved the same level of success.

Articulatory synthesis typically involves models of the human articulators and vocal cords. The articulators are usually modeled with a set of area functions between glottis and mouth. The first articulatory model was based on a table of vocal tract area functions from larynx to lips for each phonetic segment [39]. For rule-based synthesis the articulatory control parameters may be for example lip aperture, lip protrusion, tongue tip height, tongue tip position, tongue height, tongue position and velic aperture. Phonatory or excitation parameters may be glottal aperture, cord tension, and lung pressure [43].

When speaking, the vocal tract muscles cause articulators to move and change shape of the vocal tract which causes different sounds. The data for articulatory model is usually derived from X-ray analysis of natural speech. However, this data is usually only 2-D when the real vocal tract is naturally 3-D, so the rule-based articulatory synthesis is very difficult to optimize due to the unavailability of sufficient data of the motions of the

articulators during speech. Other deficiency with articulatory synthesis is that X-ray data do not characterize the masses or degrees of freedom of the articulators. Also, the movements of tongue are so complicated that it is almost impossible to model them precisely.

Advantages of articulatory synthesis are that the vocal tract models allow accurate modeling of transients due to abrupt area changes, whereas formant synthesis models only spectral behavior [12]. The articulatory synthesis is quite rarely used in present systems, but since the analysis methods are developing fast and the computational resources are increasing rapidly, it might be a potential synthesis method in the future.

2.5.2 Formant Synthesis

Probably the most widely used synthesis method during last decades has been formant synthesis. There are two basic structures in general, parallel and cascade, but for better performance some kind of combination of these is usually used. Formant synthesis also provides infinite number of sounds which makes it more flexible than for example concatenation methods.

At least three formants are generally required to produce intelligible speech and up to five formants to produce high quality speech. Each formant is usually modelled with a two-pole resonator which enables both the formant frequency (pole-pair frequency) and its bandwidth to be specified [15].

Rule-based formant synthesis is based on a set of rules used to determine the parameters necessary to synthesize a desired utterance using a formant synthesizer. The input parameters may be for example the following, where the open quotient means the ratio of the open-glottis time to the total period duration [44]:

- Voicing fundamental frequency (F0)
- Voiced excitation open quotient (OQ)
- Degree of voicing in excitation (VO)
- Formant frequencies and amplitudes (F1...F3 and A1...A3)
- Frequency of an additional low-frequency resonator (FN)
- Intensity of low- and high-frequency region (ALF, AHF)

A cascade formant synthesizer (Figure 2.10) consists of band-pass resonators connected in series and the output of each formant resonator is applied to the input of the following one. The cascade structure needs only formant frequencies as control information. The main advantage of the cascade structure is that the relative formant amplitudes for vowels do not need individual controls [40].

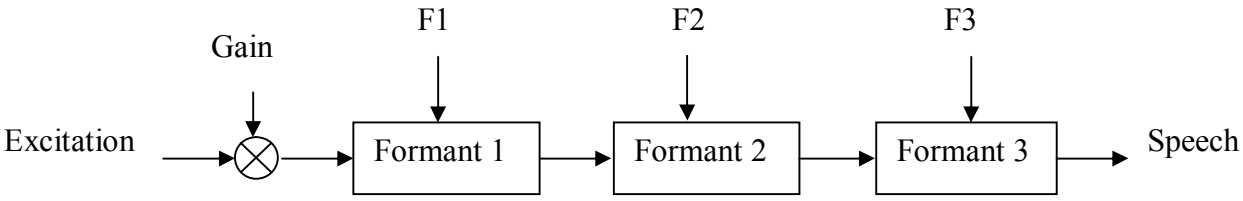


Figure 2.10 Basic structure of cascade formant synthesizer.

The cascade structure has been found better for non-nasal voiced sounds and because it needs less control information than parallel structure, it is then simpler to implement. However, with cascade model the generation of fricatives and plosive bursts is a problem.

A parallel formant synthesizer (Figure 2.11) consists of resonators connected in parallel. Sometimes extra resonators for nasals are used. The excitation signal is applied to all formants simultaneously and their outputs are summed. Adjacent outputs of formant resonators must be summed in opposite phase to avoid unwanted zeros or antiresonances in the frequency response [12]. The parallel structure enables controlling of bandwidth and gains for each formant individually and thus needs also more control information.

The parallel structure has been found to be better for nasals, fricatives, and stop-consonants, but some vowels can not be modelled with parallel formant synthesizer as well as with the cascade one.

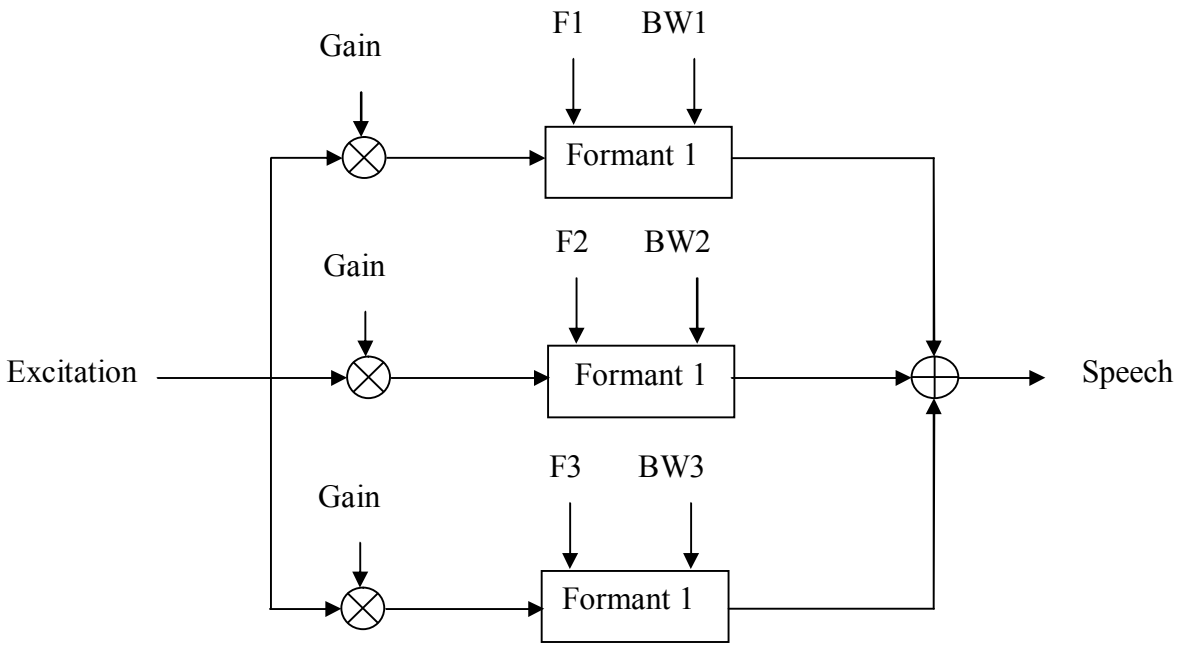


Figure 2.11 Basic structure of a parallel formant synthesizer.

There has been widespread controversy over the quality and suitably characteristics of these two structures. It is easy to see that good results with only one basic method is difficult to achieve so some efforts have been made to improve and combine these basic models.. One solution is to use so called PARCAS (Parallel-Cascade) model introduced and patented by Laine (1982) for SYNTE3 speech synthesizer for Finnish [45]. The model presented in Figure: 2.12, the transfer function of the uniform vocal tract is modelled with two partial transfer functions, each including every second formant of the transfer function. Coefficients k1, k2, and k3 are constant and chosen to balance the formant amplitudes in the neutral vowel to keep the gains of parallel branches constant for all sounds.

The PARCAS model uses a total of 16 control parameters:

- F0 and A0 - fundamental frequency and amplitude of voiced component.
- Fn and Qn - formant frequencies and Q-values (formant frequency/bandwidth).
- VL and VH - voiced component amplitude, low and high.
- FL and FH - unvoiced component amplitude, low and high.
- QN - Q-value of the nasal formant at 250 Hz.

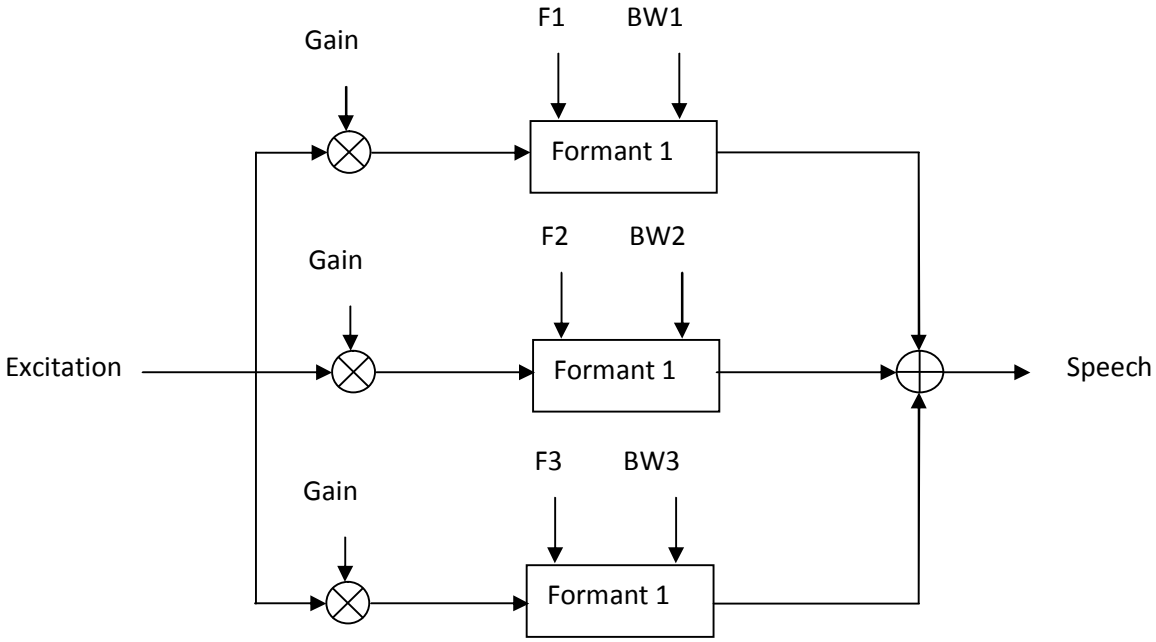


Figure 2.12 PARCAS model (Laine 1989).

The correct and carefully selected excitation is important especially when good controlling of speech characteristics is wanted.

The formant filters represent only the resonances of the vocal tract, so additional provision is needed for the effects of the shape of the glottal waveform and the radiation characteristics of the mouth. Usually the glottal waveform is approximated simply with -12dB/octave filter and radiation characteristics with simple +6dB/octave filter.

2.5.3 Concateative Synthesis

Connecting pre-recorded natural utterances is probably the easiest way to produce intelligible and natural sounding synthetic speech. However, concatenative synthesizers are usually limited to one speaker and one voice and usually require more memory capacity than other methods.

One of the most important aspects in concatenative synthesis is to find correct unit length. The selection is usually a trade-off between longer and shorter units. With longer units, high naturalness, less concatenation points and good control of coarticulation are achieved, but the amount of required units and memory is increased. With shorter units,

less memory is needed, but the sample collecting and labelling procedures become more difficult and complex. In present systems units used are usually words, syllables, demisyllables, phonemes, diphones, and sometimes even triphones.

Word is perhaps the most natural unit for written text and some messaging systems with very limited vocabulary. Concatenation of words is relative easy to perform and coarticulation effects within a word are captured in the stored units. However, there is a great difference with words spoken in isolation and in continuos sentence which makes the continuous speech to sound very unnatural [40]. Because there are hundreds of thousands of different words and proper names in each language, word is not a suitable unit for any kind of unrestricted TTS system.

The number of different syllables in each language is considerably smaller than the number of words, but the size of unit database is usually still too large for TTS systems. For example, there are about 10,000 syllables in English. Unlike with words, the coarticulation effect is not included in stored units, so using syllables as a basic unit is not very reasonable. There is also no way to control prosodic contours over the sentence. At the moment, no word or syllable based full TTS system exists. The current synthesis systems are mostly based on using phonemes, diphones, demisyllables or some kind of combinations of these.

Demisyllables represents the initial and final parts of syllables. One advantage of demisyllables is that only about 1,000 of them is needed to construct the 10,000 syllables of English [15]. Using demisyllables, instead of for example phonemes and diphones, requires considerably less concatenation points. Demisyllables also take account of most transitions and then also a large number of coarticulation effects and also covers a large number of allophonic variations due to separation of initial and final consonant clusters. However, the memory requirements are still quite high, but tolerable. Compared to phonemes and diphones, the exact number of demisyllables in a language can not be defined. With purely demisyllable based system, all possible words can not be synthesized properly. This problem is faced at least with some proper names. However, demisyllables and syllables may be successfully used in a system which uses variable length units and affixes.

Phonemes are probably the most commonly used units in speech synthesis because they are the normal linguistic presentation of speech. The inventory of basic units is usually between 40 and 50, which is clearly the smallest compared to other units [40]. Using phonemes gives maximum flexibility with the rule-based systems. However, some phones that do not have a steady-state target position, such as plosives, are difficult to synthesize. The articulation must also be formulated as rules. Phonemes are sometimes used as an input for speech synthesizer to drive for example diphone-based synthesizer.

Diphones (or dyads) are defined to extend the central point of the steady state part of the phone to the central point of the following one, so they contain the transitions between adjacent phones. That means that the concatenation point will be in the most steady state region of the signal, which reduces the distortion from concatenation points. Another advantage with diphones is that the coarticulation effect needs no more to be formulated as rules. In principle, the number of diphones is the square of the number of phonemes (plus allophones), but not all combinations of phonemes are needed. For example, in Finnish the combinations, such as /hs/, /sj/, /mt/, /nk/, and /ŋ p/ within a word are not possible. The number of units is usually from 1500 to 2000, which increases the memory requirements and makes the data collection more difficult compared to phonemes. However, the number of data is still tolerable and with other advantages, diphone is a very suitable unit for sample-based text-to-speech synthesis. The number of diphones may be reduced by inverting symmetric transitions, like for example /as/ from /sa/.

Longer segmental units, such as triphones or tetraphones, are quite rarely used. Triphones are like diphones, but contains one phoneme between steady-state points (half phoneme - phoneme - half phoneme). In other words, a triphone is a phoneme with a specific left and right context. For English, more than 10,000 units are required.

Building the unit inventory consists of three main phases [46]. First, the natural speech must be recorded so that all used units (phonemes) within all possible contexts (allophones) are included. After this, the units must be labelled or segmented from spoken speech data, and finally, the most appropriate units must be chosen. Gathering the samples from natural speech is usually very time-consuming. However, some is this work may be done automatically by choosing the input text for analysis phase properly. The

implementation of rules to select correct samples for concatenation must also be done very carefully [46].

There are several problems in concatenative synthesis compared to other methods.

- Distortion from discontinuities in concatenation points, which can be reduced using diphones or some special methods for smoothing signal.
- Memory requirements are usually very high, especially when long concatenation units are used, such as syllables or words.
- Data collecting and labelling of speech samples is usually time-consuming. In theory, all possible allophones should be included in the material, but trade-offs between the quality and the number of samples must be made.

Some of the problems may be solved with methods described below and the use of concatenative method is increasing due to better computer capabilities [15].

2.5.3.1 PSOLA Methods

The PSOLA (Pitch Synchronous Overlap Add) method was originally developed at France Telecom (CNET). It is actually not a synthesis method itself but allows prerecorded speech samples smoothly concatenated and provides good controlling for pitch and duration, so it is used in some commercial synthesis systems, such as ProVerbe and HADIFIX [15].

There are several versions of the PSOLA algorithm and all of them work in essence the same way. Time-domain version, TD-PSOLA, is the most commonly used due to its computational efficiency [47]. The basic algorithm consists of three steps[8]. The analysis step where the original speech signal is first divided into separate but often overlapping short-term analysis signals (ST), the modification of each analysis signal to synthesis signal, and the synthesis step where these segments are recombined by means of overlap-adding. Short term signals $x_m(n)$ are obtained from digital speech waveform $x(n)$ by multiplying the signal by a sequence of pitch-synchronous analysis window $h_m(n)$

$$x_m(n) = h_m(t_m - n)x(n) \tag{2.1}$$

Where m is an index for the short-time signal. The windows, which are usually Hanning type, are centered around the successive instants t_m, called pitch-marks. These marks are set at a pitch-synchronous rate on the voiced parts of the signal and at a constant

rate on the unvoiced parts. The used window length is proportional to local pitch period and the window factor is usually from 2 to 4[9, 48]. The pitch markers are determined either by manually inspection of speech signal or automatically by some pitch estimation methods [47]. The segment recombination in synthesis step is performed after defining a new pitch-mark sequence.

Manipulation of fundamental frequency is achieved by changing the time intervals between pitch markers (see Figure: 2.13.). The modification of duration is achieved by either repeating or omitting speech segments. In principle, modification of fundamental frequency also implies a modification of duration [47].

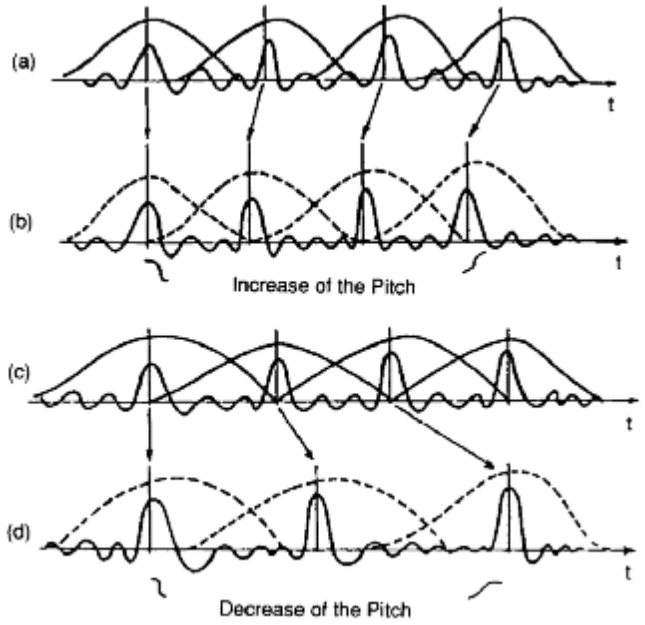


Figure 2.13 Pitch modification of a voiced speech segment.

2.5.3.2 Sinusoidal Models

Sinusoidal models are based on a well known assumption that the speech signal can be represented as a sum of sine waves with time-varying amplitudes and frequencies [9, 49]. In the basic model, the speech signal *s(n)* is modeled as the sum of a small number *L* of sin usoids

$$s(n) = \sum_{l=1}^{L} A_l \cos(w_l n + \phi_l n) \qquad (2.2)$$

Where $A_l(n)$ and $\phi_l(n)$ represent the amplitude and phase of each sinusoidal component associated with the frequency track ω_l. To find these parameters $A_l(n)$ and $\phi_l(n)$, the DFT of windowed signal frames is calculated, and the peaks of the spectral magnitude are selected from each frame (see Figure 2.14). The basic model is also known as the McAulay/Quatieri Model. The basic model has also some modifications such as ABS/OLA (Analysis by Synthesis / Overlap Add) and Hybrid / Sinusoidal Noise models.

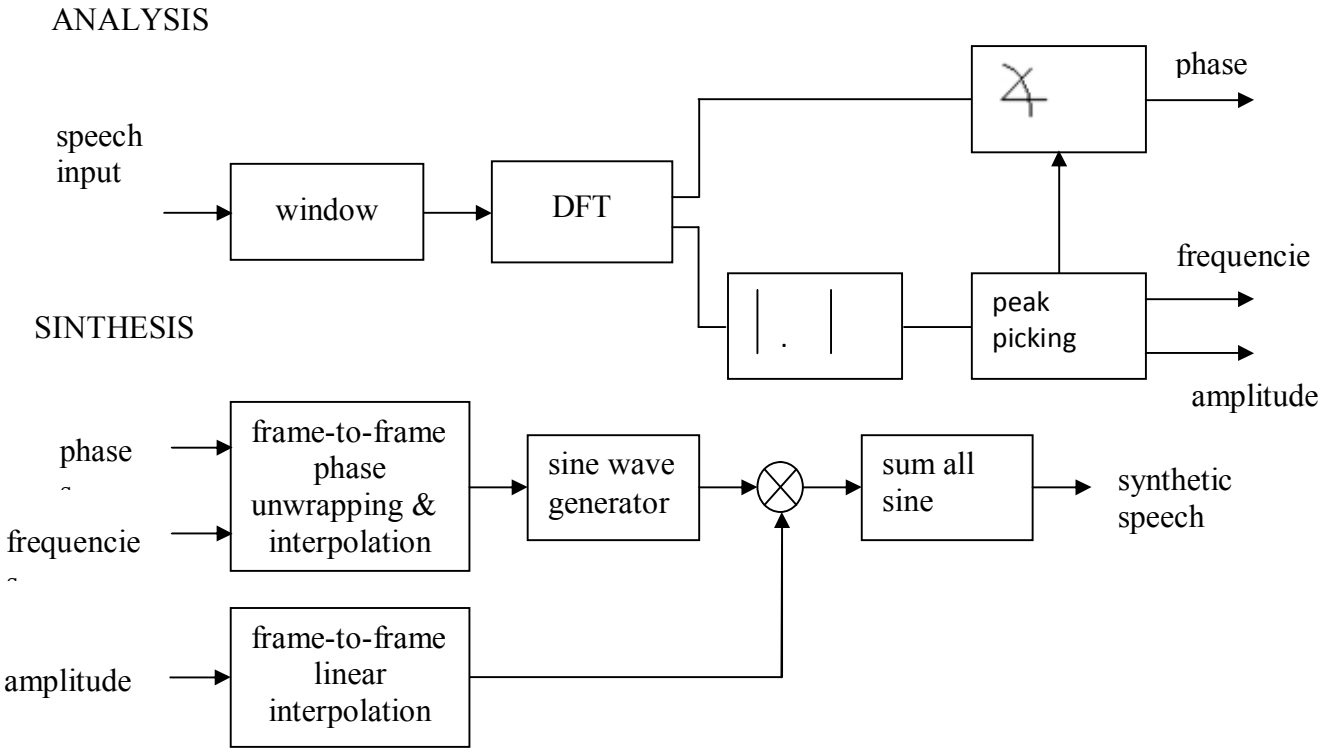


Figure 2.14 Sinusoidal analysis/synthesis system.

While the sinusoidal models are perhaps very suitable for representing periodic signals, such as vowels and voiced consonants, the representation of unvoiced speech becomes problematic [49].

2.6 Applications of OCR Based Synthetic Speech. System

Synthetic speech generated by OCR base speech synthesis system may be used in several applications. Communication aids have developed from low quality talking calculators to modern 3D applications, such as talking heads. The implementation method depends mostly on used application. Some applications are given bellow for OCR based speech synthesis system:

2.6.1 Applications for the Blind

Probably the most important and useful application field in speech synthesis is the reading and communication aids for the blind. Before synthesized speech, specific audio books were used where the content of the book was read into audio tape. It is clear that making such spoken copy of any large book takes several months and is very expensive.

2.6.2 Educational Applications

Synthesized speech can be used also in many educational situations. A computer with speech synthesizer can teach 24 hours a day and 365 days a year. It can be programmed for special tasks like spelling and pronunciation teaching for different languages. It can also be used with interactive educational applications.

Especially with people who are impaired to read (dyslexics), speech synthesis may be very helpful because especially some children may feel themselves very embarrassing when they have to be helped by a teacher [39]. It is also almost impossible to learn write and read without spoken help. With proper computer software, unsupervised training for these problems is easy and inexpensive to arrange.

2.6.3 Applications for the Deafened and Vocally Handicapped

People who are born-deaf cannot learn to speak properly and people with hearing difficulties have usually speaking difficulties. Synthesized speech gives the deafened and vocally handicapped an opportunity to communicate with people who do not understand

the sign language. With a talking head it is possible to improve the quality of the communication situation even more because the visual information is the most important with the deaf and dumb. A speech synthesis system may also be used with communication over the telephone line [39].

Chapter 3

OCR based Speech Synthesis System

3.1 Introduction

A OCR based Speech Synthesis System is a computer-based system that should be able to read any text and give voice output, when the text is scanned and submitted to an Optical Character Recognition (OCR) system.

3.2 Hardware Requirements

The system comprises of mostly software portion but had some hardware involved too. The hardware that we used was:

- Scanner
- P.C
- Speaker

3.3.1 Scanner

a scanner is a device that optically scans images, printed text, handwriting, or an object, and converts it to a digital image Modern scanners typically use a charge-coupled device (CCD) or a Contact Image Sensor (CIS) as the image sensor, whereas older drum scanners use a photomultiplier tube as the image sensor. A rotary scanner, used for high-speed document scanning, is another type of drum scanner, using a CCD array instead of a photomultiplier. Other types of scanners are planetary scanners, which take photographs of books and documents, and 3D scanners, for producing three-dimensional models of objects.

3.3.2 Computers/Processors

OCR based Speech Synthesis System applications require a high processing speed computer system to perform specified task. It's possible to do with 100MHz and 16M RAM, but for fast processing (large dictionaries, complex recognition schemes, or high sample rates), you should shoot for a minimum of a 400MHz and 128M RAM. Because of the processing required, most software packages list their minimum requirements. It requires a operating system and sound must be installed in PC.

3.3.3 Speaker

OCR based Speech Synthesis System applications requires a good quality, low cost speaker to produce a good quality of sound.

3.3 Software Platform

The software platform used here is LabVIEW (Laboratory Virtual Instrument Engineering Workbench).

3.3.1 LabVIEW

LabVIEW is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine the order of program execution, LabVIEW uses dataflow programming, where the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions.VIs, or virtual instruments, are LabVIEW programs that imitate physical instruments.

In LabVIEW, user builds a user interface by using a set of tools and objects. The user interface is known as the front panel. User then adds code using graphical representations of functions to control the front panel objects. This graphical source code is also known as G code or block diagram code. The block diagram contains this code. In some ways, the block diagram resembles a flowchart.

3.3.2 Virtual instruments

LabVIEW works on a data flow model in which information within a LabVIEW program, called a virtual instrument (VI), flows from data sources to data sinks connected by wires. The data can be modified as it is passed from source to sink by other VIs LabVIEW supports two types of VIs--internal VIs and user created VIs Internal VIs are packaged with LabVIEW and perform simple functions like adding numbers or opening files. User created VIs consists of both a graphical user interface called the front panel and a code pipeline called the block diagram. These VIs tend to be much more complex considering that they can contain any number of internal or user created VIs in an endless number of configurations.

Consider a simplistic LabVIEW program which takes a single number from the user and multiplies it by 10. Analyzing such a program reveals the following data flow structure:

1. The user inputs a number (data source).
2. The program executes an addition VI taking the user's number and the number 10 as its inputs (data sink).
3. The addition VI returns the result of the addition operation (data source).
4. The result is displayed on the screen (data sink).

While this example is simplistic, it exemplifies how all LabVIEW VIs work. Data always flows from data sources to data sinks according to the block diagram, much like how water flows through a pipe.

3.3.3 LabVIEW Program Structure

A LabVIEW program is similar to a text-based program with functions and subroutines; however, in appearance it functions like a virtual instrument (VI). A real instrument may accept an input, process on it and then output a result. Similarly, a LabVIEW VI behaves in the same manner.

A LabVIEW VI has 3 main parts:

a) Front Panel window

Every user created VI has a front panel that contains the graphical interface with which a user interacts. The front panel can house various graphical objects ranging from simple buttons to complex graphs. Various options are available for changing the look and feel of the objects on the front panel to match the needs of any application.

b) Block Diagram window

Nearly every VI has a block diagram containing some kind of program logic that serves to modify data as it flows from sources to sinks. The block diagram houses a pipeline structure of sources, sinks, VIs, and structures wired together in order to define this program logic. Most importantly, every data source and sink from the front panel has its analog source and sink on the block diagram. This representation allows the input values from the user to be accessed from the block diagram. Likewise, new output values can be shown on the front panel by code executed in the block diagram.

c) Controls, Functions and Tools Palette

Windows, which contain icons associated with extensive libraries of software functions, subroutines, etc.

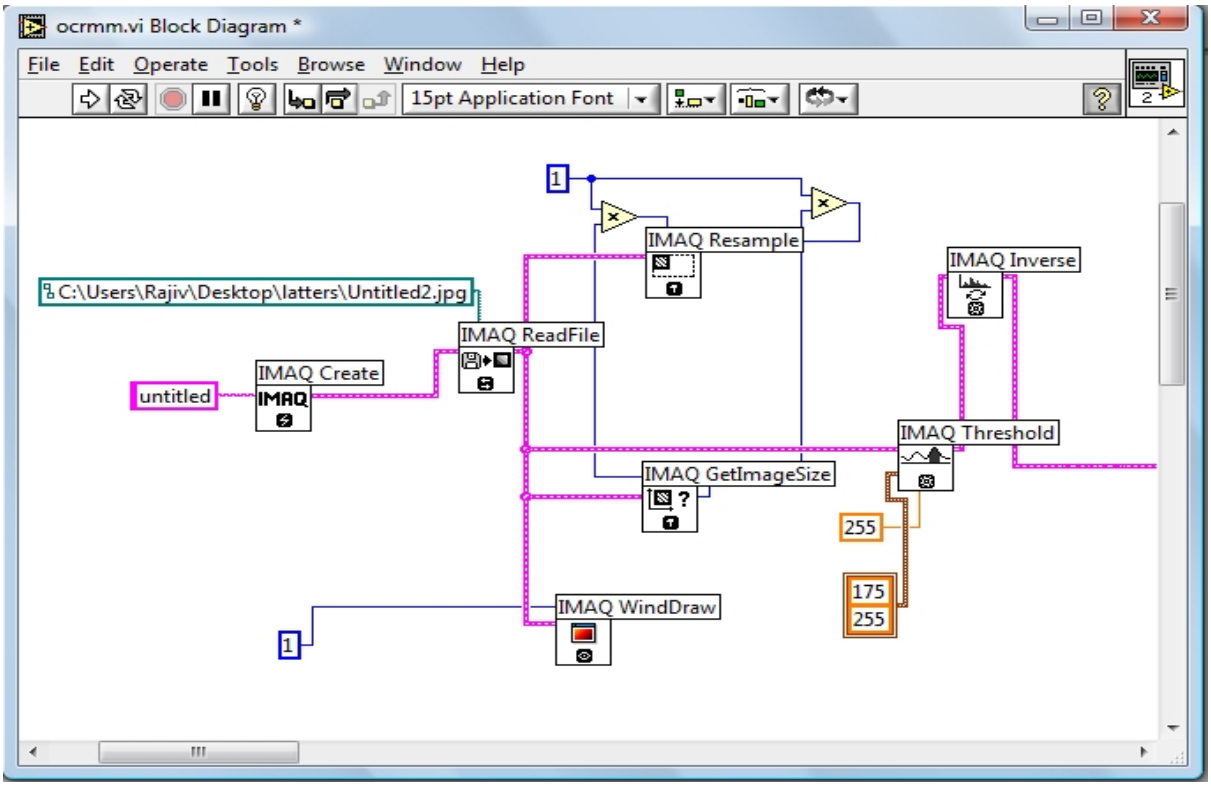


Figure 3.1 Block Diagram of OCR System.

3.4 Software Implementation

LabVIEW software of OCR base speech synthesis system includes two steps:

1 Optical character recognition

2 Text to speech synthesis

3.4.1 Optical Character Recognition

In optical character recognition process image of printed text is used as input for OCR system. Figure: 3.2 show the flow chart of OCR system.

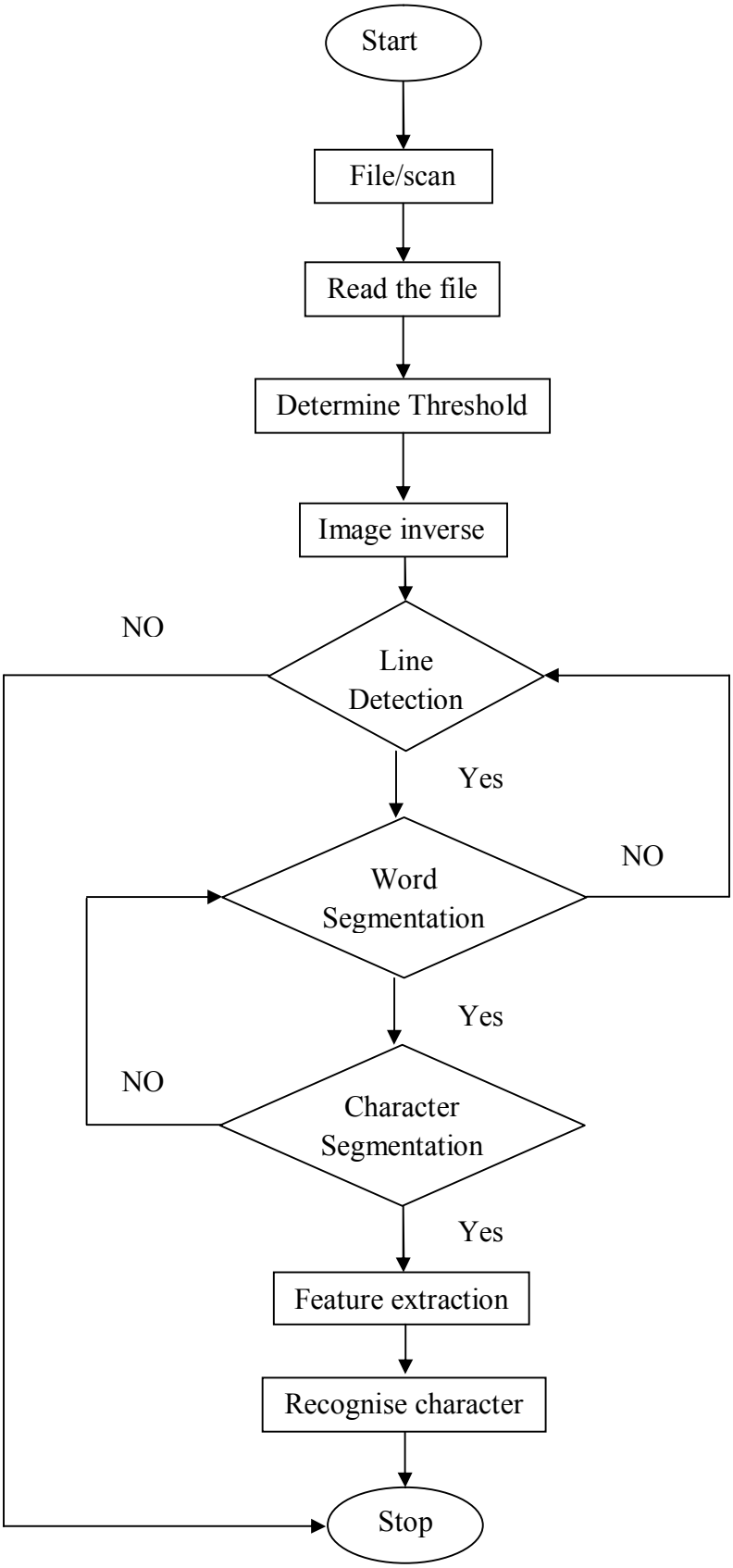


Figure 3.2 Flow chart OCR system.

In optical character recognition process five steps are involve:

1 Image Acquisition

2 Image Pre-processing (Binarization)

3 Image Segmentation

4 Template matching

5 Recognition

3.4.1.1 Image Acquisition

The image has been captured using a digital HP scanner. The flap of the scanner had been kept open during the acquisition process in order to obtain a uniform black background.

Image configuration has been done with the help of Imaq create subvi of LabVIEW. The configuration of the image means selecting the image type and border size (default is 3) of the image as per the requirement. Then Imaq file read subvi is use to read the file as shown in Figure 3.2

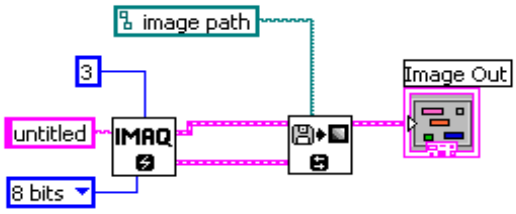


Figure 3.3 Image configuration.

3.4.1.2 Image Pre-processing (Binarization, Thresholding)

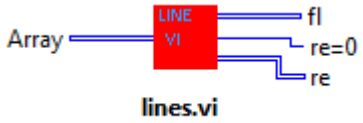
Binarization is the process of converting a grayscale image (0 to 255 pixel values) into binary image (0 to1 pixel values) by a threshold value of 175. the pixels lighter than the threshold are turned to white and the remainder to black pixels.

3.4.1.3 Image Segmentation

The input of this step is obtained thresholded image from above step. Three steps are involved in segmentation process are described below.

3.4.1.3.1 Line Detection and Segmentation

The input of line detection sub-vi is array of image. It scane BMP image horizontally to find first ON pixel and remember that y coordinate as y1. Continue scanning the BMP image then we would find lots of ON pixel since the characters would have started. When finally we get the first OFF pixel and remember that y coordinate as y2 Then this Sub-vi clip the first line (fl) from input image between the coordinate y1 to y2, pixel values are zero(OFF) below that line up to starting point of new line. So that line will be clipped and used further for word detection and segmentation. Here re is rest image value.



Block diagram of the line detection subvi is shown below in Figure 3.4.

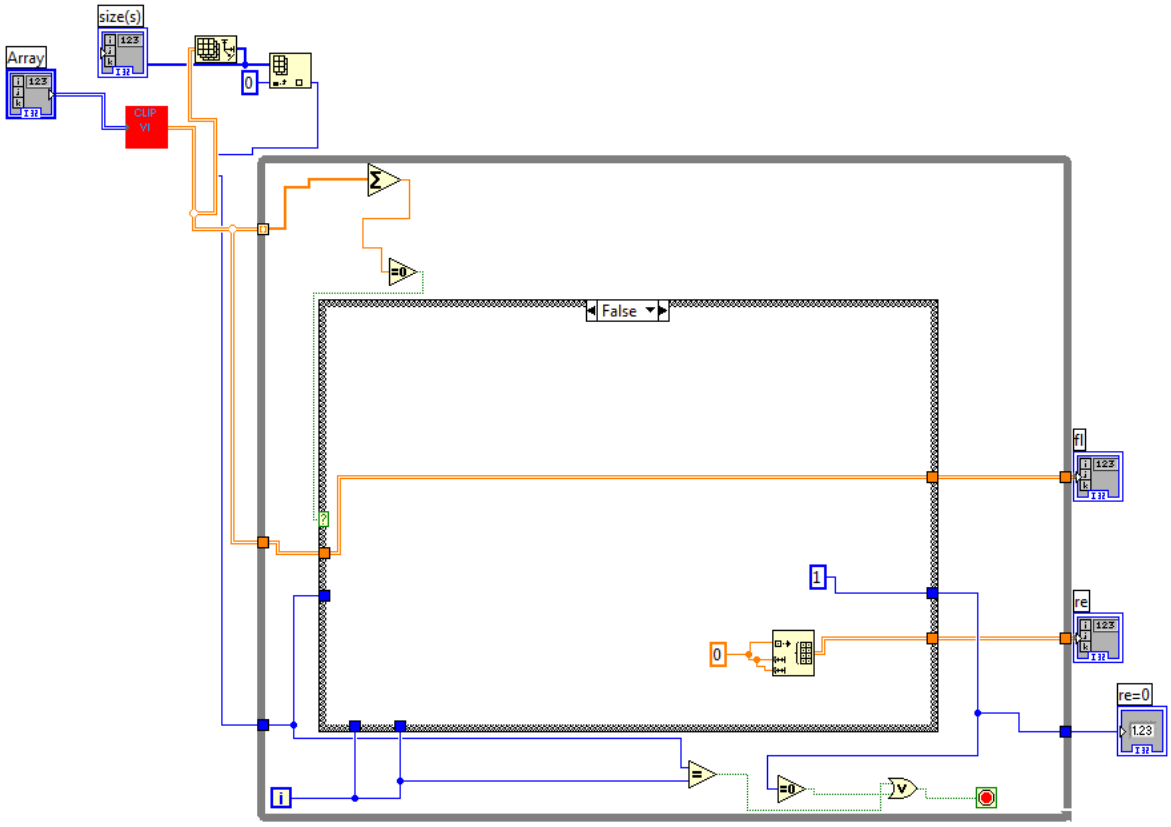


Figure 3.4 Block diagram of the line detection vi.

3.4.1.3.2 Word Segmentation

The space sub-vi is used for word detection and segmentation. It Scan the BMP image vertically for the recognized line segment, to find first ON pixel and remember that x coordinate as x1. Treat this as starting coordinate for the word. Continue scanning the BMP image then we would find lots of ON pixel since the word would have started. Finally we get the successive fifteen (this is assumed word distance) OFF pixel column and remember that x coordinate as x2. x1 to x2 is the word.

Input of this sub-vi is segmented image line of text by previous line detection sub-vi. Here re output is rest words in that line and next is segmented word in that will be used in next step for character recognition. The block diagram of that sub-vi is shown below in Figure 3.5.

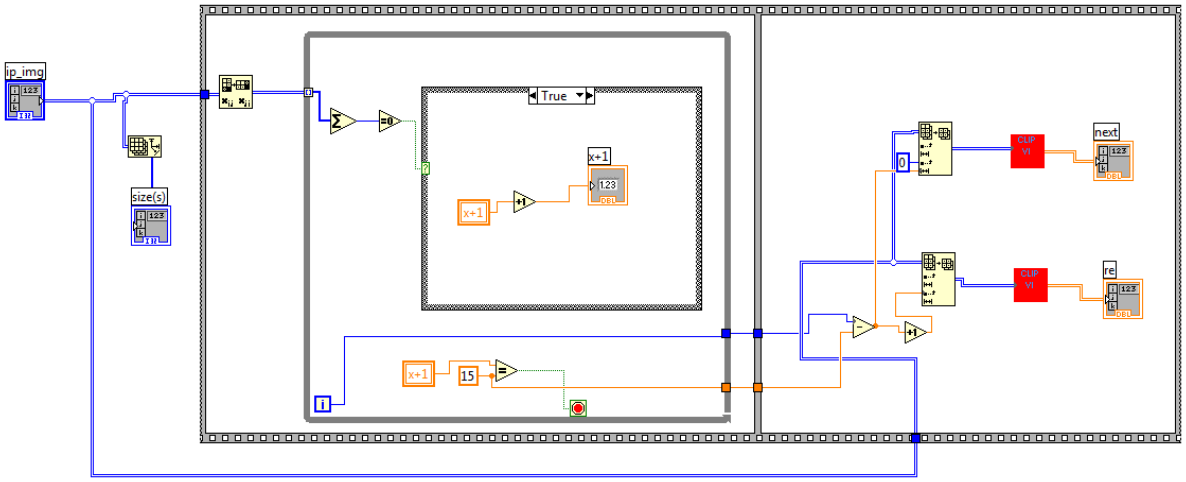
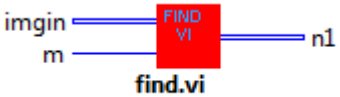


Figure 3.5 Block diagram of word segmentation sub-vi.

3.4.1.3.3 Character Segmentation

Scan the BMP image vertically for the recognized word segment, to find first ON pixel and remember that x coordinate as x1. Treat this as starting coordinate for the character. Continue scanning the BMP image then we would find lots of ON pixel since the characters would have started. Finally we get the OFF pixel column and remember that x coordinate as x2. x1 to x2 is the character. In character segmentation two subvi's are used. The label sub-vi generally assigns a label (numeric value), likewise 1 to first, and 2 to second up to last character of the coming input word. Here two output of the vi first is array of labelled image, and numeric is gives how many particles or character in that particular word.

Character segmentation is done here.by find sub vi This sub vi have two input and one output. first input is imgin it is lablled characters image of that perticular word comming from above lable vi, and second is m to give the which number(label) character will be seperated from word. If value of m(1) first character will be segmented from word. Here output n1 is character seperated. Repeat the above steps till the end of the word segment, line segment.



Block diagram of Find sub vi is shown below in Figure 3.6.

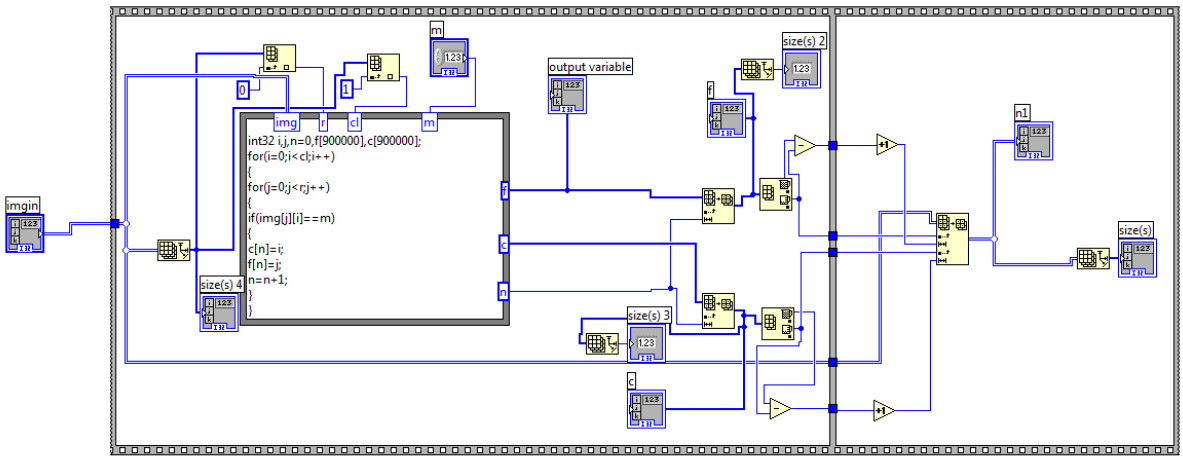


Figure 3.6 Block diagram of Find vi.

3.4.1.4 Template matching

Template matching is process in which correlation between stored templates and segmented character will be finding in Lab View by using correlation vi, which is described below.

3.4.1.4.1 Correlation

The correlation sub vi find best correlation between segmented character and stored templates of each character. Here two inputs first one is segmented character image and second one stored template image. Output of this is correlations between segmented character and every stored template.

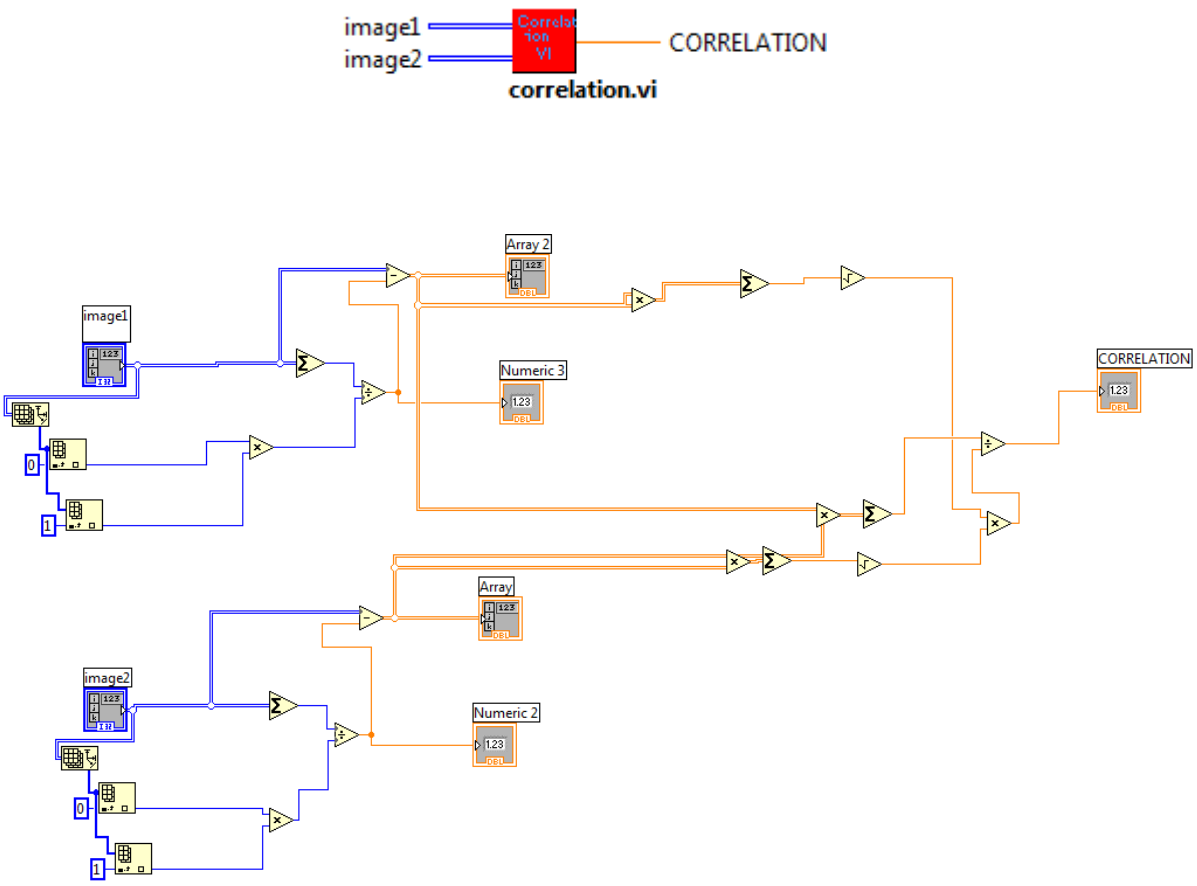


Figure 3.7 Block diagram of correlation vi.

3.4.1.5 Recognition

After we got the character by character segmentation we store the character image in a structure. This character as to be identified for the pre defined character set. There will be preliminary data will be stored for all characters for a identified font and size. This data contains the following information

1. Character ASCII value
2. Character name
3. Character BMP image
4. Character width and length

5. Total number of ON pixel in the image.

For every recognized Character above mentioned information will be captured. The recognized character information will be compared with the pre defined data which we have stored in the system.

As we are using the same font and size for the recognition there will be exact one unique match for the character. This will identify us the name of the character.

If the size of the character varies it will be scaled to the known standard and then recognizing process will be done.

3.4.2 Text to speech synthesis

In text to speech module text recognised by OCR system will be the inputs of speech synthesis system which is to be converted into speech in .wav file format and creates a wave file named output .wav, which can be listen by using wave file player.

Two steps involved in text to speech synthesis

1 Text to speech conversion

2 Play speech in .wave file formate

3.4.2.1 Text to speech conversion

In the text speech conversion input text is converted speech (in LabVIEW) by using automation open, invoke node and property node will be described below in next section of this chapter. Flow chart text speech conversion is shown below in Figure 3.7.

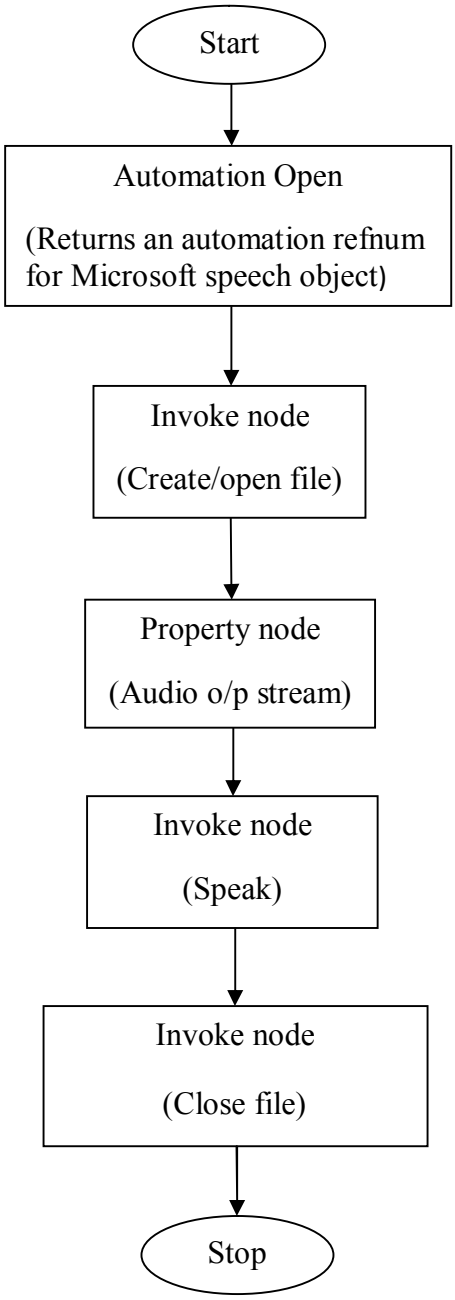


Figure 3.8 Flowchart for the text to speech wave file conversion.

In LabVIEW the ACTIVE X sub pallet in Communication pallet and its functions to exchange data between applications. ActiveX technology provides a standard model for interapplication communication that different programming languages can implement on different platforms

3.4.2.1.1 Overview of ActiveX

ActiveX is the general name for a set of Microsoft Technologies that allows you to reuse code and link individual programs together to suit your computing needs. Based on COM (Component Object Model) technologies, ActiveX is an extension of a previous technology called OLE (Object Linking and Embedding). Each program does not need to regenerate components, but rather, reuse components to give user the power to combine applications together. LabVIEW offers support for ActiveX automation as a server as well as support for ActiveX Containers, and ActiveX Events [50].

3.4.2.1.2 ActiveX Automation

ActiveX/COM refers to the process of controlling one program from another via ActiveX. Like networking, one program acts as the client and the other as the server. LabVIEW supports automation both as the client and the server. Both programs, client and server, exist independent of each other but are able to share information. The client communicates with the ActiveX objects that the server opens to allow the sharing of information. The automation client can access the object's properties and methods. Properties are attributes of an object. Another program can set or retrieve an object's attributes. Similarly, methods are functions that perform an operation on objects. Other applications can invoke methods. An example of an ActiveX property is the program name, height or width. An example of an ActiveX method is the save or print method [50].

3.4.2.1.3 ActiveX Automation with LabVIEW

LabVIEW as an ActiveX server or ActiveX client can interface with other programs from the LabVIEW programming interface. In this case, LabVIEW acts as the automation client and requests information of the automation server, or other program. Likewise, other ActiveX automation clients can interface with the LabVIEW ActiveX automation server.

3.4.2.1.4 LabVIEW as an Automation Client

LabVIEW provides functions in its API that allow LabVIEW to act as an automation client with any automation server. The diagram below shows Lab View’s programming flow, and gives the associated functions with each block [50].

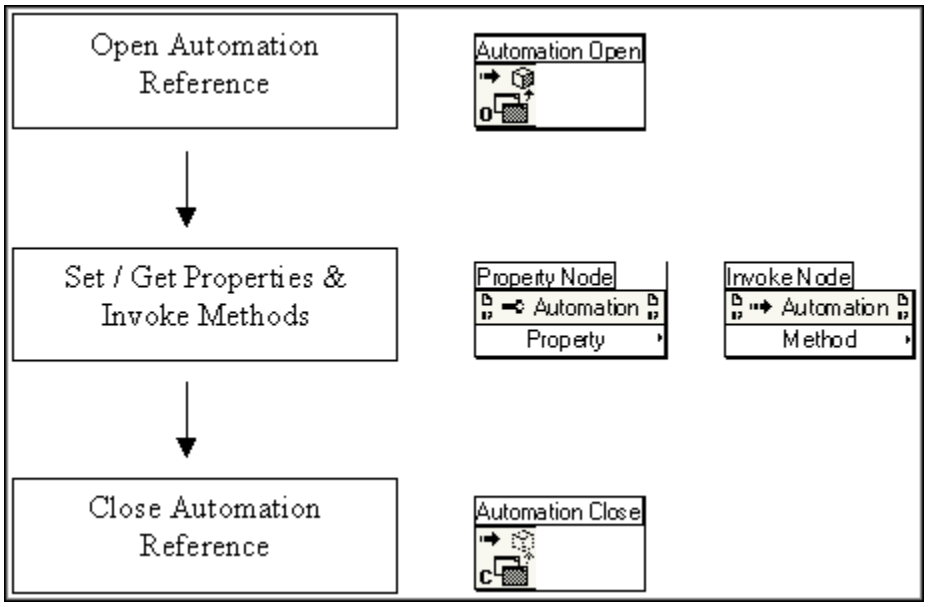
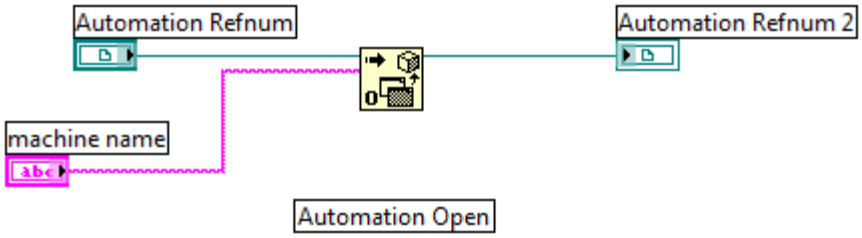


Figure 3.9 Programming flow of ActiveX used in LabVIEW.

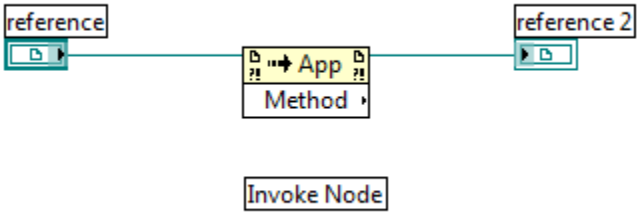
3.4.2.1.5 Automation Open (Windows)

Returns an automation refnum, which points to a specific ActiveX object. In Text to Speech VI, it gives refnum for Microsoft speech object library.



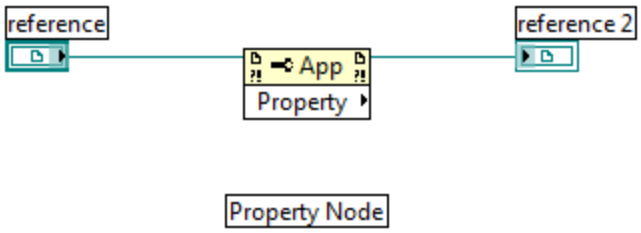
3.4.2.1.6 Invoke Node

Invokes a method or action on a reference. Most methods have associated parameters. If the node is configured for VI Server Application class or Virtual Instrument class and reference is unwired, reference defaults to the current Application or VI.



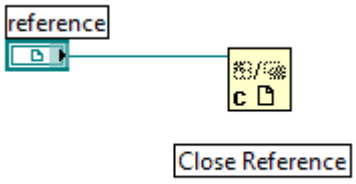
3.4.2.1.7 Property Node

Gets (reads) and/or sets (writes) properties of a reference. The Property Node automatically adapts to the class of the object that you **reference**. LabVIEW includes Property Nodes preconfigured to access VISA properties and ActiveX properties.



3.4.2.1.8 Close Reference

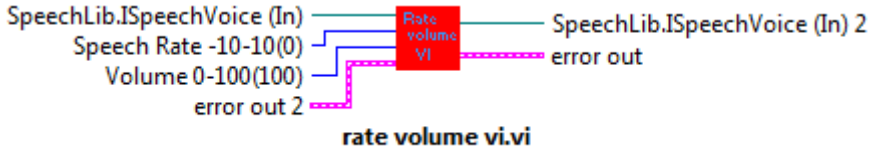
Closes a refnum associated with an open VI, VI object, an open instance of LabVIEW, or an ActiveX or .NET object.



Some sub vis are used in text to speech conversion like wise rate volume vi and status vi are described below

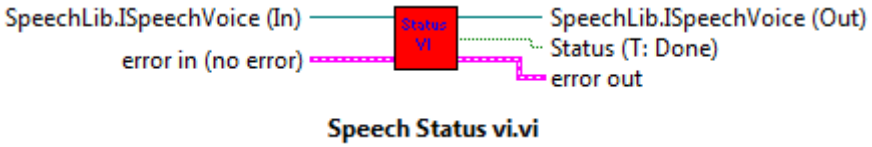
3.4.2.1.9 Rate Volume VI

Rate volume vi is used to control speech rate and volume of genarated output speech. In this rate of speech can vary in the rane from -10 to 10 and vollume is vary in the range between 0 to 100.



3.4.2.1.10 Status VI

Status VI is used to show current status of speech signal which is currently generating or not. If speech is currently generating then Boolean LED will glow and speech is currently not generating then LED will not glow in front panel of VI.



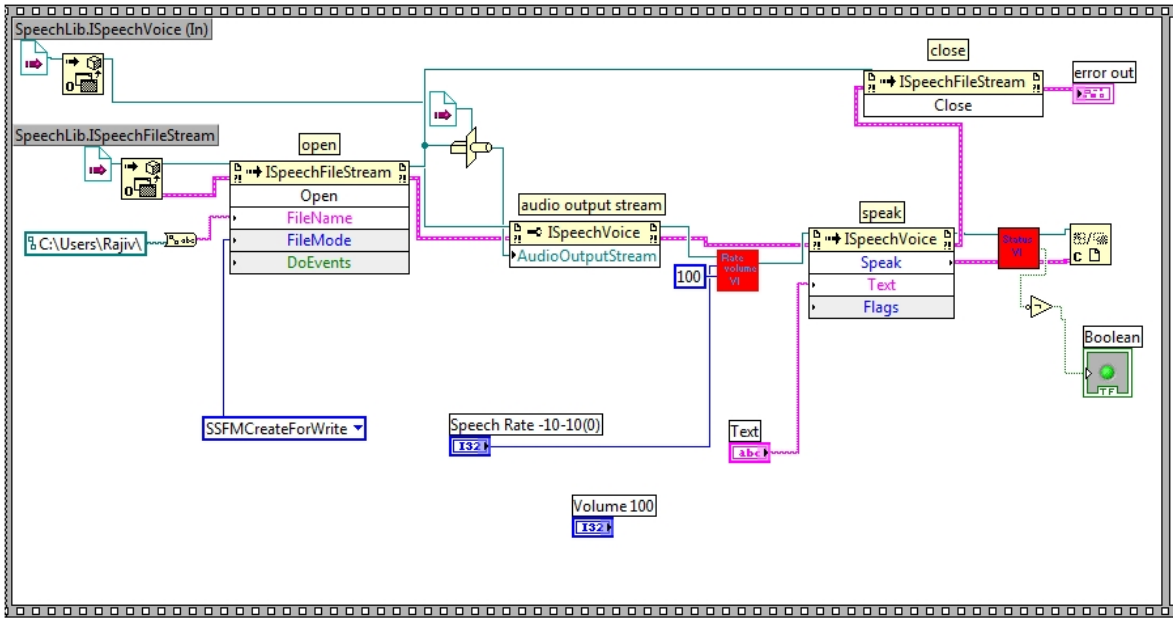


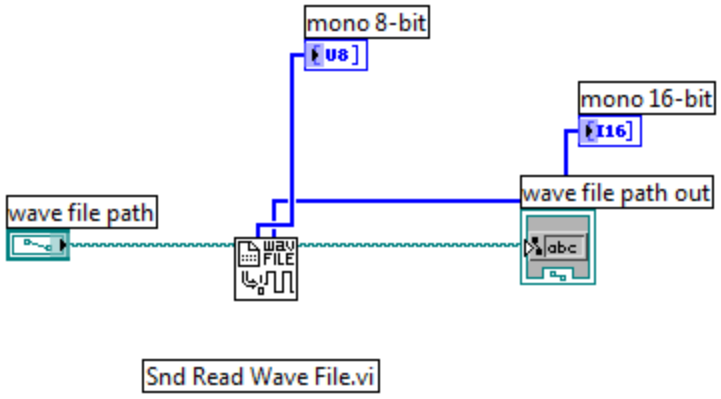
Figure 3.10 Block diagram text to speech Synthesis.

3.4.2.2 Play Speech in Wave File Player

This VI illustrates how to playback a wave file using sound board, information such as file name, sound quality rate & bits/sample are also displayed. We are using this VI to listen the speech generated by text to speech conversion. The Flowchart for wave file player.vi is shown in Figure 4.3. Various Functions used in wave file player.vi are described here.

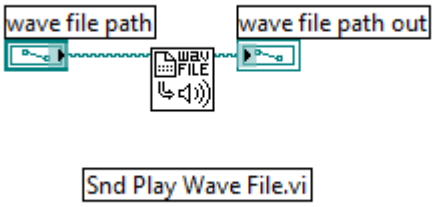
3.4.2.2.1 Snd Read Wave File

Retrieves a PC wave file (.wav) specified in **wave file path**. The information returned includes both waveform data and sound format data, which is necessary for configuring a sound device to play the waveform.



3.4.2.2.2 Snd Play Wave File

Plays a sound file from disk. Windows This VI plays the sound file. If wait until finish is set to TRUE, it waits until the sound finishes playing. If wait until finish is set to FALSE, it continues executing the VI. Mac OS Only supports uncompressed wave files. This VI always waits until the sound finishes playing to continue executing.



Block diagram of wave file player is shown below in Figure 3.11.

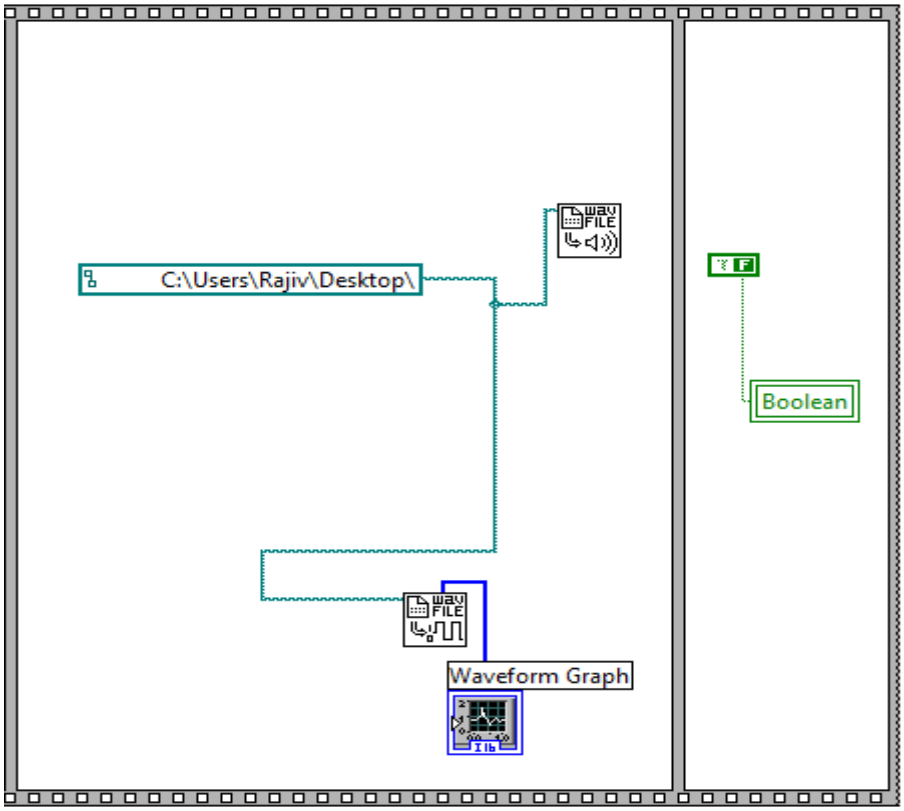


Figure 3.11 Block diagram of wave file player.

Flowchart for wave file player.vi is shown below in Figure 3.11.

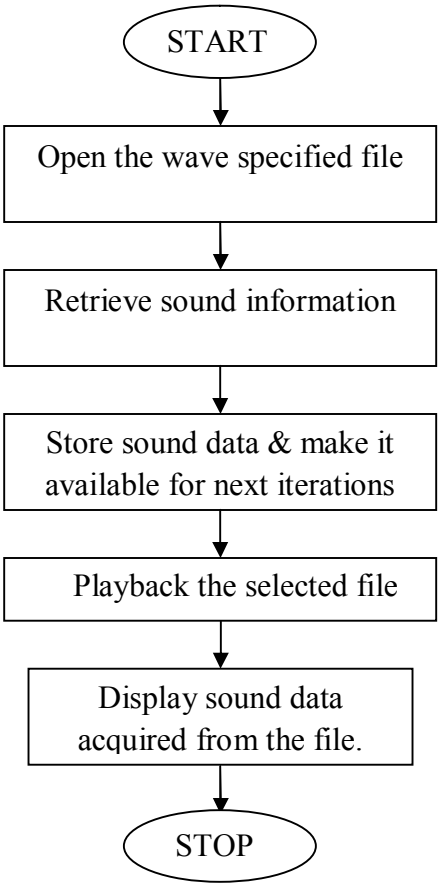


Figure 3.12 Flow chart of wave file player.vi

Chapter 4

Results and Discussion

4.1 Introductio

Experiments have been performed to test the proposed system. Here whole system is implemented using LabVIEW 7.1 version. The front panel of OCR based speech recognition system is shown below in Figure 4.1.

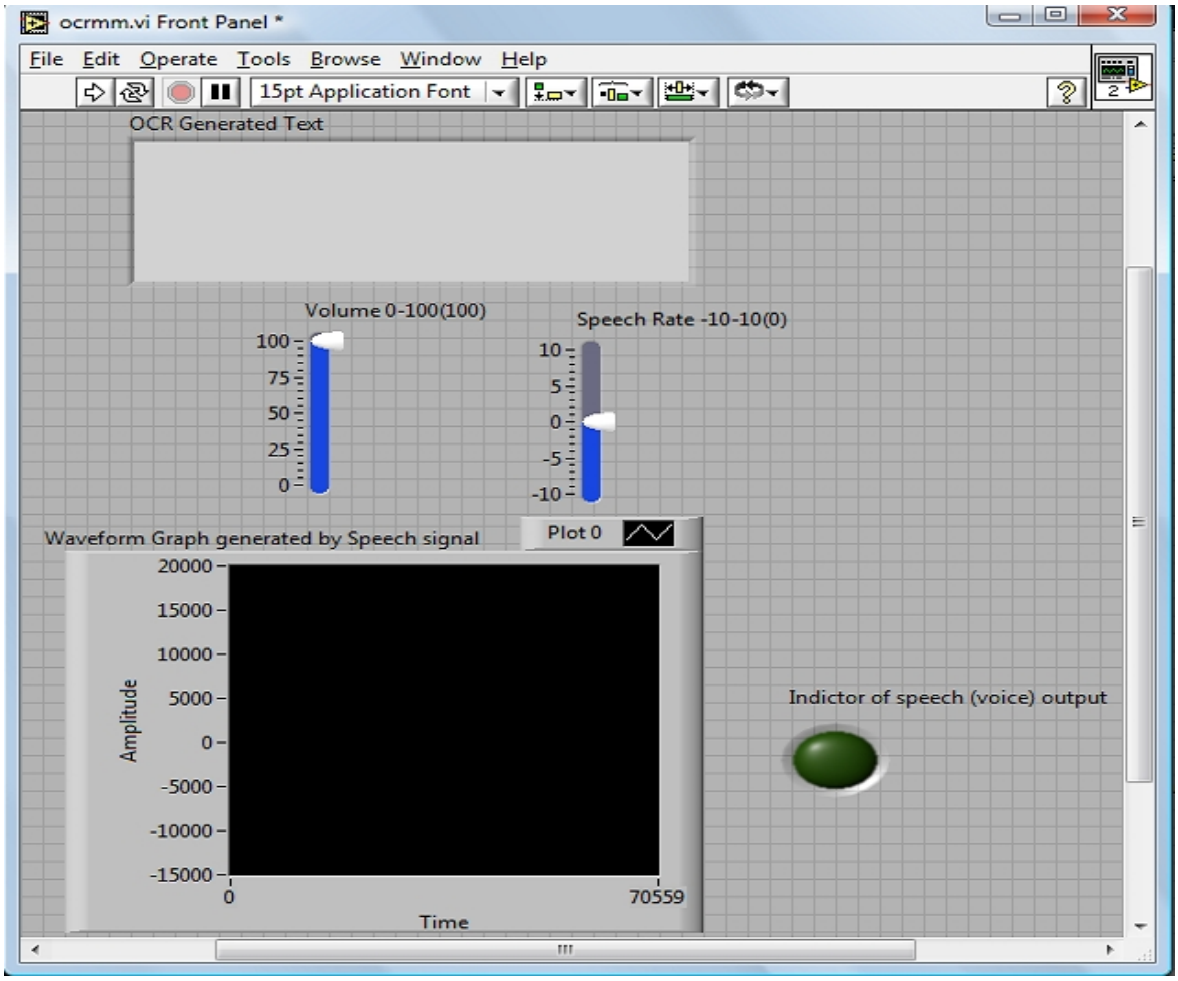


Figure 4.1 Front panel of OCR based speech recognition system.

LabVIEW software of OCR based speech recognition system is having two steps:

i) Optical Character Recognition

ii) Speech Synthesis

4.2 Optical Character Recognition

STEP 1: When the scanner scans the printed text and these texts are stored in an image file. In the first step of result the output of IMAQ ReadFile reads an image file, this image file will be display by using IMAQ WindDraw (Displays an image in an image window) at the output of IMAQ Read File. Image window is shown below.

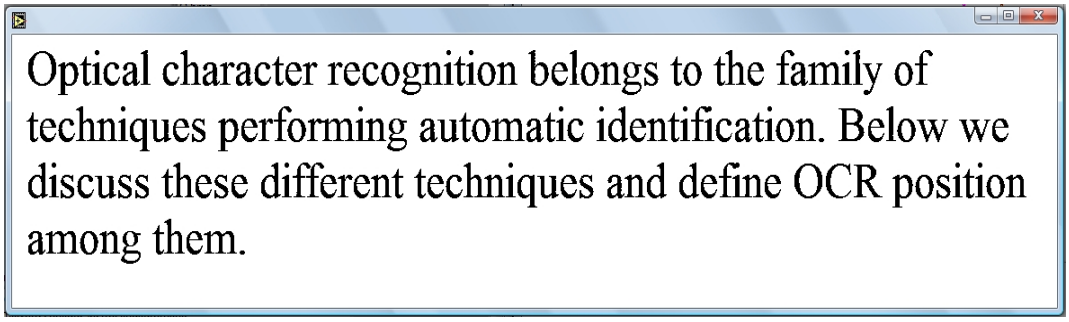


Figure 4.2 Read an image.

STEP 2: Applies a threshold (175 assumed) to an image and IMAQ Inverse, Image window is shown below.

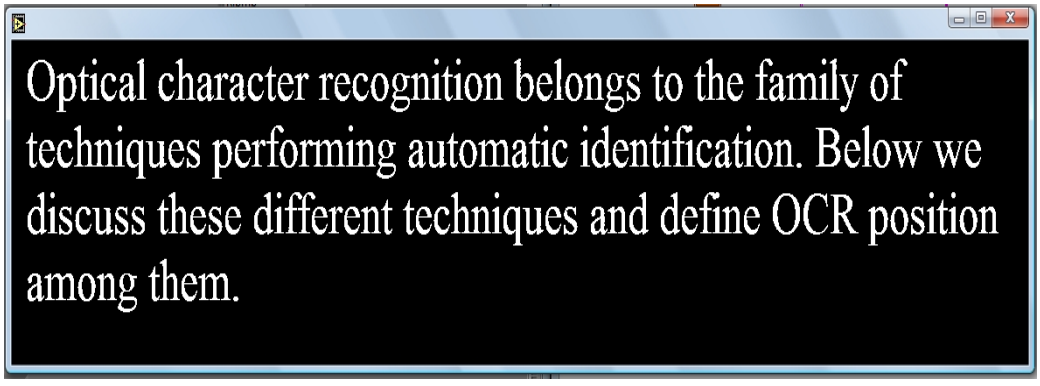


Figure 4.3 Image after thresholding and inverting.

STEP3: In this step line segmentation of thresholded image find from above step has been done. Figure 4.4 shows line segmentation process result.

Figure 4.4 Image window segmented line.

STEP 4: In this step word are segmented from a particular line. Figure 4.5 shows word segmentation process result.

Figure 4.5 Image window segmented word.

STEP 5: This step applies segmentation process by which all the chacter in above image window are segmentated, the segmenatation first three character of word ”Optical” is shown below in image window.

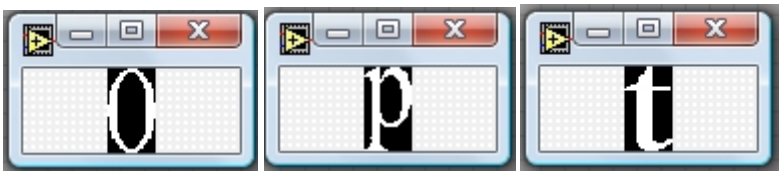


Figure 4.6 Image window segmented character image.

After segmentation each character correlate with stored character templates, and recognition of printed text is done by this OCR software system.

STEP 6: Finally the output of OCR system is in text format which can stored in a computer system. The result of recognise text will be shown on Front panel. So the final result of OCR is shown bellow in figure 4.7.

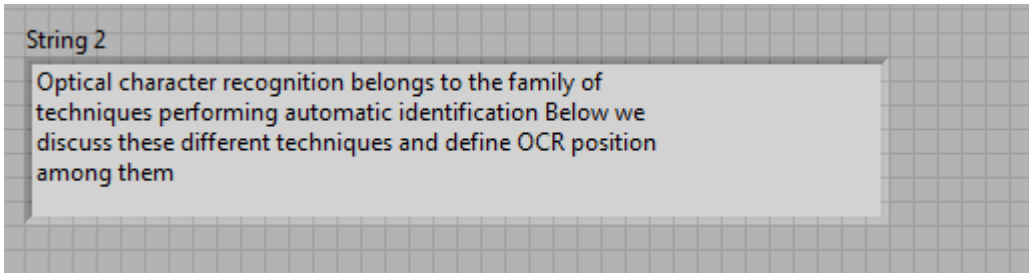


Figure 4.7 Final result of OCR system.

4.3 Speech Synthesis

A wave file output.wav is created containing text converted into speech which can listen using wave file player.

The waveform will vary according to the different text from OCR output in the text box and can be listened on the speaker. The wave form for above recognize text is shown in figure 4.8

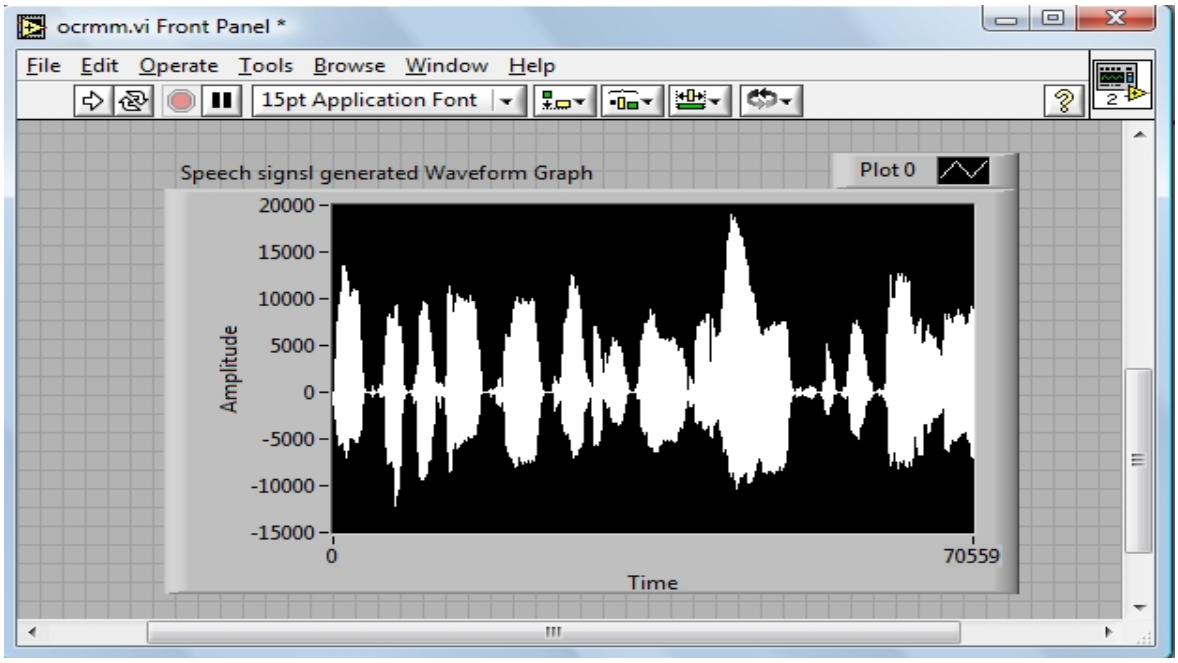


Figure 4.8 Output of file wave player.

Chapter 5

Conclusion and Future Scope

5.1 Conclusion

This thesis work describes OCR based Speech Synthesis System to produces a wave file output can be used as a good mode of communication between people. The system is implemented on LabVIEW 7.1 platform. There is two session of system first is *OCR* and second is *Speech Synthesis*. In OCR printed or written character documents are scanned and image is acquired by using IMAQ Vision for LabVIEW and then characters are recognized using segmentation and template matching methods developed in LabVIEW. In second section recognised text is converted into speech. The ACTIVE X sub pallet in Communication pallet is used to exchange data between applications. ActiveX technology provides a standard model for interapplication communication that different programming languages can implement on different platforms. *Microsoft Speech Object Library (Version 5.1)* has been used to build speech-enabled applications, which retrieve the voice and audio output information available for computer. This library allows selecting the voice and audio device one would like to use, OCR recognized text to be read, and adjust the rate and volume of the selected voice.

The application developed is user friendly, cost effective and gives the result in the real time. Moreover, the program has the required flexibility to be modified easily if the need arises.

5.2 Future Scope

OCR base Speech recognition system using LabVIEW is an efficient program giving good results for specific fonts (equal to or above to 48 font size) still there are chances to improve it. The system can be improved by making it omnifont. Because there was no

OCR base Speech recognition system implemented using Lab VIEW there is a good future scope to develop it using other methods more fast and efficient.

References

1. T. Dutoit, "High quality text-to-speech synthesis: a comparison of four candidate algorithms," Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on, vol.i, no., pp.I/565-I/568 vol.1, 19-22 Apr 1994.
2. B.M. Sagar, Shobha G, R. P. Kumar, "OCR for printed Kannada text to machine editable format using database approach" WSEAS Transactions on Computers Volume 7, Pages 766-769, 6 June 2008.
3. G. Nagy, "At the frontiers of OCR," Proceedings of the IEEE, vol.80, no.7, pp.1093-1100, Jul 1992.
4. Landt, Jerry. "Shrouds of Time: The history of RFID," AIM, Inc., 31 May 2006.
5. R.C. Palmer, "The Bar Code Book," Helmers Publishing.
6. Mandell, Lewis. "Diffusion of EFTS among National Banks: Note", Journal of Money, Credit and Banking Vol. 9, No. 2, May, 1977.
7. Bergeron, P. Bryan (1998, August). "Optical mark recognition," Postgraduate Medicine online. June 7, 2006.
8. I. Witten, "Principles of Computer Speech.", Academic Press Inc., 1982.
9. K. Kleijn, K. Paliwal (Editors). "Speech Coding and Synthesis," Elsevier Science B.V., The Netherlands, 1998.
10. Y. Sagisaga, "Speech Synthesis from Text," 1998.
11. J. Flanagan, "Speech Analysis, Synthesis, and Perception," Springer-Verlag, Berlin-Heidelberg-New York, 1972.
12. D. O'Saughnessy. "Speech Communication - Human and Machine," Addison-Wesley, 1987.
13. G. Fant, "Acoustic Theory of Speech Production. Mouton," The Hague, 1970.
14. A. Breen, E. Bowers, W. Welsh. "An Investigation into the Generation of Mouth Shapes for a Talking Head," Proceedings of ICSLP 96 (4), 1996.

15. R. Donovan, “Trainable Speech Synthesis,” PhD. Thesis. Cambridge University Engineering Department, England, 1996. <ftp://svr-ftp.eng.cam.ac.uk/pub/reports/donovan_thesis.ps.Z>.
16. T. Altosaar, M. Karjalainen, M. Vainio. “A Multilingual Phonetic Representation and Analysis for Different Speech Databases,” Proceedings of ICSLP 96 (3), 1996.
17. W. Hallahan. “DECtalk Software: Text-to-Speech Technology and Implementation,” Digital Technical Journal, 1996.
18. G. Cawley, Noakes B. “Allophone Synthesis Using a Neural Network,” Proceedings of the First World Congress on Neural Networks (WCNN-93) (2): 122-125, 1993.
19. N. Arica; F.T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol.31, no.2, pp.216-233, May 2001.
20. S. Mori; C.Y. Suen; K. Yamamoto, "Historical review of OCR research and development," Proceedings of the IEEE, vol.80, no.7, pp.1029-1058, Jul 1992.
21. Smith R., "An Overview of the Tesseract OCR Engine," Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on, vol.2, no., pp.629-633, 23-26 Sept. 2007.
22. Jisheng Liang; I.T Phillips.; V. Chalana; R. Haralick, "A methodology for special symbol recognitions," Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol.4, no., pp.11-14 vol.4, 2000.
23. E. Nunes; E. Abreu; J.C. Metrolho; N. Cardoso; M. Costa; E. Lopes, "Flour quality control using image processing," Industrial Electronics, 2003. ISIE '03. 2003 IEEE International Symposium on , vol.1, no., pp. 594-597 vol. 1, 9-11 June 2003.
24. Y.M. Alginahi., "Thesholding and Character Recognition in Security Documents with Watermarked Background," Computing: Techniques and Applications, 2008. DICTA '08.Digital Image, vol., no., pp.220-225, 1-3 Dec. 2008.
25. C. Mancas-Thillou; B. Gosselin, "Character Segmentation-by-Recognition Using Log-Gabor Filters," Pattern Recognition, 2006. ICPR 2006. 18th International Conference on , vol.2, no., pp.901-904, 0-0 0
26. C.Y. Suen; M. S. Berthod; Mori, "Automatic recognition of handprinted characters—The state of the art," Proceedings of the IEEE, vol.68, no.4, pp. 469-487, April 1980.

27. M. Sarfraz; S.A. Shahab, "An efficient scheme for tilt correction in Arabic OCR system," Computer Graphics, Imaging and Vision: New Trends, 2005. International Conference on, vol., no., pp. 379-384, 26-29 July 2005.
28. S. Mori; C.Y. Suen; K. Yamamoto, "Historical review of OCR research and development," Proceedings of the IEEE, vol.80, no.7, pp.1029-1058, July 1992.
29. N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. Syst. Man Cybern. 9 (1979) (1), pp. 62–66. View Record in Scopus | Cited By in Scopus (2626), IEEE Transactions on Systems, Man, and Cybernetics, VOL. SMC-9, NO. 1, Jan. 1979.
30. A. B.S. Hussain, G.T. Toussaint, and R. W. Donaldson, "Results obtained using a simple character recognition procedure on Munson's hand printed data," IEEE Trans Comput., vol. 21, pp. 201-205, Feb. 1972.
31. A. A. Spanjersberg, "Combinations of different systems for the recognition of handwritten digits," Proc. 2n. Joint Conf. Pattern Recognition, pp. 208-209, Aug. 1974.
32. R. Bornat and J. M. Brady, "Using knowledge in the computer interpretation of handwritten FORTRAN coding sheets," Inc. J. Man-Mach. Studies, vol. 8, pp. 13-27, 1976.
33. R. Bakis, N. M. Herbst, and G. Nagy, "An experimental study of machine recognition of hand-printed numerals," IEEE Trans. Syst. Sci. Cybern., vol. 4, pp. 119-132, July 1968.
34. H. A. Glucksman, "Multicategory classification of patterns rep- resented by high-order vectors of multilevel measurements," IEEE Trans. Comput., vol. 20, pp. 1593-1598, Dec. 1971.
35. J. T. Tou and R. C. Gonzalez, "A new approach to automatic recognition of handwritten characters," in Proc. Two-Dimensional Digital Signal Processing Conf., pp. 3-2-1-3-2-10, Oct. 1971.
36. Dutoit, Thierry. "An Introduction to Text-To-Speech Synthesis," Boston: Kluwer Academic Publishers, 1997.

37. Wu Chung-Hsien., & Chen Jau-Hung. Automatic generation of synthesis units and prosodic information for Chinese concatenative synthesis. Speech Communication, 35, 219-237, 2001.

38. M. Edgington, A. Lowry, P. Jackson, A.P., Breen & S. Minnis. "Overview of current text-to-speech techniques: Part I – text and linguistic analysis," BT Technology Journal. Vol.14 No.1, 1996.

39. D. Klatt, "Review of Text-to-Speech Conversion for English," Journal of the Acoustical Society of America, JASA vol. 82 (3), pp.737-793, 1987.

40. J. Allen, S. Hunnicutt, D. Klatt. "From Text to Speech," The MITalk System. Cambridge University Press, Inc., 1987.
41. E. Abadjieva, I. Murray, J. Arnott. "Applying Analysis of Human Emotion Speech to Enhance Synthetic Speech," Proceedings of Eurospeech 93 (2): 909-912, 1993.

42. I. Murray, J. Arnott, N. Alm, "A. Newell. A Communication System for the Disabled with Emotional Synthetic Speech Produced by Rule," Proceedings of Eurospeech 91 (1): 311-314, 1991.
43. B. Kröger. "Minimal Rules for Articulatory Speech Synthesis," Proceedings of EUSIPCO92 (1): 331-334, 1992.
44. W. Holmes, J. Holmes, M. Judd. "Extension of the Bandwith of the JSRU Parallel-Formant Synthesizer for High Quality Synthesis of Male and Female Speech," Proceedings of ICASSP 90 (1): 313-316, 1990.
45. U. Laine. PARCAS, "a New Terminal Analog Model for Speech Synthesis," Proceedings of ICASSP 82 (2) 1982.
46. H. Hon, A. Acero, X. Huang, J. Liu, M. Plumpe. "Automatic Generation of Synthesis Units for Trainable Text-to-Speech Systems," Proceedings of ICASSP 98 (CD-ROM), 1998.
47. R. Kortekaas, A. Kohlrausch. "Psychoacoustical Evaluation of the Pitch-Synchronous Overlap-and-Add Speech-Waveform Manipulation Technique Using Single-Formant

Stimuli," Journal of the Acoustical Society of America, JASA, Vol. 101 (4): 2202-2213, 1997.

48. F. Charpentier, E. Moulines. "Pitch-Synchronous Waveform Prosessing Techniques for Text-to-Speech Synthesis Using Diphones," Proceedings of Eurospeech 89 (2): 13-19, 1989.
49. M. Macon, C. Clements. "Speech Concatenation and Synthesis Using an Overlap-Add Sinusoidal Model," Proceedings of ICASSP 96: 361-364, 1996.
50. http://zone.ni.com/devzone/cda/tut/p/id/2983.