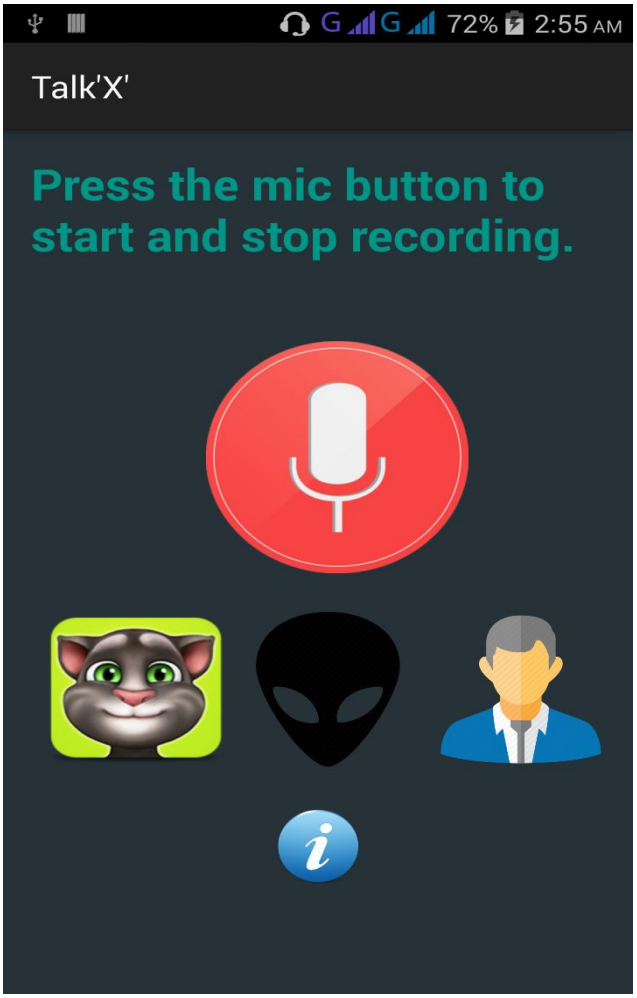
Voice morphing based on UP/DOWN sampling implemented on android device



Project by,
Aravind Kumar-2013105004,
Ashvath Nirmal-2013105005,
Bharath-2013105006.

Table of contents:

Introduction:

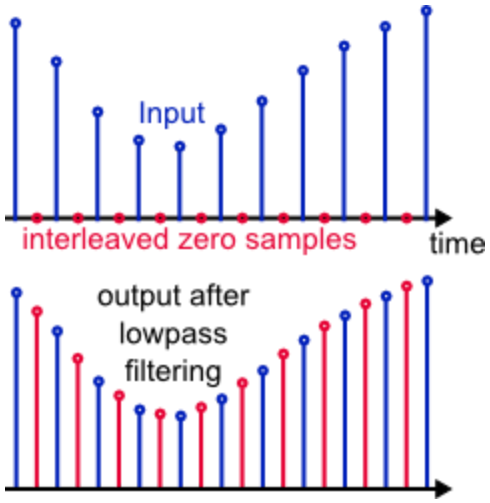
Voice morphing is a fun to play with from talking tom to making pranks with friends changing male to female voice and vice versa.There are several ways to transform voice from one to another/morph voices, in this project we have used a lame method of upsampling and downsampling to change the voices.Every aspect of the project deals with only the time domain part and everything is constrained in this domain.This project gives option to morph to two pre-defined voices, **TOM-cat** and **Alien**.The whole project is made into an android app, which runs a piece of patch called **PURE DATA** in the background and it takes care of the signal processing aspect of the project, and it will be explained later.

DSP Principle:

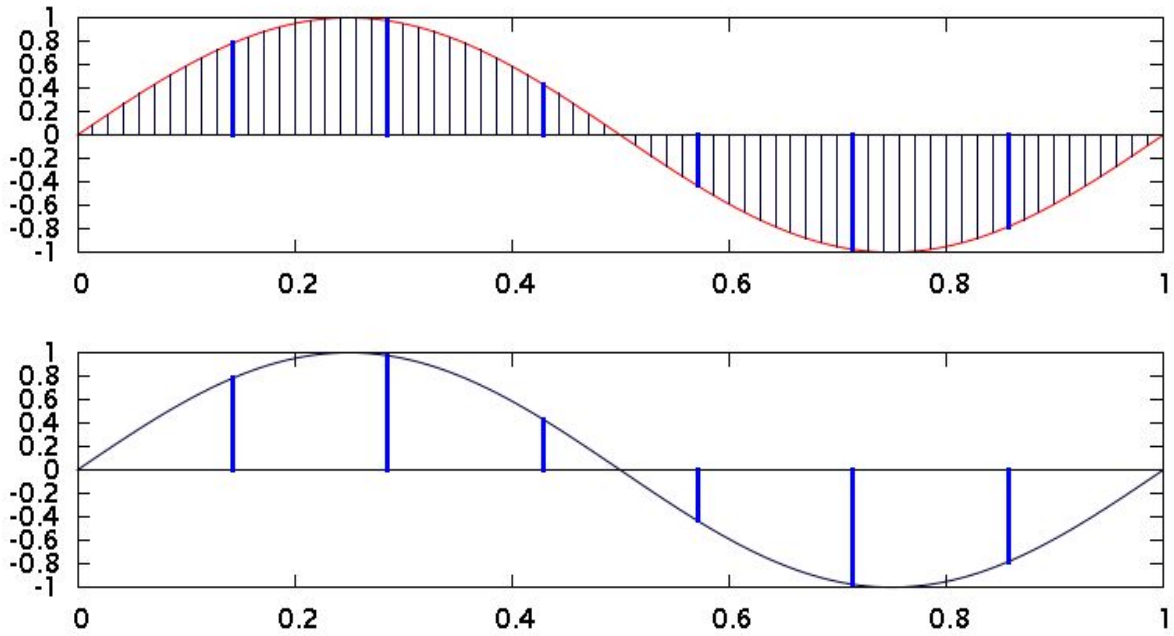
The process of upsampling and downsampling is used to morph the voices.

Upsampling is used to transform the voice into an alien voice and downsampling is used to transform the voice into tom voice. Now both the process will be briefed.

Upsampling is the process of inserting zero-valued samples between original samples to increase the sampling rate.This is called "zero-stuffing".Upsampling adds to the original signal undesired spectral images which are centered on multiples of the original sampling rate.So this is a method of **stretching out** the audio signal in time domain and the undesired spectral images are beyond the scope of this project.



Decimation is the process of reducing the sampling rate.In practice, genrally this implies low pass filtering a signal and throwing away some of its samples.**Downsampling** is more specific term just refers to throwing away the samples without low pass filtering the signal.For our project we assume it **shrink** the audio signal in time domain.



Thus using upsampling the audio can stretched out and it can be made to hear like alien and downsampling the audio can shrink the audio in time domain and can be made to hear like tom-cat.

Pure Data:

Pure Data is an DSP implementation software which can run on PC,Mac,Android and even iOS.

Here we have used this software as backend to do the audio signal processing and android as frontend.

Pure Data is an open source software and everything is based on patches and blocks.Below is the pure data patch,

The patch is explained in the preceding paragraph,

- Initially the voice is converted to digital using adc~ and as the human speech ranges from 300hz to 3400 hz only those frequencies are bandpass filtered and other than these are just pure noise and rejected out of the saved audio.
- After filtering the audio it is saved on the mobile's sd card under the name rec.wav,pure data samples like all other at 44100samples/sec.
- For human voice the recorded .wav file is read into an array and the output is given to dac~, which converts digital to analog audio.

- For the tom voice the recorded .wav file is downsampled and given to dac~, e.g, if an audio is 2sec duration , then the audio is played only for 1sec where some of the samples are left out and we perceive that audio as **shrunken version** of the original audio.
- For the alien voice he recorded .wav file is upsampled and given to the dac~,e.g, if an audio is 2sec duration , then the audio is played for 4sec where zeros samples are padded and we perceive that audio as **elongated version** of the original audio.
- Thus the functions of patch are explained and in the future section we will be discussing about integrating it with the android studio and how to make it as an android application.

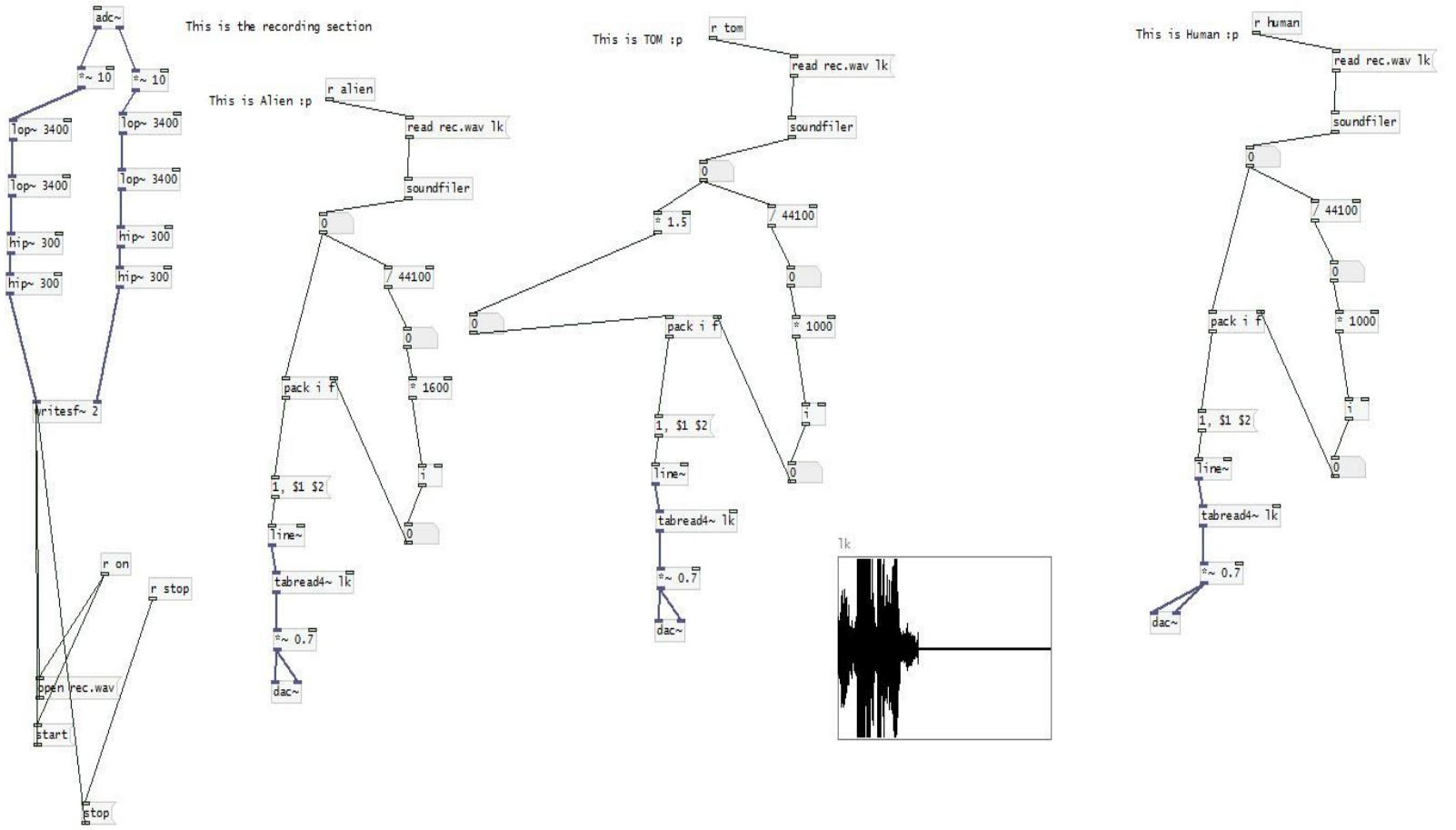
Android implementation:

- Android studio is used to build android applications and it is based on java platform.
- Pure Data has a set of libraries which can be synced with the android studio and the patch can be run at the background of the app.

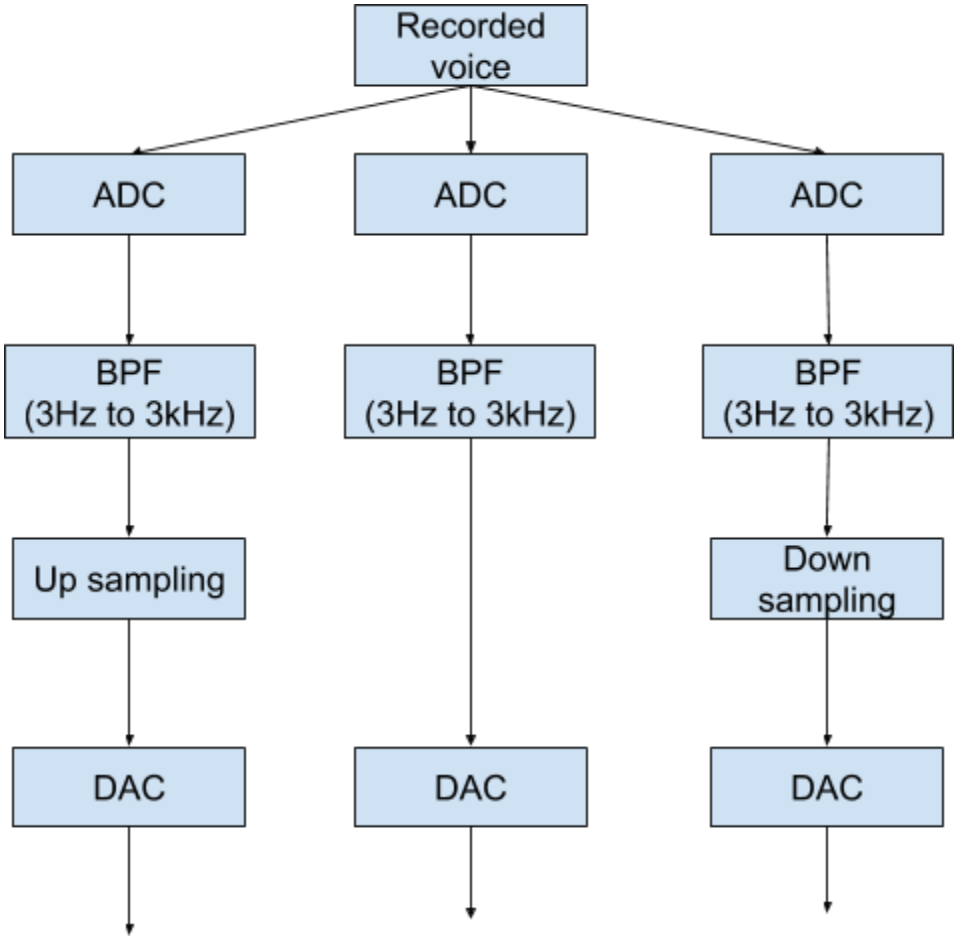
- The finished pure data patch is compressed and zipped in a file and is placed under the res/raw directory of the application what we are currently working on.
- The input mic channels and output channels are set in the android studio by which the patch gets initialised.
- The android UI elements are made using the XML language and the actions for the buttons are coded using the MainActivity,java file.
- The receiver objects in the PD patch can receive the float values from android app.
- The app is thus programmed so as to send appropriate send float values to the patch running in the background.
- And by following the above procedure, app is made.

Working:

- Press the mic icon on the screen to start and stop recording
- After recording the audio, press mic again to stop recording.
- Press any of the three button below mic to play the recorded audio in tom's,alien's and the original voice.


adc~
*~ 10
*~ 10
lop~ 3400
lop~ 3400
lop~ 3400
lop~ 3400
hip~ 300
hip~ 300
hip~ 300
hip~ 300
writesf~ 2
r on
r stop
open rec.wav
start
stop
This is the recording section
This is Alien :p
r alien
read rec.wav lk
soundfiler
0
/ 44100
0
pack i f
* 1600
1, $1 $2
line~
tabread4~ lk
*~ 0.7
dac~
This is TOM :p
r tom
read rec.wav lk
soundfiler
0
* 1.5
/ 44100
0
0
pack i f
* 1000
1, $1 $2
line~
tabread4~ lk
*~ 0.7
dac~
lk
This is Human :p
r human
read rec.wav lk
soundfiler
0
/ 44100
0
pack i f
* 1000
1, $1 $2
line~
tabread4~ lk
*~ 0.7
dac~

Block Diagram:



Alien Voice **Human Voice** **Cat Voice**

Thank you