realtime vocal formant detection using web audio API

+AWESOME interactive graphics

Can the algorithm underlying biofeedback therapy for speech be replicated in JavaScript?

{

For my final project I would like to attempt to recreate an LPC-algorithm in javascript, which can be used to analyze audio input from a microphone and output coefficients for use in the creation of original real time graphics.

ok... but WHY?

I'm currently writing an HTML5 GUI for a speech lab, which hopes to produce a mobile application-based research/treatment tool for speech therapy in /r/ misarticulation. I am <u>not</u> part of the acoustics team on this project. My role is simply to mark-up the graphics.

HOWEVER, as a web-enthusiast, I think we may be able to recreate our LPC algorithm using javascript and the web audio api. By moving the core analysis of this tool from Objective-C (iOS and Apple App Store -dependent) to web languages, we may be able to **create a more accessible, cross-platform tool.**

#research_context

/R/ is the most difficult sound in the English language to master and the most frequently reported residual articulatory error encountered by speech therapists.

In the absence of developmental or anatomical abnormalities, the etiology of residual articulatory errors is generally unknown and **treatment methods are limited**. Treatment for /r/ misarticulation is especially difficult because clinicians are unable to provide visual cues (all of the muscle constriction takes place inside your mouth and throat) and this sound lacks strong tactile-kinesthetic cues for the speaker.

It is generally thought that residual errors are (to some degree) the result of problems perceiving the American /r/ sound. If this etiology is correct, then **a motor-learning approach using visual biofeedback as reinforcement may prove therapeutically advantageous.**

#lab-based_investigation {

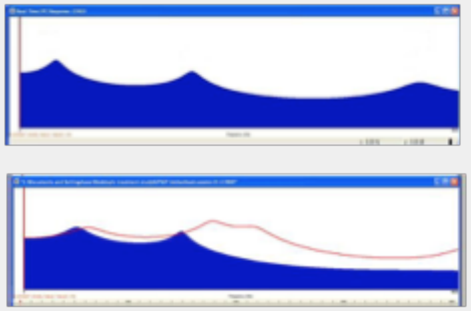
Can a visual biofeedback reinforced motor-learning approach to speech sound disorder therapy produce clinical outcomes equal to or greater than traditional therapy methods?

Currently, my PI is testing this question by using real-time visualizations of speech to teach individuals to manipulate their vocal output to match a graphic model representing correct pronunciation. Over time, individuals learn to use their vocal output to control the lab-based speech wave visualization. Through practice matching personal vocal output to the visual model, an individual can relearn how a correctly pronounced /r/ should feel when speaking.

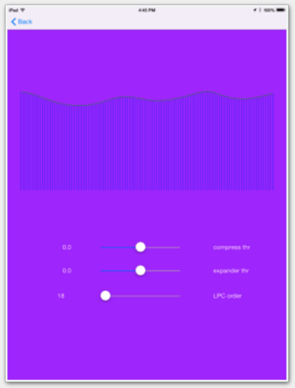
To test the effectiveness of this therapy, the speech lab would like to expand the sample size of their study by producing an iPad-based version of the speech wave visualization and biofeedback therapy training exercises, which can be distributed to Speech Language Pathologists across the country.

#personal_motivation {

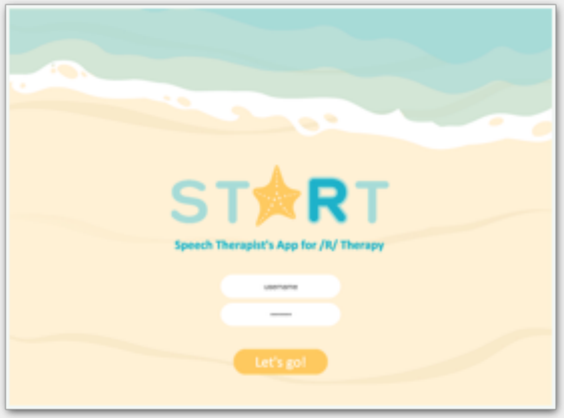
As the the visual designer on this project, I would like a little **more control over the display of the speech visualization.** The LPC aspect of the app is currently being developed by a separate team and is written in Objective-C. This language difference can create technological obstacles when trying to integrating the display of their algorithm with the changing aesthetics of the app. Similarly, the language disparity may produce issues for deployment.



current lab-based visualization



current app-based LPC display



current HTML5 GUI

#learning_objectives {

1

I'm not a musician or audio engineer, but **I'd like to learn a little more about sound.**

By researching the LPC algorithm, I hope to learn a little about how speech is measured, how formants and resonance are detected, and the pros and cons of various speech analysis methodologies.

2

As a designer, I view this project as an opportunity to experiment with interactivity (real time audio input) and learn a little about graphics programming in the language of the world's greatest forum – the Internet!

Specially, I would like the opportunity to experiment with the three.js, tone.js, and p5.js libraries.

#methods_and_timeline {

1

months

2

Code

- Research LPC algorithm and js libraries for audio via github
- Meet with Luke DuBois to discuss algorithm development and any issues understanding audio signal processing
- Gain a better understanding of web audio and js programming
  (my current js experience is solely UI or display-based)

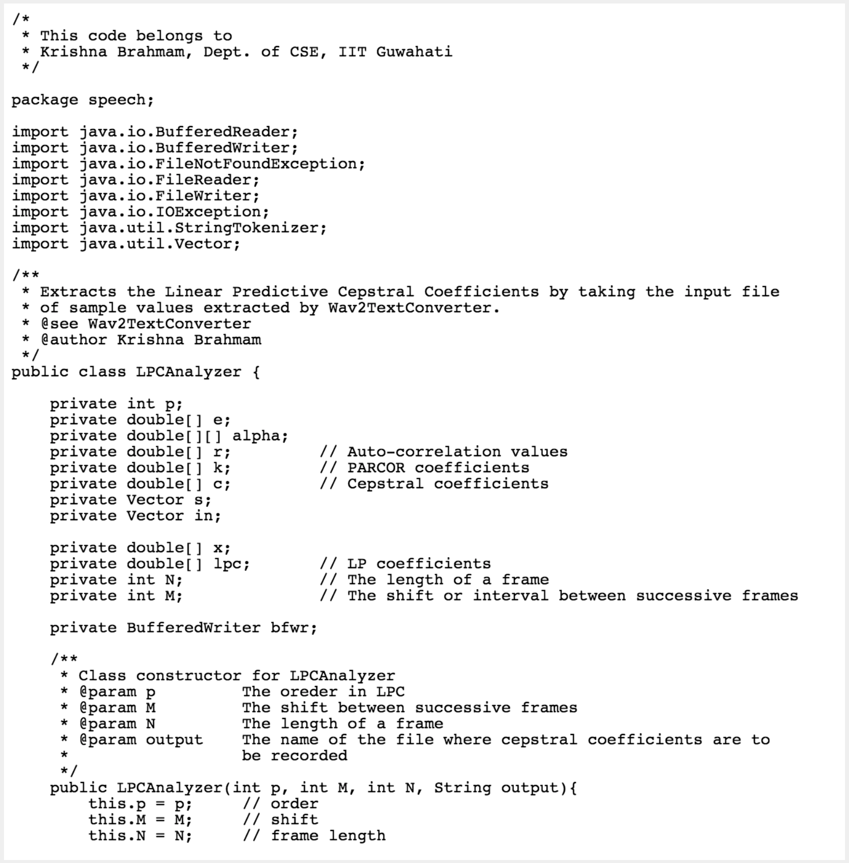
Graphics

- Research three.js, two.js, and p5.js
- Research visual representations of sound (speech & music)
- Make beautiful, responsive graphics based voice input!!!

#inspiration_code {

LPC algorithm written in Java

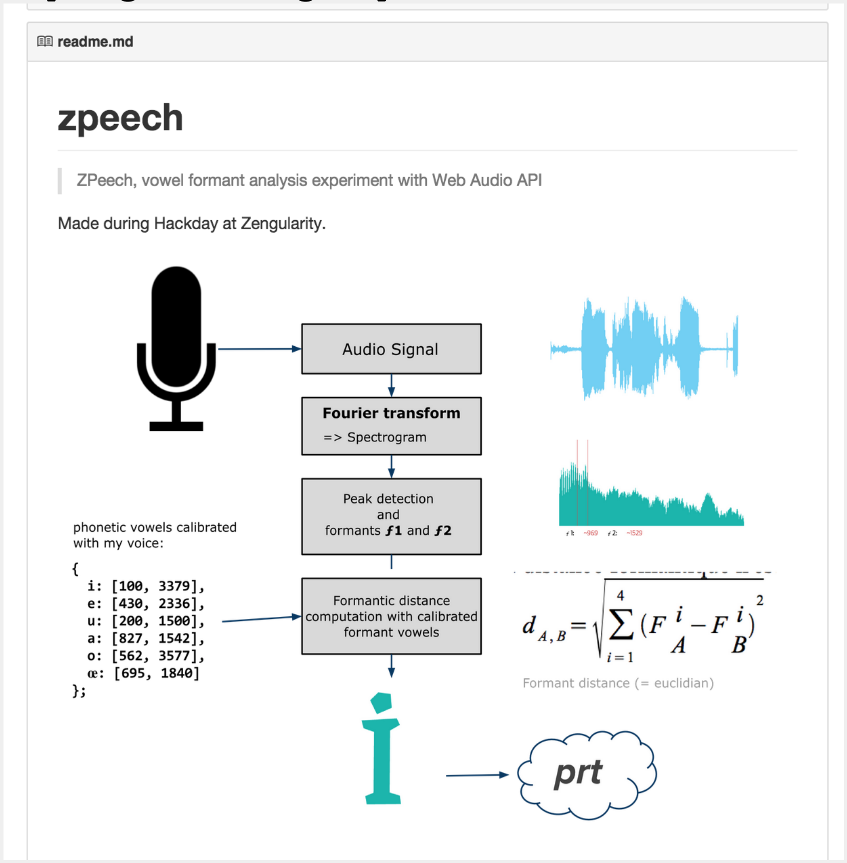
http://speech-recog.googlecode.com/svn/trunk/Speech/src/speech/LPCAnalyzer.java



```
/*
 * This code belongs to
 * Krishna Brahmam, Dept. of CSE, IIT Guwahati
 */

package speech;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.StringTokenizer;
import java.util.Vector;

/**
 * Extracts the Linear Predictive Cepstral Coefficients by taking the input file
 * of sample values extracted by Wav2TextConverter.
 * @see Wav2TextConverter
 * @author Krishna Brahmam
 */
public class LPCAnalyzer {

    private int p;
    private double[] e;
    private double[][] alpha;
    private double[] r;         // Auto-correlation values
    private double[] k;         // PARCOR coefficients
    private double[] c;         // Cepstral coefficients
    private Vector s;
    private Vector in;

    private double[] x;
    private double[] lpc;       // LP coefficients
    private int N;              // The length of a frame
    private int M;              // The shift or interval between successive frames

    private BufferedWriter bfwr;

    /**
     * Class constructor for LPCAnalyzer
     * @param p         The oreder in LPC
     * @param M         The shift between successive frames
     * @param N         The length of a frame
     * @param output    The name of the file where cepstral coefficients are to
     *                  be recorded
     */
    public LPCAnalyzer(int p, int M, int N, String output){
        this.p = p;     // order
        this.M = M;     // shift
        this.N = N;     // frame length
```

Experimental vowel formant analysis program in js

https://github.com/gre/zpeech

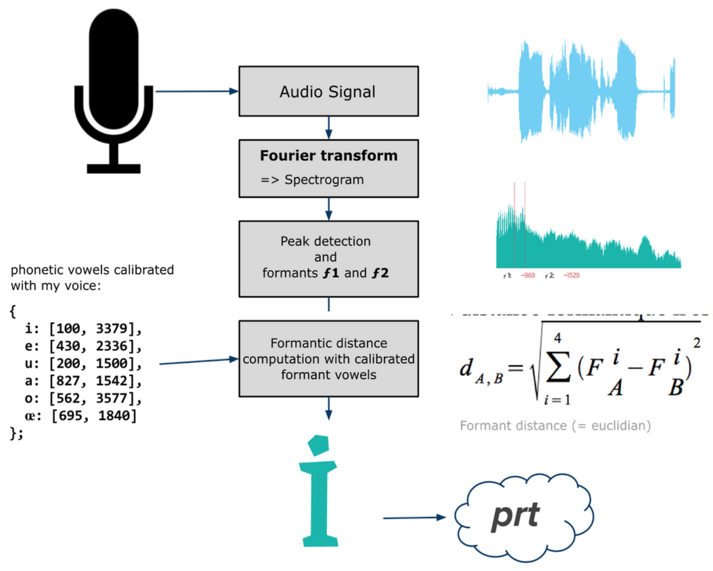


readme.md

zpeech

> ZPeech, vowel formant analysis experiment with Web Audio API

Made during Hackday at Zengularity.



Audio Signal
Fourier transform
=> Spectrogram
Peak detection
and
formants ƒ1 and ƒ2
Formantic distance computation with calibrated formant vowels
phonetic vowels calibrated with my voice:
i
prt
Formant distance (= euclidian)

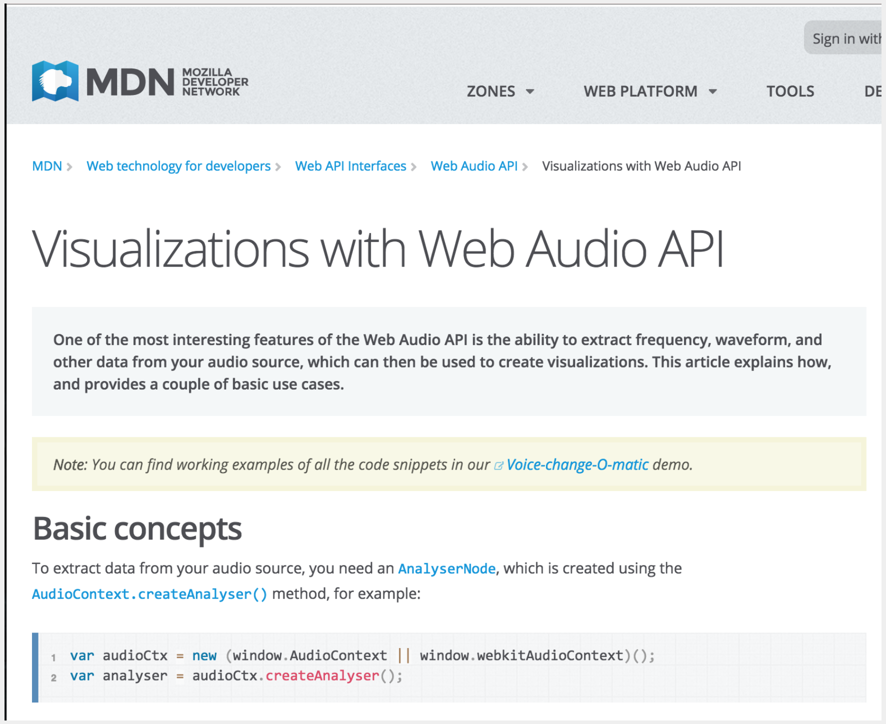
```
{
  i: [100, 3379],
  e: [430, 2336],
  u: [200, 1500],
  a: [827, 1542],
  o: [562, 3577],
  œ: [695, 1840]
};
```

$$d_{A,B} = \sqrt{\sum_{i=1}^{4} (F_A^i - F_B^i)^2}$$

#inspiration_tutorials

Web Audio API Tutorial

https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API/Visualizations_with_Web_Audio_API



MDN > Web technology for developers > Web API Interfaces > Web Audio API > Visualizations with Web Audio API

Visualizations with Web Audio API

One of the most interesting features of the Web Audio API is the ability to extract frequency, waveform, and other data from your audio source, which can then be used to create visualizations. This article explains how, and provides a couple of basic use cases.

Note: You can find working examples of all the code snippets in our Voice-change-O-matic demo.

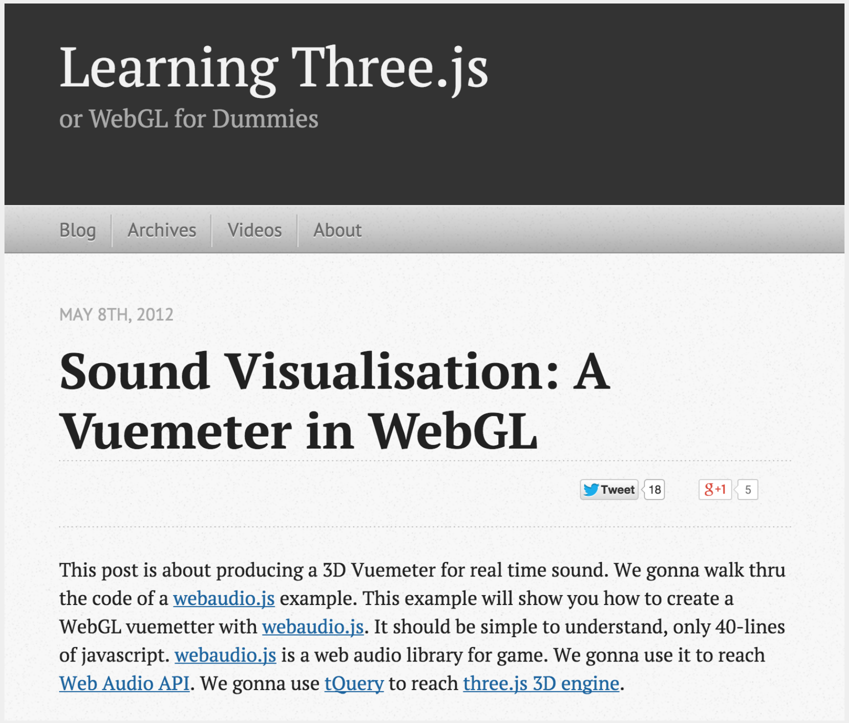
Basic concepts

To extract data from your audio source, you need an `AnalyserNode`, which is created using the `AudioContext.createAnalyser()` method, for example:

```
var audioCtx = new (window.AudioContext || window.webkitAudioContext)();
var analyser = audioCtx.createAnalyser();
```

Web Audio + Three.js Tutorial

http://learningthreejs.com/blog/2012/05/08/sound-visualisation-vuemeter-in-webgl/



MAY 8TH, 2012

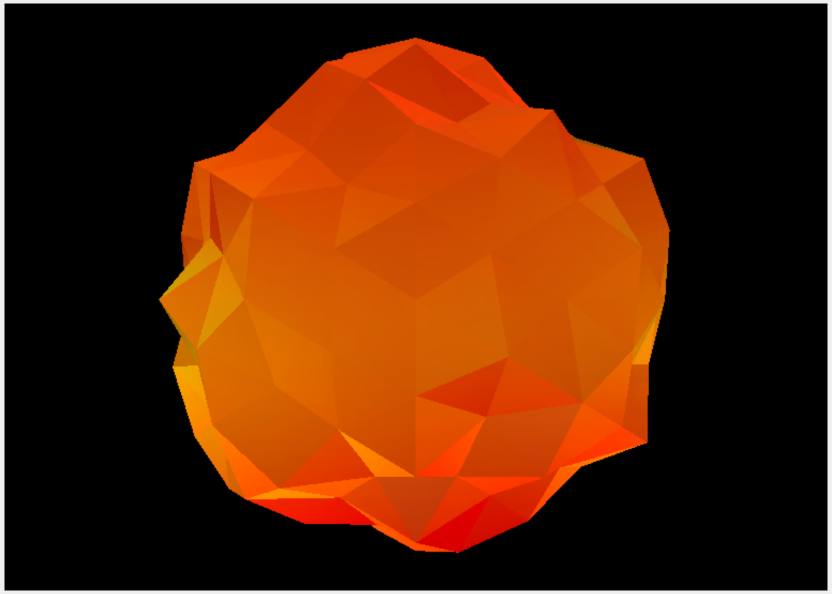
Sound Visualisation: A Vuemeter in WebGL

This post is about producing a 3D Vuemeter for real time sound. We gonna walk thru the code of a webaudio.js example. This example will show you how to create a WebGL vuemetter with webaudio.js. It should be simple to understand, only 40-lines of javascript. webaudio.js is a web audio library for game. We gonna use it to reach Web Audio API. We gonna use tQuery to reach three.js 3D engine.

#inspiration_graphics

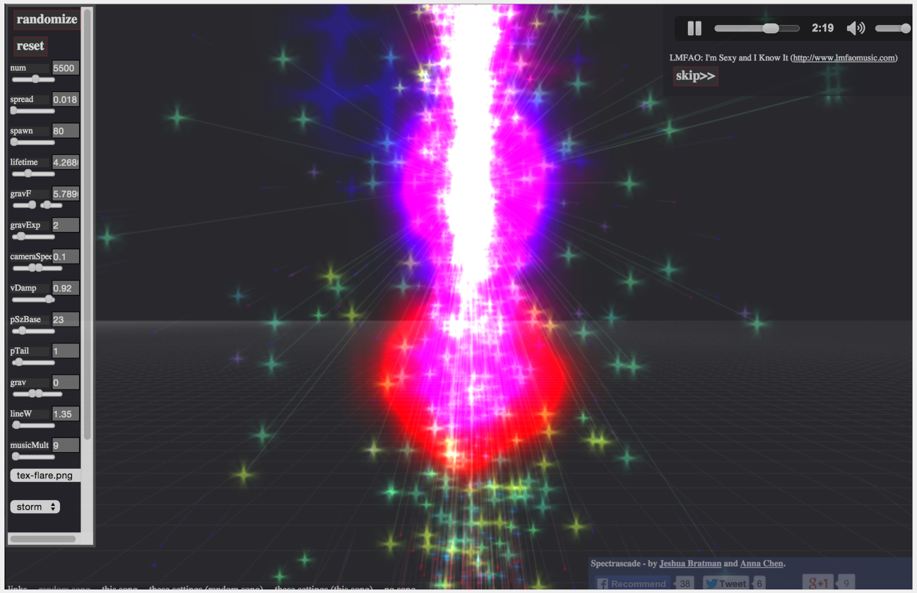
Visualization from mic input

http://bartekdrozdz.com/project/soundviz



Visualization from mic input

http://www.jeshua.me/spectrascade/sc



```
num           5500
spread        0.018
spawn         80
lifetime      4.268
gravF         5.789
gravExp       2
cameraSpee    0.1
vDamp         0.92
pSzBase       23
pTail         1
grav          0
lineW         1.35
musicMult     9
```

LMFAO: I'm Sexy and I Know It (http://www.lmfaomusic.com)

Spectrascade - by Jeshua Bratman and Anna Chen.

function CreativeCoding(everyone) {

console.log(**thank you!**);

}