Engn 395 Project 07*: Good (?) Vibrations*

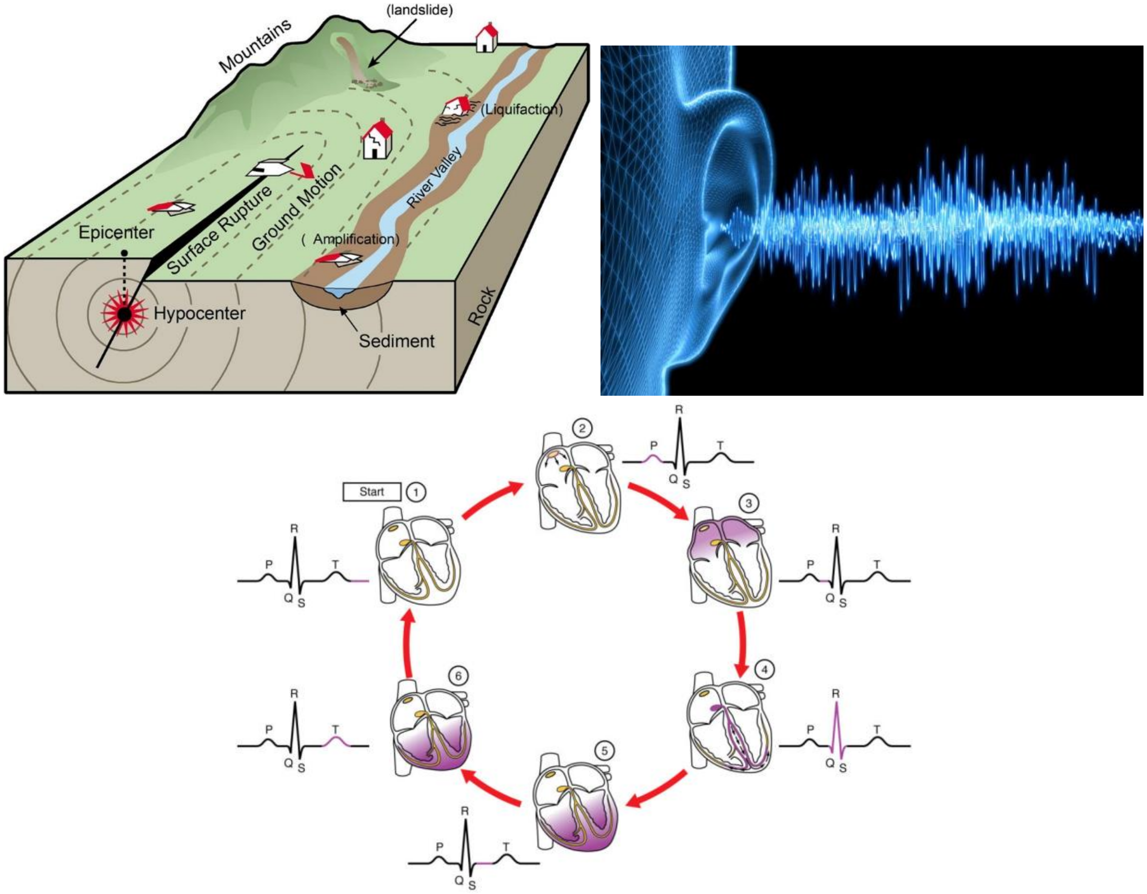

Figure 1. Top left: Surface rupture caused by earthquake. Image credit: https://ascelibrary.org/doi/10.1061/%28ASCE%29NH.1527-6996.0000264. Top right: sound waves arrive at the ear. Image credit: dreamstime.com Bottom: ECG signal generated by various electromechanical phase of the heart. Image credit: https://brilliantnurse.com/nclex-cardiac-electrical-conduction-system/. Enjoy your 3-pack o fun today!

Background/Introduction:

You never know when or where and earthquake is going to strike. Needless to say, earthquakes wreak havoc around the world. One major obstacle to more fully understanding them and building an early warning system is the availability (or lack thereof) of sensors capable of long-term monitoring in the field. For one, power consumption is an issue---how are you going to keep a sensor alive and well for years at a time? Another issue is that transmitting ground acceleration data via wireless networks is very energetically expensive operation.

One way to tackle these related issues is to use data compression. Data compression transforms a signal into a sparser representation. For instance, Alice might have 1000 samples of a sine wave oscillating at 50 Hz, which she knows from computing the FFT. She would like to store and/or transmit these samples to Bob. Only problem: both of these operations are 'expensive' (in terms of power, storage space, etc). Well, instead of transmitting and/or storing 1000 samples, why not just send 2 numbers. Alice could send the frequency of oscillation and the strength thereof to Bob. On the other end, Bob knows how to interpret that pair of numbers and then run an inverse FFT to reconstruct the original sine wave.

A very similar process can be done with discrete wavelet decomposition and reconstruction.

This project explores the basis for such data compression and denoising schemes applied to 3 different application fields: earthquakes; music/speech; and biomedical. Let's get started!

Data sets:

Earthquake:

PEER offers download of a large number of ground motion records. You will need to sign-up to access files: https://ngawest2.berkeley.edu/

Matlab uses the Kobe earthquake data in its examples. You can access/load it in the workspace by doing >> load kobe;

Music/Sound:

You will also want a short music file of your choosing, such as an .mp4 or (preferably) a .wav file. There are several classics available in the box folder for this project.

ECG data:

Use data from the MIT-stress test database, same as in Project 02

Matlab functions you will likely want to use:

- Statistics: `mean, median, std, mad (or robustSTD)`
- Wavelets: `wavedec, waverec, wmaxlev, wdencmp() [discourage use of the latter]`
- Audio: `audioread, audiowrite`
- Other: `find, length`

A. Earthquake Analysis:

Your aim here is to develop a data compression/denoising scheme based on thresholding of discrete wavelet transform coefficients. To this end, identify parameters that are optimal in the sense of:

- Keeping enough, but not too many coefficients. E.g. how exactly will you set your threshold value? You are essentially trading off the compression ratio (how many coefficients you zero out) vs the reconstruction quality.
  Let your mind's eye be the first judge of signal quality.
  Quantify it using this metric: $|xrec - x|/|x|$. Quantify this.
- Which wavelet is best? Try a few different families such as Daubechies ('db'), Symmlet ('sym') , etc. See more here: https://www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html

- What is the overall net result? How good was your data compression scheme? How much data savings did you realize vs. how much degradation in signal quality.

B. Sound/speech Analysis:

Your aim here is similar to that of the Earthquake problem. Instead of ground acceleration records, the ultimate judge of quality is your ear. How good (or bad) is the sound quality when playing back the reconstructed signal? Quantify this. Have a play with various thresholding values and/or schemes, try different wavelet families. What were your ultimate findings?

C. ECG Analysis:

Your aim here is to develop a data compression and/or denoising scheme that accurately reconstructs the QRS complex. No other phase of the cardiac signal is considered important for this task. Basically, the scenario is that you are part of a biomedical team building a long-term heart-rate monitor. Data storage is critical, and you want to transmit the least possible amount of bits. Similar to the earthquake phase of this project, the ultimate judge of quality is your mind's eye. Could you automated algorithm from project 2 still tag the reconstructed QRS complex? Only one way to find out!

How good (or bad) is the quality of the reconstructed ECG just judgement by eye? AS before, have a play with various thresholding values and/or schemes, try different wavelet families. What were your ultimate findings?