Signal Processing Basics – 2

Miroslav Živković
Recap last week
Signal presentation

- Time domain
  - Complex to solve
- Frequency domain

See the Fourier transform handout
Complex Wave Patterns

- Sound waves occupying the same space combine to form a new wave of a different shape.
- Harmonically related waves add together and can create any complex wave pattern.
- Harmonically related waves have frequencies that are multiples of a basic frequency.

Note: All frequency waves do not have to start at zero, they can be “out of phase”. The amount of shift in degrees is called their phase angle.
Generation of digital signal

- Analog signal
- Sampling
- Quantizing
- Encoding
- Quantized signal
- Digital data

Multimedia Retrieval
Sampling

- Analog signal is sampled every $T_s$ seconds (sampling interval)
- $f_s = 1/T_s$ is called the **sampling rate** (sampling frequency).
- Discretization in time-domain
According to the Nyquist theorem, the sampling rate must be at least 2 times the highest frequency contained in the signal.
Quantization and encoding of a sampled signal

Quantization codes

<table>
<thead>
<tr>
<th>Quantization codes</th>
<th>Normalized amplitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Normalized quantized values

<table>
<thead>
<tr>
<th></th>
<th>-1.50</th>
<th>1.50</th>
<th>3.50</th>
<th>3.50</th>
<th>2.50</th>
<th>-1.50</th>
<th>-2.50</th>
<th>-1.50</th>
<th>-1.50</th>
</tr>
</thead>
</table>

Normalized error

<table>
<thead>
<tr>
<th></th>
<th>-0.38</th>
<th>0</th>
<th>+0.26</th>
<th>-0.44</th>
<th>+0.30</th>
<th>-0.40</th>
<th>-0.24</th>
<th>+0.38</th>
<th>-0.30</th>
</tr>
</thead>
</table>

Quantization code

| 2 | 5 | 7 | 7 | 6 | 2 | 1 | 2 | 2 |

Encoded words

| 010 | 101 | 111 | 111 | 110 | 010 | 001 | 010 | 010 |
Homework

<table>
<thead>
<tr>
<th>n</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>x(n)</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x(n-1)</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>x(n-2)</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$x - N_1$, $h - N_2$ samples
How many samples in a result?
Exploiting Superposition and Time-Invariance

• Are there sets of “basic” signals, \( x_k[n] \), such that:
  • We can represent any signal as a linear combination (weighted sum) of these building blocks?
  • The response of an LTI system to these basic signals is easy to compute and provides significant insight.

• DT Systems: 
  (unit pulse)

\[ \delta[n] \]

\[ \delta[n - n_0] \]

• CT Systems: 
  (impulse)

\[ K \delta(t) \]

\[ \delta(t - t_0) \]
Define the unit pulse response, $h[n]$, as the response of a LTI system to a pulse function, $\delta[n]$.

Using the principle of time-invariance:

$$\delta[n] \rightarrow h[n] \quad \Rightarrow \quad \delta[n-k] \rightarrow h[n-k]$$

Using the principle of linearity:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \delta[n-k] \quad \Rightarrow \quad y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] = x[n] * h[n]$$

Comments:

- The summation is referred to as the convolution sum.
- The symbol * is used to denote the convolution operation.
• Any LTI system is completely characterized by its impulse response.

• Convolution has a graphical interpretation:
Visualizing convolution

• Four basic steps

\[ h[k] \xrightarrow{Flip} h[-k] \]
\[ \xrightarrow{Shift} h[n - k] \]
\[ \xrightarrow{Multiply} x[k]h[n - k] \]
\[ \xrightarrow{Sum} \sum_{k=-\infty}^{+\infty} x[k]\ h[n - k] \]
Calculating successive values

• We can calculate each output point by shifting the unit pulse response one sample at a time:

\[ y[n] = \sum_{k=-\infty}^{\infty} x[k] h[n-k] \]

• \( y[n] = 0 \) for \( n < ??? \)
• \( y[-1] = ..., y[0] = ..., y[1] = ..., ..., \)
• \( y[n] = 0 \) for \( n > ??? \)
Graphical convolution

\[
\begin{align*}
&y(-3) = \sum_{k=-\infty}^{\infty} x(k)h(-3-k) = 0 \\
&y(-2) = \sum_{k=-\infty}^{\infty} x(k)h(-2-k) = 0 \\
&y(-1) = \sum_{k=-\infty}^{\infty} x(k)h(-1-k) = 0 \\
&y(0) = (4)(3) = 12
\end{align*}
\]
Graphical convolution

\[ h(k) \]
\[ x(k) \]
\[ h(1-k) \]
\[ h(2-k) \]
\[ h(3-k) \]
\[ h(4-k) \]

\[ k = -6 \quad -5 \quad -4 \quad -3 \quad -2 \quad -1 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \]

\[ y(1) = (4)(2) + (5)(3) = 23 \]
\[ y(2) = (4)(1) + (5)(2) + (3)(3) = 23 \]
\[ y(3) = (4)(0) + (5)(1) + (3)(2) + (1)(3) = 14 \]
\[ y(4) = (3)(1) + (1)(2) = 5 \]
\[ y(5) = (1)(1) = 1 \]

(not shown)
Observations

• Observations:
  – $y[n] = 0$ for $n > 5$

• Define the duration of signal as the difference in time from the first nonzero sample to the last nonzero sample
  – The duration of $h[n]$, $L_h = 3$ samples
  – The duration of $x[n]$, $L_x = 4$

• The duration of $y[n]$ is: $L_y = L_x + L_h - 1 = 6$ (sanity check)

• The output has a duration longer than the input => convolution often acts like a low pass filter and smooths the signal.
Properties of convolution

- **Commutative**
  \[ x[n] * h[n] = h[n] * x[n] \]

- **Distributive**
  \[ x[n] * (h_1[n] + h_2[n]) = (x[n] * h_1[n]) + (x[n] * h_2[n]) \]

- **Associative**
  \[ x[n] * h_1[n] * h_2[n] = (x[n] * h_1[n]) * h_2[n] = (x[n] * h_2[n]) * h_1[n] \]
End-of-homework

- A method for computing the output of a discrete-time (DT) linear time-invariant (LTI) system known as convolution.
- Demonstrated how this operation can be performed analytically and graphically.
- Three important properties: commutative, associative and distributive.
- See also handout (from https://www.ece.umd.edu/class/enee322-1.F2004/Handouts/conv.pdf)
Eye movements and tracking
The eye—“the world’s worst camera”

- suffers from numerous optical imperfections...
- ...endowed with several compensatory mechanisms
The retina is a light sensitive structure inside of the eye responsible for transforming light into signals, which are later converted into an image by the visual cortex in the brain.

The fovea is a section of the retina that contains a high density of both kinds of light receptor cells found in the eye, i.e. Cone and Rod cells.
– Noton and Stark (1971)

- showed that participants tend to fixate identifiable regions of interest, containing “informative details”;
- coined term “scanpath” describing eye movement patterns

An example

Instructions
1: examine at will
2: estimate wealth
3: estimate ages
4: guess previous activity
5: remember clothing
6: remember position
7: time since last visit
The Eye Tracker
The Eye Tracker
The Eye Tracker

• monitors eye-movements from millisecond to millisecond

• provides information about where people look and for how long

• Does it with a certain sampling rate
Applications

- Ergonomics and Human Factors
- Marketing and Advertising
- Websites, Virtual reality
- Displays, HCI
- Psychology, Psychophysics, Neuroscience
Websites

- Most people view websites in a “F” shaped flow.
  - First they scan the page at the top, from left to right.
  - Then the eyes go back to the left and down the page.
  - They again scan to the right and back along the same pattern.
Eye-Tracking While Reading
Some Facts About Reading

- Readers **fixate** a particular word (200 - 300ms) and jump to the next (no “smooth” reading)
- Readers often make a jump (or **saccade**) covers 7-9 letter spaces
- During a saccade visual input is reduced (we are practically blind)
- Readers skip short words and words that are highly predictable (these words are identified in the parafoveal region)
- Readers regress (look back)
- Readers often undershoot on return sweeps (going from the end of a line to the next line)
Since Jay always jogs a mile seems like a short distance to him.

- Green circle = fixation after progressive saccade (first-pass)
- Red circle = fixation after regressive saccade
- Blue circle = fixation after progressive saccade (second-pass)
• If readers experience some sort of trouble they may fixate the difficult region longer and the may even regress to earlier parts of the sentence/text.
Different Measures

- **First fixation duration**: duration of first fixation in a region
- **First-pass duration**: time spent in a region before moving on or looking back
- **Regression path duration**: time from first entering a region until moving the eyes beyond that region, includes regression time
- **Second-pass duration**: duration of re-fixations
- **Total duration**: the sum of all fixations in a region
- **Probability of a regression**: the percentage of regressive eye-movements out of a region
Different Measures

Bart annoyed *Homer* because…

Reading Times for word 3 (*Homer*)

- First fixation duration = 3
- First-pass duration = 3 + 4
- Regression Path duration = 3 + 4 + 5
- Second-pass duration = 6
- Total duration = 3 + 4 + 6
The linking problem

eye

\[ \leftrightarrow \]

mind
Eye Mind Assumption

• Cooper (1974), Tanenhaus et al. (1995)

• The Mind-Eye hypothesis
  • Relationship between eye fixations and the meaning of concurrently spoken sentence
  • Using this relationship as a research tool in cognitive psychology and psycholinguistics
  • Applications:
    – Speech perception and memory
    – Language processing


Eye-Movement Parameters

• Using eye movements in reading as a means to infer cognitive processes (e.g., language processing)
Eye movements

- Eye movements are mainly used to reposition the fovea
- Five main classes of eye movements:
  - Saccadic (saccades and fixations)
  - Smooth pursuit
  - Vergence
  - Vestibular
  - Physiological nystagmus (“micro saccades”)
- Other types of movements are e.g. adaptation, accommodation
Saccades

- Rapid eye movements between fixations used to reposition fovea
- Involuntary, abrupt, rapid, small movements or jerks of both eyes simultaneously in changing the point of fixation
- Range in duration from 10 ms – 100 ms
- Effectively blind during transition
- Deemed *ballistic* (pre-programmed) and *stereotyped* (reproducible)
Saccades

- Amplitude: 400 - 600°/sec
- Skipping words: 2-3-letter words are skipped 75%, 8-letter words are never skipped;
- Regressions: 10-letter spaces happen because of problems in understanding text
Fixations

• Probably the most important type of eye movement for attentional applications

• Relatively stable eye-in-head position:
  – 90% viewing time is devoted to fixations
  – duration: 100 ms – 600 ms
  – Spatial dispersion: < 2°
  – Threshold velocity: < 15 – 100° /msec
Blinks

- Rapid bilateral eyelid closure and co-occurring eye movement
- Blink types and initiators (Smit, 2008)

<table>
<thead>
<tr>
<th>Type</th>
<th>Elicited through</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reflex</td>
<td>Air puff, tactile / electrical stimulation, Glabellar tapping, electrical stimulation</td>
</tr>
<tr>
<td>Trigeminal</td>
<td>Air puff, tactile / electrical stimulation</td>
</tr>
<tr>
<td>Non-trigeminal</td>
<td>Loud sounds</td>
</tr>
<tr>
<td></td>
<td>Touch</td>
</tr>
<tr>
<td></td>
<td>Light flash</td>
</tr>
<tr>
<td></td>
<td>Menacing visual input</td>
</tr>
<tr>
<td>Spontaneous</td>
<td>Generator</td>
</tr>
<tr>
<td>Voluntary</td>
<td>Conscious thought</td>
</tr>
</tbody>
</table>
Salvucci & Goldberg, 2000

• Identifying Fixations and Saccades in Eye-Tracking Protocols

• Event detection algorithm can have a significant impact on the results of higher-level analyses

• Badly designed detection algorithm can detect too little or a lot of fixations

• The authors propose taxonomy to facilitate the comparison of different algorithms
<table>
<thead>
<tr>
<th>Criteria</th>
<th>Spatial</th>
<th>Temporal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Velocity-based</td>
<td>Duration sensitive</td>
</tr>
<tr>
<td></td>
<td>Dispersion-based</td>
<td>Locally adaptive</td>
</tr>
<tr>
<td></td>
<td>Area-based</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Criteria</td>
<td>I-VT I-HMM I-DT I-MST I-AOI</td>
<td></td>
</tr>
<tr>
<td>Spatial</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Temporal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Taxonomy

• Velocity-based algorithms emphasize the velocity information in the eye-tracking protocols, taking advantage of the fact that fixation points have low velocities and saccade points have high velocities.

• Assuming a constant sampling rate, velocities are simply distances between sampled points and thus we can ignore the temporal component implicit in velocities.
Taxonomy

- Dispersion-based algorithms emphasize the dispersion (i.e. spread distance) of fixation points, under the assumption that fixation points generally occur near one another.

- Area-based algorithms identify points within given areas of interest (AOIs) that represent relevant visual targets.
Taxonomy

- Temporal characteristics
  - Duration information
  - Locally adaptive

- Duration information motivated by the fact that fixations are rarely less than 100 ms and often in the range of 200-400 ms.

- Local adaptivity allows the interpretation of a given data point to be influenced by the interpretation of temporally adjacent points; this is useful, for instance, to compensate for differences between 'steady-eyed' individuals and those who show large and frequent eye movements.
Velocity-threshold Identification (I-VT)

- Based on point-to-point velocities-
  - Threshold: 20°/sec, Sen & Megaw, 1984
  - When only point-to-point velocities are known, infer the velocity threshold based on sampling frequency along with some exploratory data analysis
  - Fixations (< 100°/sec)
  - Saccades (> 300°/sec)

- Pseudo Code Algorithm (protocol, velocity threshold)
  - Calculate point-to-point velocities for each point in the protocol
  - Label each point below velocity threshold as a fixation point, otherwise as a saccade point
  - Collapse consecutive fixation points into fixation groups, removing saccade points
  - Map each fixation group to a fixation at the centroid of its points
  - Return fixations in format <x,y,t,d>
Dispersion-based algorithm (I-DT)

- Based on groups of consecutive points within a maximum separation
  - Moving window initially spans minimum consecutive data points based on duration threshold and sampling frequency
  - Checks dispersion of points in window by summing differences between the points’ max and min x and y values
  - \[ D = [\max(x) - \min(x)] + [\max(y) - \min(y)] \]. Note that alternative dispersion metrics could be based upon spatial variance or area of samples.
  - When dispersion is above the dispersion threshold, the window does not represent a fixation, and the window moves one point to the right
  - If the dispersion is below the dispersion threshold, the window represents a fixation. Expand the window to the right until the dispersion is above threshold.
- Output is \(<x,y, t, d>\)
- Requires two parameters, the dispersion threshold and the duration threshold
Dispersion-based algorithm (I-DT)

I-DT (protocol, dispersion threshold, duration threshold)

While there are still points

  Initialize window over first points to cover the duration threshold

  If dispersion of window points <= threshold
     Add additional points to the window until dispersion > threshold
     Note a fixation at the centroid of the window points.
     Remove window points from points
  Else
     Remove first point from points

Return fixations
R package: \texttt{emov (github)}

- An R package for fixation and saccade detection in eye tracking recordings
- This package implements a dispersion-based algorithm (I-DT)
- Detects fixations in the first place, compared to the velocity threshold algorithms which detect saccades.
• Eye-position $X_t = \{x_t, y_t\}$
• Distances to $D(X_{t-1})$ and $D(X_{t+1})$ in both horizontal and vertical directions (two features)
• 2-point angular velocities between $X_t$ and $D(X_{t-1})$ (absolute difference/sampling period)
• 2-point angular velocities between $X_t$ and $D(X_{t+1})$
• Angular acceleration, high acceleration => small saccades
Sampling frequency and eye-tracking measures

- Anderson, Nystrom, Holmqvist: *Sampling frequency and eye-tracking measures: how speed affects durations, latencies and more*
Initial observations

• RQ: what sampling frequency and/or data amount is necessary to be certain of eye-tracking results where the sampling-related uncertainty exceeds the effect magnitudes found

• Eye tracking devices could have sampling frequencies from 50 Hz (even less!) to 2KHz
  – Enright (1998) provides evidence that saccadic peak velocity can be well estimated in 60 Hz data from eye-trackers, but only for saccades larger than 10°.
  – For saccades shorter that 10°, typical of reading, the peak velocity calculation is not accurate with 60 Hz data.
As it is impossible to have an infinite sampling frequency, each eye-tracker instead takes an instantaneous snapshot of the eye at a fixed rate (typically 25-2000 Hz). Each snapshot is a point in time, taken to be representative of a whole interval of time. For instance, with a 50 Hz system, the position of the eye at each sample is assumed to be valid for the whole 20 ms, even if it is very likely that the eye did not have that exact position just before the moment of sampling. The eye-tracker cannot sample the eye in a position which it has not moved to yet, but the system may sample the eye in the correct position or a position it recently had. By necessity, sampling always lags behind the position of the eye because the eye is constantly moving to some extent. Figure illustrates the resulting temporal sampling error, where the eye-tracker mis-estimates the correct point in time that a particular event (the triggering of a velocity criterion) takes place.
Temporal sampling error

- One point temporal sampling error
- A visual search task: how fast participants can locate a target object in a cluttered scene ("saccadic latency")
- Duration from the onset of a stimulus until a saccade is launched towards a designated target on the screen if the time-stamp of the
- Example: stimulus onset is 5674 ms
- The saccade to the target is detected to be launched at 6743 ms
- Resulting saccadic latency is 6743 - 5674 = 1069 ms
- This assumes that we correctly detected the saccade launch at exactly 6743 ms
- Sampling frequency =?
Temporal sampling error

- Sampling frequency $f_s$ => sampling interval = ?
  
  $\epsilon = t_{\text{Measured}} - t_{\text{True}}, \epsilon \sim U(0, \frac{1}{f_s})$

- $t_{\text{TRUE}}$ is time instance when actual eye-movement happened
- $t_{\text{MEASURED}}$ is time instance where actual sampling took place
- On average, the sampling error will amount to half a sample worth of time.
- With a 50 Hz system, how much is this?
• Given the sampling frequency, we can solve for the minimum number of data points required to contain the temporal sampling error within 1 ms of the expected mean of the error.
  – \( N = c(f_s)^{-2} \)

• \( N \) – number of data points
• \( c \) – constant (1208500)
• \( f_s \) – sampling frequency
The aim of this simulation is to explore how much data we need to reduce the variance of the temporal sampling error for one-point measures to a level where it is easy to estimate and compensate for. The goal is that 95% of all temporal sampling errors should be within the same 1 ms span.

Simulation 1 Pseudo code for Simulation 1.
for $f_s = 10$ to 2000 in steps of 10 (sampling frequency) do
  $i = 1$ (data amount, i.e. number of one-point measures)
  $E(\varepsilon) = \frac{1}{f_s}/2$ (expected temporal sampling error)
  while 1 do
    Generate 10000 one-point error vectors $\varepsilon = \{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_i\}$ from eq. (1) of length $i$
    if 95% of the $\varepsilon$ have means within $[E(\varepsilon) - 0.5, E(\varepsilon) + 0.5]$ ms then
      store $i$ and $f_s$
      Break while-loop
    else
      $i = i + 1$
    end if
  end while
end for
Experiment – impact of sampling frequency to event detection

• To investigate the effect sampling frequency has on event detection, data collected at 1250 Hz was used as a baseline, and then compared to the same data resampled to 250 and 50 Hz respectively.

<table>
<thead>
<tr>
<th>Sampling frequency (Hz)</th>
<th>1250</th>
<th>250</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixations (number)</td>
<td>19656</td>
<td>19793</td>
<td>19955</td>
</tr>
<tr>
<td>Fixation duration, M±SD, (ms)</td>
<td>177±83</td>
<td>177±83</td>
<td>175±85</td>
</tr>
<tr>
<td>Saccades (number)</td>
<td>24684</td>
<td>24768</td>
<td>24835</td>
</tr>
<tr>
<td>Saccade duration, M±SD, (ms)</td>
<td>43±19</td>
<td>43±19</td>
<td>48±22</td>
</tr>
</tbody>
</table>
Examples of Eye Movements while Reading

• A gaze replay, recorded at 300Hz using the Tobii TX300 eye tracker, of a participant in a reading study: https://www.youtube.com/watch?v=VBTZNydhUh0w

• This may be something to show in your project
In this task, you start with “raw” data, and identify the events from it. A raw data contains (per sample) information about the horizontal/vertical pupil position, horizontal/vertical pupil diameter, etc.

You may need to perform data cleaning, and/or filtering of data.

An event is either a fixation, a saccade or a blink. A good starting point describing (some of) algorithms may be the following article, and references therein:


In this task, you will use the events data that you determined from the measurements (raw data).

You will have to:

• Clean the data (remove missing values)
• Explore it – there are readily available packages in R for this.
• Make a (Machine Learning) model for prediction of interest/comprehensibility of users.

This requires to include feedback data from users (interest/comprehensibility)
Based on analysis performed in previous point

Take a single events data file, and, once you read the data, predict the interest and comprehensibility of the user. These will be compared to the actual user feedback(s). You will reason about discrepancies in results, and identify the issues for it.

(Optional) You will identify the document that will give the highest values for interest and comprehensibility for a given user.

Once you do it, you will verify this by reading the events data and make predictions, which will be compared with actual feedback for a given document.