Signal and Image Processing by Computer – Spring 2016

Final Project

Project List

Clarifications:

- The project descriptions given here are for general definition of the project goals, and particularly emphasize the required extensions to the material taught in class.

- In your presentations and reports you are required to show a full understanding of the material from class and its extensions.

- You are encouraged to include in your projects discussions and demonstrations further to those explicitly specified here. In turn, this may positively affect the evaluation of your project in the originality and independent-work aspects.
Project #1: Bit-Allocation for One-Dimensional Signal Digitization

Explore and extensively demonstrate the problem of digitizing one-dimensional signals defined on a continuous-domain via the bit-allocation procedure.

The above should be thoroughly mathematically analyzed and demonstrated in Matlab for the following one-dimensional signals (a signal \( \varphi(t) \) is defined over \( t \in [0,1] \)):

1. **Sine**: \( \varphi(t) = A \cdot \sin(2\pi \omega t + \phi) \)
   where
   \( A > 0 \) is the amplitude
   \( \omega > 0 \) is the frequency (consider integer and non-integer values)
   \( \phi \in [0,2\pi] \) is the phase

2. **Cosine**: \( \varphi(t) = A \cdot \cos(2\pi \omega t + \phi) \)
   The parameters are defined as for the sine function.

3. **Sinusoidal Linear Chirp**: \( \varphi(t) = A \cdot \sin(2\pi t(\omega_0 + \alpha t) + \phi) \)
   where
   \( A > 0 \)
   \( \omega_0 > 0 \) is the initial frequency
   \( \alpha > 0 \) is the rate of linear growth of the frequency
   \( \phi \in [0,2\pi] \) is the phase

4. **Sinusoidal Exponential Chirp**: \( \varphi(t) = A \cdot \sin(2\pi t(\omega_0 + \alpha t - 1) + \phi) \)
   where
   \( A > 0 \)
   \( \omega_0 > 0 \) is the initial frequency
   \( \alpha > 1 \) is the rate of exponential growth of the frequency
   \( \phi \in [0,2\pi] \) is the phase

5. **Sinusoidal Linear Chirp with an Increasing Amplitude**: \( \varphi(t) = (A_0 + \beta t) \cdot \sin(2\pi t(\omega_0 + \alpha t) + \phi) \)
   where
   \( A_0 > 0 \) is the initial amplitude
   \( \beta > 0 \) is the rate of linear growth of the amplitude
   \( \omega_0 > 0 \) is the initial frequency
   \( \alpha > 0 \) is the rate of linear growth of the frequency
   \( \phi \in [0,2\pi] \) is the phase

Additional guidelines:

- The demonstrations should exhibit the bit-allocation results for various bit-budgets and signal parameters.
- The digitized signals will be presented by graph plots and played as audio signals.
• In the mathematical analysis you should formulate relevant properties of the signals (e.g.,
energy of derivatives, maximal/minimal value, etc.) and look for particular cases (i.e., specific
parameter forms or values) that clearly exhibit interesting findings.
• The initial continuous-time signal should be modeled in your Matlab experiments by taking a
very high resolution digital version of the \(\varphi(t)\) function.

Example: the function \(\varphi(t) = \cos(2\pi t)\) \((t \in [0,1])\) can be approximated in Matlab using the
high resolution grid, \(t_{\text{grid}}=0:0.00001:1\), and a vector of high precision signal values (e.g., use
double type variables), \(\phi_t=\cos(2\pi t_{\text{grid}})\).

**Project #2: Bit-Allocation for Two-Dimensional Signal Digitization**

Explore and extensively demonstrate the problem of digitizing two-dimensional signals defined on a
continuous-domain via the bit-allocation procedure.

The above should be thoroughly mathematically analyzed and demonstrated in Matlab for the following
two-dimensional signals (a signal \(\varphi(x, y)\) is defined over \((x, y) \in [0,1] \times [0,1]):\

1. **Sine:**
   a. Separable: \(\varphi(x, y) = A \cdot \sin(2\pi \omega_x x) \sin(2\pi \omega_y y)\)
   b. Non-separable: \(\varphi(x, y) = A \cdot \sin\left(2\pi (\omega_x x + \omega_y y)\right)\)
      where, \(A > 0\) is the amplitude,
      \(\omega_x > 0\) is the horizontal frequency (consider integer and non-integer values)
      \(\omega_y > 0\) is the vertical frequency (consider integer and non-integer values)

2. **Cosine:**
   a. Separable: \(\varphi(x, y) = A \cdot \cos(2\pi \omega_x x) \cos(2\pi \omega_y y)\)
   b. Non-separable: \(\varphi(x, y) = A \cdot \cos\left(2\pi (\omega_x x + \omega_y y)\right)\)
      The parameters are defined as for the sine function.

3. **Sinusoidal Linear Chirp:**
   a. Separable: \(\varphi(x, y) = A \cdot \sin\left(2\pi x(\omega_{0x} + \alpha_x x)\right) \sin\left(2\pi y(\omega_{0y} + \alpha_y y)\right)\)
   b. Non-separable: \(\varphi(x, y) = A \cdot \sin\left(2\pi x(\omega_{0x} + \alpha_x x) + 2\pi y(\omega_{0y} + \alpha_y y)\right)\)
      where, \(A > 0\),
      \(\omega_{0x} > 0\) and \(\omega_{0y} > 0\) are the initial horizontal and vertical frequencies, respectively.
      \(\alpha_x > 0\) and \(\alpha_y > 0\) are the rates of linear growth of the horizontal and vertical
      frequencies, respectively.
Additional guidelines:

- The demonstrations should exhibit the bit-allocation results for various bit-budgets and signal parameters.
- The digitized signals will be presented by as images.
- In the mathematical analysis you should formulate relevant properties of the signals (e.g., energy of derivatives, maximal/minimal value, etc.) and look for particular cases (i.e., specific parameter forms or values) that clearly exhibit interesting findings.
- The initial continuous-time signal should be modeled in your Matlab experiments by taking a very high resolution digital version of the \( \varphi(x, y) \) function.
  
  Example: the function \( \varphi(x, y) = \cos(2\pi(x + y)) \) \(((x, y) \in [0,1] \times [0,1])\) can be approximated in Matlab using the high resolution grid:

  ```matlab
  grid=0:0.00001:1;
  [x_grid,y_grid]=meshgrid(grid,grid);
  and a matrix of high precision signal values (e.g., use double type variables):
  phi_xy=cos(2*pi*(x_grid,y_grid));
  ```

**Project #3: Bit-Allocation for Coding Digital (One-Dimensional) Signals**

Bit-allocation is useful for two different purposes:

1. Digitization of an analog signal (i.e., defined over a continuous interval and takes values from a continuous range), providing a sampled and quantized form of the signal under a bit-budget constraint.
2. Coding of a digital signal (i.e., one that was already sampled and quantized to have \( N_0 \) samples that each of them is represented using \( b_0 \) bits, providing a representation cost of \( B_0 = N_0 b_0 \)).

Here, the bit-allocation purpose is to represent the signal using a reduced bit-cost of \( B < B_0 \) bits. This is achieved by optimizing the new number of samples and quantizer-resolution \( N \) and \( b \), respectively, under a bit-budget constraint of \( B \) bits.

Note that here the initial signal is a vector, thus, the continuous sampling analysis that is based on integrals reduces to a discrete one that relies on sums.

In this project you will compare the two procedures above by mathematically and experimentally studying them in a way that emphasizes the similarities and differences.

More specifically, for each of the analog signals, \( \varphi(t) \ t \in [0,1] \), that will be defined below you will apply:
1. Bit allocation for digitization of $\varphi(t)$ under a bit-budget constraint of $B_0$ bits. This results in the optimal vector $\varphi_0$, having $N_0$ samples that each requires $b_0$ bits (and $N_0 b_0 \leq B_0$ holds).

2. Bit allocation for coding $\varphi_0$ to a reduced representation bit-cost $B_0 < B$. This results in the optimal vector , having $N$ samples that each requires $b$ bits (and $Nb \leq B$ holds).

The above should be studied and demonstrated for:

1. **Sine**: $\varphi(t) = A \cdot \sin(2\pi \omega t + \phi)$
   where
   - $A > 0$ is the amplitude
   - $\omega > 0$ is the frequency (consider integer and non-integer values)
   - $\phi \in [0,2\pi]$ is the phase

2. **Cosine**: $\varphi(t) = A \cdot \cos(2\pi \omega t + \phi)$
   The parameters are defined as for the sine function.

3. **Sinusoidal Linear Chirp**: $\varphi(t) = A \cdot \sin(2\pi t (\omega_0 + \alpha t) + \phi)$
   where
   - $A > 0$
   - $\omega_0 > 0$ is the initial frequency
   - $\alpha > 0$ is the rate of linear growth of the frequency
   - $\phi \in [0,2\pi]$ is the phase

Additional guidelines:

- The demonstrations should exhibit the bit-allocation results for various bit-budgets ($B_0, B$ and their pairing) and signal parameters.
- The digitized signals will be presented by graph plots and played as audio signals.
- In the mathematical analysis you should formulate relevant properties of the signals (e.g., energy of derivatives, maximal/minimal value, etc.) and look for particular cases (i.e., specific parameter forms or values) that clearly exhibit interesting findings.
- The initial continuous-time signal should be modeled in your Matlab experiments by taking a very high resolution digital version of the $\varphi(t)$ function.

Example: the function $\varphi(t) = \cos(2\pi t)$ ($t \in [0,1]$) can be approximated in Matlab using the high resolution grid, $t_{\text{grid}}=0:0.00001:1$, and a vector of high precision signal values (e.g., use double type variables), $\text{phi}_t=\cos(2\pi t_{\text{grid}})$. 
Project #4: Bit-Allocation for Coding Digital Two-Dimensional Signals

This project is defined as an extension of Project #3 to two-dimensional signals.

The signals that should be examined here are:

A signal $\varphi(x,y)$ is defined over $(x,y) \in [0,1] \times [0,1]$.

1. **Sine:**
   a. Separable: $\varphi(x,y) = A \cdot \sin(2\pi \omega_x x) \sin(2\pi \omega_y y)$
   b. Non-separable: $\varphi(x,y) = A \cdot \sin(2\pi(\omega_x x + \omega_y y))$
   where, $A > 0$ is the amplitude,
   $\omega_x > 0$ is the horizontal frequency (consider integer and non-integer values)
   $\omega_y > 0$ is the vertical frequency (consider integer and non-integer values)

2. **Sinusoidal Linear Chirp:**
   a. Separable: $\varphi(x,y) = A \cdot \sin(2\pi x (\omega_{0,x} + \alpha_x x)) \sin(2\pi y (\omega_{0,y} + \alpha_y y))$
   b. Non-separable: $\varphi(x,y) = A \cdot \cos(2\pi(\omega_x x + \omega_y y))$
   where, $A > 0$,
   $\omega_{0,x} > 0$ and $\omega_{0,y} > 0$ are the initial horizontal and vertical frequencies, respectively.
   $\alpha_x > 0$ and $\alpha_y > 0$ are the rates of linear growth of the horizontal and vertical frequencies, respectively.

Project #5: Bit-Allocation based on Improved Sampling

In class, we considered bit-allocation in the standard basis by uniform sampling that is coupled with a constant-value reconstruction within each sampling interval, i.e., the reconstructed (continuous time) signal is formed as

$$\hat{\varphi}(t) = \varphi_i \text{ for } i \in \left[\frac{i-1}{N}, \frac{i}{N}\right], i = 1, ..., N.$$ 

In this project you will explore bit-allocation that relies on an improved reconstruction having the linear form of:

$$\hat{\varphi}(t) = a_i + c_i (t - t_i) \text{ for } i \in \left[\frac{i-1}{N}, \frac{i}{N}\right], i = 1, ..., N.$$ 

where $t_i$ is the center of the $i^{th}$ interval.

1. Mathematically analyze the optimal sampling for the above reconstruction, i.e., for a given subinterval of the signal (arbitrary $i$ and $N$ are given) what are optimal sampling coefficients $a_i$ and $c_i$ that minimize the MSE over the interval.
2. Define and study the bit-allocation problem that is based on linear-representation and scalar quantization of the representation coefficients.
The coefficient quantization is defined as follows: all the \( \{a_i\}_{i=1}^N \) coefficients are quantized using a scalar quantizer of \( b_a \) bits, and all the \( \{c_i\}_{i=1}^N \) coefficients are quantized using a scalar quantizer of \( b_c \) bits.

The optimal bit-allocation procedure provides the optimal number of equal-length intervals \( N \), and the optimal quantizer resolutions \( b_a, b_c \) that minimize the reconstruction MSE under a bit-budget constraint of \( N(b_a + b_c) \leq B \).

In this project you may consult parts of the paper:


Study the above described procedure for various bit-budgets and signal parameters, and compare its performance to the standard bit-allocation that uses constant-value reconstruction rule.

The above should be thoroughly mathematically analyzed and demonstrated in Matlab for the following one-dimensional signals (a signal \( \varphi(t) \) is defined over \( t \in [0,1] \)):

1. **Sine**: \( \varphi(t) = A \cdot \sin(2\pi \omega t + \phi) \)
   
   where
   
   \( A > 0 \) is the amplitude
   
   \( \omega > 0 \) is the frequency (consider integer and non-integer values)
   
   \( \phi \in [0,2\pi] \) is the phase

2. **Sinusoidal Linear Chirp**: \( \varphi(t) = A \cdot \sin(2\pi t (\omega_0 + \alpha t) + \phi) \)

   where

   \( A > 0 \)
   
   \( \omega_0 > 0 \) is the initial frequency
   
   \( \alpha > 0 \) is the rate of linear growth of the frequency
   
   \( \phi \in [0,2\pi] \) is the phase

Additional guidelines:

- The demonstrations should exhibit the bit-allocation results for various bit-budgets and signal parameters.
- The digitized signals will be presented by graph plots and played as audio signals.
- In the mathematical analysis you should formulate relevant properties of the signals (e.g., energy of derivatives, maximal/minimal value, etc.) and look for particular cases (i.e., specific parameter forms or values) that clearly exhibit interesting findings.
- The initial continuous-time signal should be modeled in your Matlab experiments by taking a very high resolution digital version of the \( \varphi(t) \) function.

Example: the function \( \varphi(t) = \cos(2\pi t) \) (\( t \in [0,1] \)) can be approximated in Matlab using the high resolution grid, \( t_{\text{grid}}=0:0.00001:1 \), and a vector of high precision signal values (e.g., use double type variables), \( \text{phi}_t = \cos(2\pi t_{\text{grid}}) \).
Project #6: Bit-Allocation via Adaptive Tree-Structured Signal-Partitioning

Following the material taught in class, consider representation of a signal $\phi(t)$ over the interval $[l_1, l_2)$ given $b$ bits and using only a single sample. This representation introduce an error formulated as:

$$
\varepsilon^2(l_1, l_2, b) = \frac{1}{l_2 - l_1} \int_{l_1}^{l_2} \phi^2(t) \, dt - \left( \phi_{\text{opt}}^{l_1, l_2} \right)^2 + \frac{1}{12} \cdot \frac{(\phi_H - \phi_L)^2}{2^{2b}}
$$

where $\phi_{\text{opt}}^{l_1, l_2} \triangleq \frac{1}{l_2 - l_1} \int_{l_1}^{l_2} \phi(t) \, dt$ is the optimal sampling coefficient, and $\phi_H$, $\phi_L$ are the maximal and minimal values of the signal (over the entire $[0,1]$ range).

Alternatively, we can use an accurate measurement of the quantization error that updates the formula for sampling+quantization error into:

$$
\varepsilon^2(l_1, l_2, b) = \frac{1}{l_2 - l_1} \int_{l_1}^{l_2} \phi^2(t) \, dt - \left( \phi_{\text{opt}}^{l_1, l_2} \right)^2 + \left( \phi_{\text{opt}}^{l_1, l_2} - Q_b(\phi_{\text{opt}}^{l_1, l_2}) \right)^2
$$

where $Q_b(\phi_{\text{opt}}^{l_1, l_2})$ is the quantized value of the sample $\phi_{\text{opt}}^{l_1, l_2}$.

In this project you will study an adaptive coding of a given signal $\phi(t)$ ($t \in [0,1]$) via unequal partition to sub-intervals that can be represented using a binary tree structure.

The bit allocation for a given interval $[l_1, l_2]$ using $b$ bits is determined by evaluating the benefit of splitting the interval into two halves and coding each of them using $\frac{b}{2} - 1$ bits (as two bits are required for representing this split in a binary tree structure).

The above decision is recursive and can be mathematically formulated as:

Split the $[l_1, l_2]$ interval if

$$
\varepsilon^2(l_1, l_2, b) > \frac{1}{2} \left[ \varepsilon^2(l_1, l_1 + \frac{l_2 - l_1}{2}, \frac{b}{2} - 1) + \varepsilon^2(l_1 + \frac{l_2 - l_1}{2}, l_2, \frac{b}{2} - 1) \right]
$$

The bit-allocation procedure for a given signal starts from the interval $[0,1]$ and the given bit budget $B$.

In this project you will study this bit-allocation method for one-dimensional and its extension to two-dimensional signals (including natural images) using quadtreess.

An example for the quadtree structure (for a 2D signal):
this can be represented using the tree

having the binary representation:  1-0011-0001 0011

Additional notes:

1. Compare the performance of the method using each of the formulations given above for $\varepsilon^2(l_1, l_2, b)$.

2. The above method may be improved by limiting the number of bits for quantizing a sample. More specifically, determine that a sample can be quantized with at max $b_{\text{max}}$ bits, then the error for an interval becomes $\varepsilon^2(l_1, l_2, \min\{b, b_{\text{max}}\})$ in the cost of $\min\{b, b_{\text{max}}\}$ bits.

3. The recursive decision rule can be applied in two ways:
   a. Top-down: where the rule is used to determine interval splits.
   b. Bottom-up: starting from a full tree of some (high) depth and recursively using the rule to determine the merging of intervals corresponding to neighboring leaves having the same parent.

Evaluate the two approaches. Note that the bottom-up should perform much better.
**Project #7: Wiener Filtering for Signal Denoising**

Study the topic of Wiener filtering for denoising a signal from additive white Gaussian noise.

Present mathematical analysis and Matlab demonstrations.

Show the Wiener filter solution to this problem for synthetic images where the signal autocorrelation is exactly known (i.e., construct a random family of images and mathematically calculate its accurate autocorrelation).

In addition, design a nonlinear algorithm that will perform better than the Wiener filter.

A family of the synthetic images may be defined as follows:

Square $B \times B$ images formed according to the following four random variables:

- Two graylevels $L_1, L_2$
- The position of the line, $X$
- The binary variable $D$ that defines the line as vertical or horizontal.

Note that $B$ cannot be too large due to computational reasons.

![Image of synthetic images](image)

Start by implementing the above for the corresponding one-dimensional problem.
Project #8: Wiener Filtering for Denoising of Polygonal-Shaped Images

In this project you will denoise full-size corrupted images based on a statistical model of a class of small image patches.

The clean images and patches contain only two gray levels \( L_1, L_2 \) (these are deterministic values).

1. The class of random image patches
   A patch of size \( B \times B \) (it is suggested to start from trying \( B = 7 \)) is defined by a line that intersects the patch into two parts of graylevels \( L_1, L_2 \). The line is defined by its radius, \( \rho \), from the central pixel of the patch and an angle \( \theta \):

   ![Diagram](image)

   here the patch can be formulated as
   \[
   \varphi(x, y) = \begin{cases} 
   L_1, & \text{if } (x - x_0) \cdot \cos \theta + (y - y_0) \cdot \sin \theta - \rho < 0 \\
   L_2, & \text{otherwise}
   \end{cases}
   \]
   where \((x_0, y_0)\) is the center of the patch (note that the latter formulation is accurate for a continuous domain, whereas you will use it for a discrete grid).
   In addition, \( \rho \) and \( \theta \) are independent random variables, taking values from a discrete and finite set of options, i.e.,
   \[
   P_\theta(\theta_i) = w_i \quad \text{for } i = 1, \ldots, M_\theta.
   \]
   and
   \[
   P_\rho(\rho_i) = q_i \quad \text{for } i = 1, \ldots, M_\rho.
   \]
   Accordingly, this class contains \( M_\theta M_\rho \) options for the patches, each is achieved by a pair of \( \theta \) and \( \rho \) values.
   Mathematically analyze the statistical properties of this class, for various probability functions (i.e., values of \( \{w_i\}_{i=1}^{M_\theta} \) and \( \{q_i\}_{i=1}^{M_\rho} \)), and demonstrate it in Matlab.

   Formulate the second order statistics (mean, which is not necessarily zero here, and covariance) of this class and use it to construct the Wiener filter for denoising a given corrupted patch from additive white Gaussian noise.

2. Denoise a full image of arbitrary \( H \times W \) size.
   Consider a two-graylevel (the above \( L_1, L_2 \)) full-sized image containing large polygonal shapes.
   The denoising here will be applied in two different ways:
   1. Denoise all the non-overlapping patches of the image and averaging the results.
   2. Denoise all the overlapping patches of the image and averaging the results.
Project #9: PCA and SVD

It is known that the SVD achieves optimal energy compaction (in the MSE sense) for a given signal, whereas the PCA obtains the optimal energy compaction (in the expected MSE sense) for a random family of signals that is defined according to its second-order statistics.

Study the above statements and compare the SVD and the PCA both mathematically and experimentally.

Note that you are required to suggest additional mathematical and experimental ways to deeply study this topic.

Example for an image:

- Repeat the experiments for various values of:
  - The graylevels $L_1, L_2$.
  - Noise variance.
  - Probability values of $\{w_i\}_{i=1}^{M_\theta}$ and $\{q_i\}_{i=1}^{M_\rho}$.
Project #10: Denoising via K-term Approximation in Walsh-Hadamard and Haar Domains

In this project you will study the problem of denoising a given noisy image by applying K-term approximation (also known as thresholding) in a unitary transform domain.

The examined corruption model is of additive white Gaussian noise.

The unitary transforms that will be examined and compared are Walsh-Hadamard and Haar.

You should mathematically analyze the above problem and denoising method.

In addition, the provided Matlab demonstrations should include:

1. Various one-dimensional and two-dimensional signals that are defined mathematically (see examples in this project list), as well as natural images.
2. Various noise levels (i.e., noise variance values).
3. Number of coefficients kept in the K-term approximation.

The performance of the above should be analyzed in terms of average denoising (estimation) MSE as function of the noise variance and/or K-terms.

Project #11: Denoising via K-term Approximation in DFT and DCT Domains

This project is defined as Project #10 but for the DFT and DCT transforms instead of the Walsh-Hadamard and Haar.