

ECE 366 Honors Project Report

Two-dimensional Signal Processing and Touch-tone Signal Processing

Chenxin Zhang
PID: 181333637

Fall 2025

Abstract

A comprehensive examination of two-dimensional signal processing methods for image analysis and the processing of Dual-Tone Multi-Frequency (DTMF) signals used in touch-tone telecommunication systems is presented. The study encompasses two-dimensional convolution, the roles of magnitude and phase in the 2D Fourier transform, and approaches for removing structured noise in the frequency domain. In addition, a complete DTMF decoding framework is developed using MATLAB, and its performance is assessed under various noise conditions to evaluate robustness and reliability.

Contents

1	Introduction	3
2	Two-dimensional Signal Processing	3
2.1	Two-dimensional Convolution: Theory (Task 1.1.1)	3
2.1.1	Continuous-time 2D convolution	3
2.1.2	Discrete-time 2D convolution	3
2.1.3	Relationship between 1D and 2D Convolution	3
2.2	Convolution Masks and Filtering in Images (Task 1.1.2)	4
2.3	Image Loading and Basic Visualization (Task 1.1.3)	5
2.4	Blurring with Box, Gaussian, and Laplacian Kernels (Task 1.1.4)	6
2.5	Importance of Phase in 2D Fourier Transform (Tasks 1.2.x)	8
2.5.1	2D Fourier Transform and Spectra (Tasks 1.2.1–1.2.2)	8
2.5.2	Zero-Phase Reconstruction (Task 1.2.3)	9
2.5.3	Unit Magnitude Reconstruction (Task 1.2.4)	10
2.5.4	Phase Swapping Between Images (Task 1.2.5)	11
2.6	Denoising in the Frequency Domain (Tasks 1.3.1, 1.3.2, 1.3.3)	12
2.6.1	Noisy Image and its Spectrum (Tasks 1.3.1–1.3.2)	12
2.6.2	Frequency-domain Filtering Procedure (Task 1.3.3)	13
3	Processing of Touch-tone (DTMF) Signals	14
3.1	DTMF System Overview	14
3.2	Signal Generation and Spectral Analysis (Tasks 2.1–2.2)	14
3.3	Phone Number Construction and Noisy Signals (Tasks 2.3–2.4)	14
3.4	General DTMF Decoding Procedure (Task 2.5)	15
3.5	DTMF Decoder Implementation and Performance Evaluation (Task 2.6)	16
3.6	Decoding Provided Touch-tone Signals (Task 2.7)	16
4	Discussion and Conclusion	17

1 Introduction

This project is doing two major areas: two-dimensional signal processing for images and the analysis of Dual-Tone Multi-Frequency (DTMF) signals used in touch-tone communication systems. The material is divided into two primary parts.

Part 1 addresses two-dimensional convolution, the role of phase in image representation, and denoising techniques implemented in the frequency domain. Part 2 investigates the synthesis, spectral characterization, and decoding of DTMF signals.

The remainder of this report is organized as follows. Section 2 presents the principles of two-dimensional convolution, convolution masks, phase-based image analysis, and Fourier-transform-based denoising. Section 3 describes the processing and decoding of touch-tone signals. Conclusions and potential extensions are summarized in.

2 Two-dimensional Signal Processing

2.1 Two-dimensional Convolution: Theory (Task 1.1.1)

The definition of two-dimensional convolution in both the continuous-time and discrete-time domains is examined, with its relationship to the one-dimensional convolution operation highlighted.

2.1.1 Continuous-time 2D convolution

The continuous-time two-dimensional convolution of signals $x(t_1, t_2)$ and $h(t_1, t_2)$ is defined as

$$(y * h)(t_1, t_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x(\tau_1, \tau_2) h(t_1 - \tau_1, t_2 - \tau_2) d\tau_1 d\tau_2. \quad (1)$$

2.1.2 Discrete-time 2D convolution

For discrete signals $x[n, m]$ and $h[n, m]$, the 2D convolution is

$$y[n, m] = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x[k, l] h[n - k, m - l]. \quad (2)$$

2.1.3 Relationship between 1D and 2D Convolution

One-dimensional convolution operates on signals defined along a single axis. Given signals $x[n]$ and $h[n]$, the 1D convolution is

$$y[n] = \sum_k x[k] h[n - k]. \quad (3)$$

It forms each output sample by combining a local neighborhood around n .

Two-dimensional convolution extends this idea to signals defined over two axes, such as images. For a 2D input $x[n, m]$ and kernel $h[i, j]$, the convolution is

$$y[n, m] = \sum_i \sum_j x[n - i, m - j] h[i, j]. \quad (4)$$

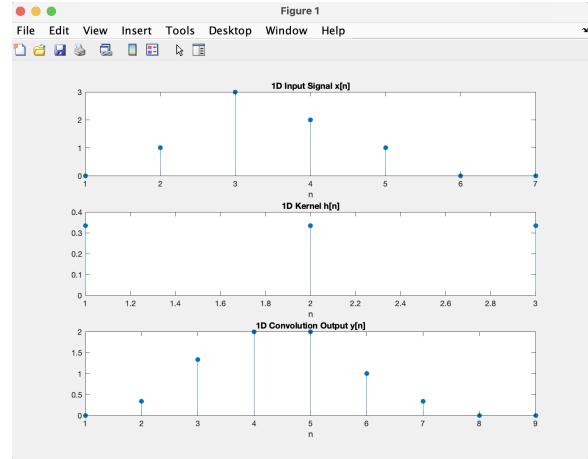
The following is the Demo to illustrate the differences between 1D and 2D Convolution.

```

x = [0 1 3 2 1 0 0]; % Random x[n]
h = [1 1 1] / 3; % simple averaging filter
y = conv(x, h); %Calculate the Conv
figure;% Plot the Figure
subplot(3,1,1);
stem(x, 'filled');
title('1D Input Signal x[n]');
xlabel('n');
subplot(3,1,2);
stem(h, 'filled');
title('1D Kernel h[n]');
xlabel('n');
subplot(3,1,3);
stem(y, 'filled');
title('1D Convolution Output y[n]');
xlabel('n');

```

(a) MATLAB Code demonstrating one-dimensional convolution.



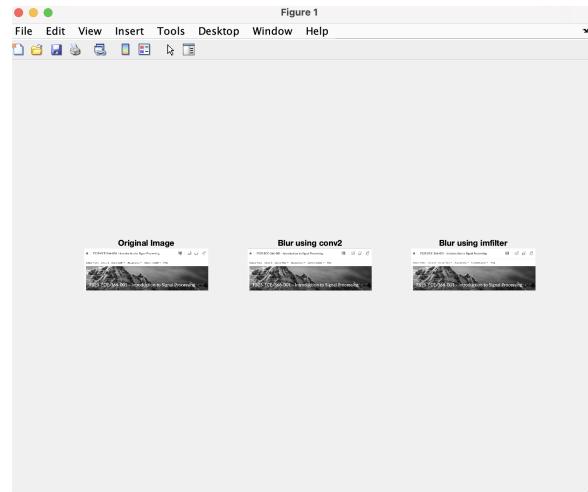
(b) Output of One-Dimensional convolution demonstration.

```

%% 2D Convolution Demo
I = im2double(imread('/Users/chenxinzhang/Desktop/2D figure.png'));
if size(I, 3) == 3
    I = rgb2gray(I);
end
h2 = ones(3,3) / 9;
I_conv2 = conv2(I, h2, 'same');
I_filt = imfilter(I, h2, 'replicate');
figure;
subplot(1,3,1); imshow(I); title('Original Image');
subplot(1,3,2); imshow(I_conv2); title('Blur using conv2');
subplot(1,3,3); imshow(I_filt); title('Blur using imfilter');

```

(c) MATLAB code demonstrating Two-Dimensional convolution.



(d) Output of Two-Dimensional convolution demonstration.

Figure 1: Comparison of One-Dimensional and Two-Dimensional convolution demos.

2.2 Convolution Masks and Filtering in Images (Task 1.1.2)

In image processing, a convolution mask (also called a kernel or filter) is a small matrix that defines a local operation applied to each pixel in an image. During convolution, this mask is placed over a neighborhood of the image centered at a given pixel, and the output value is obtained by computing a weighted sum of the neighboring pixel intensities. The weights are specified by the entries of the mask, and therefore the mask determines how the local region contributes to the filtered output.

Let $w[i, j]$ denote an $M \times N$ convolution mask. When the mask is centered at pixel (n, m) , the filtered image is computed as

$$y[n, m] = \sum_{i=-(M-1)/2}^{(M-1)/2} \sum_{j=-(N-1)/2}^{(N-1)/2} w[i, j] x[n - i, m - j]. \quad (5)$$

Different choices of masks produce different filtering effects. A box (or average) filter assigns equal weights to all pixels in the local neighborhood and produces uniform blurring. A Gaussian filter uses weights that follow a Gaussian distribution, yielding smoother and more natural blurring. A Laplacian filter contains both positive and negative coefficients and is used to emphasize edges by highlighting regions of rapid intensity change.

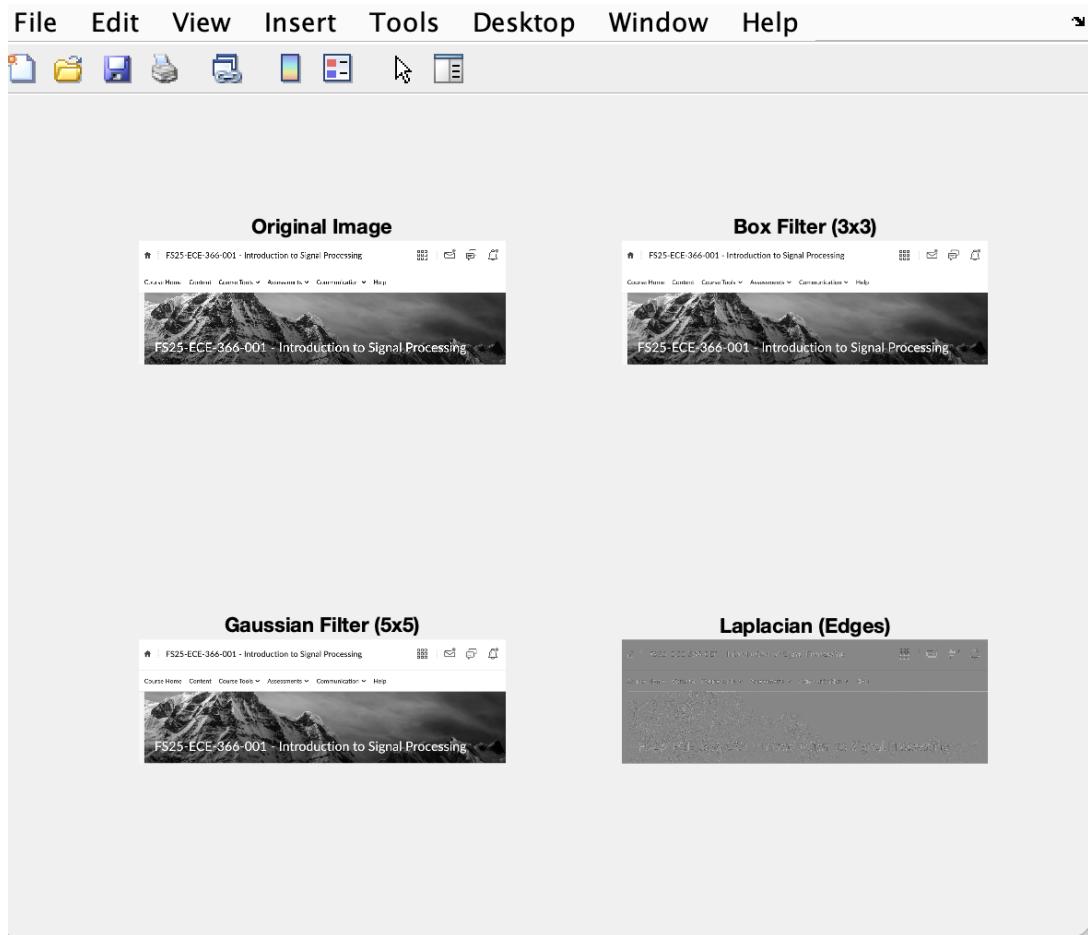


Figure 2: Comparison of box, Gaussian, and Laplacian filtering applied to the same image.

2.3 Image Loading and Basic Visualization (Task 1.1.3)

The images `lena512.mat`, `square.mat`, and `cameraman.mat` are loaded into MATLAB using the `load` command, and each is displayed using the `imagesc` and `colormap(gray)` functions. Figure 3 shows the three original images used in the two-dimensional convolution tasks.

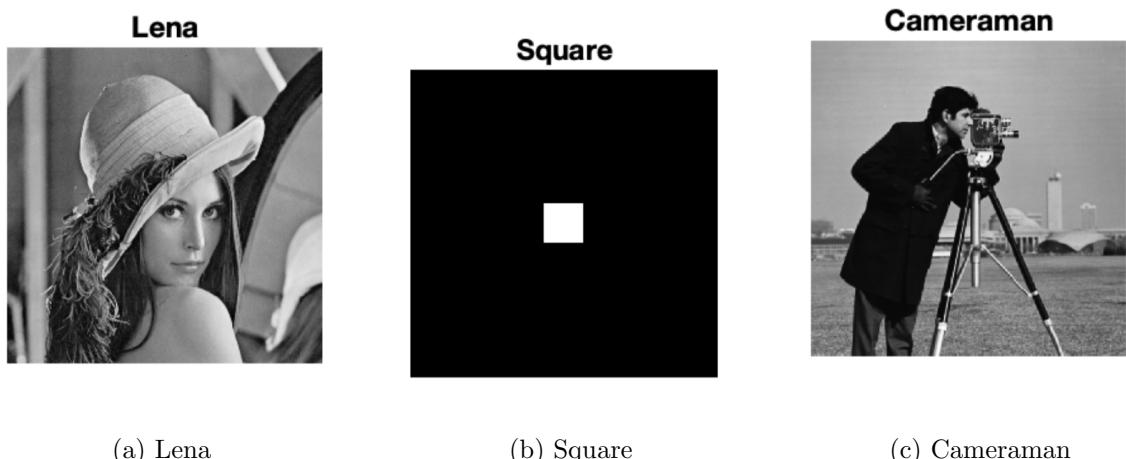


Figure 3: Original images loaded from `lena512.mat`, `square.mat`, and `cameraman.mat`.

2.4 Blurring with Box, Gaussian, and Laplacian Kernels (Task 1.1.4)

The images from Task 1.1.3 were blurred using box, Gaussian, and Laplacian kernels of sizes 3×3 , 5×5 , and 9×9 . The box filter produces uniform smoothing by assigning equal weights to all neighboring pixels. The Gaussian filter yields smoother and more natural blurring due to its spatially varying weights. The Laplacian filter emphasizes edges rather than smoothing the image. The results for the three images are shown in Figures 4, 5, and 6.

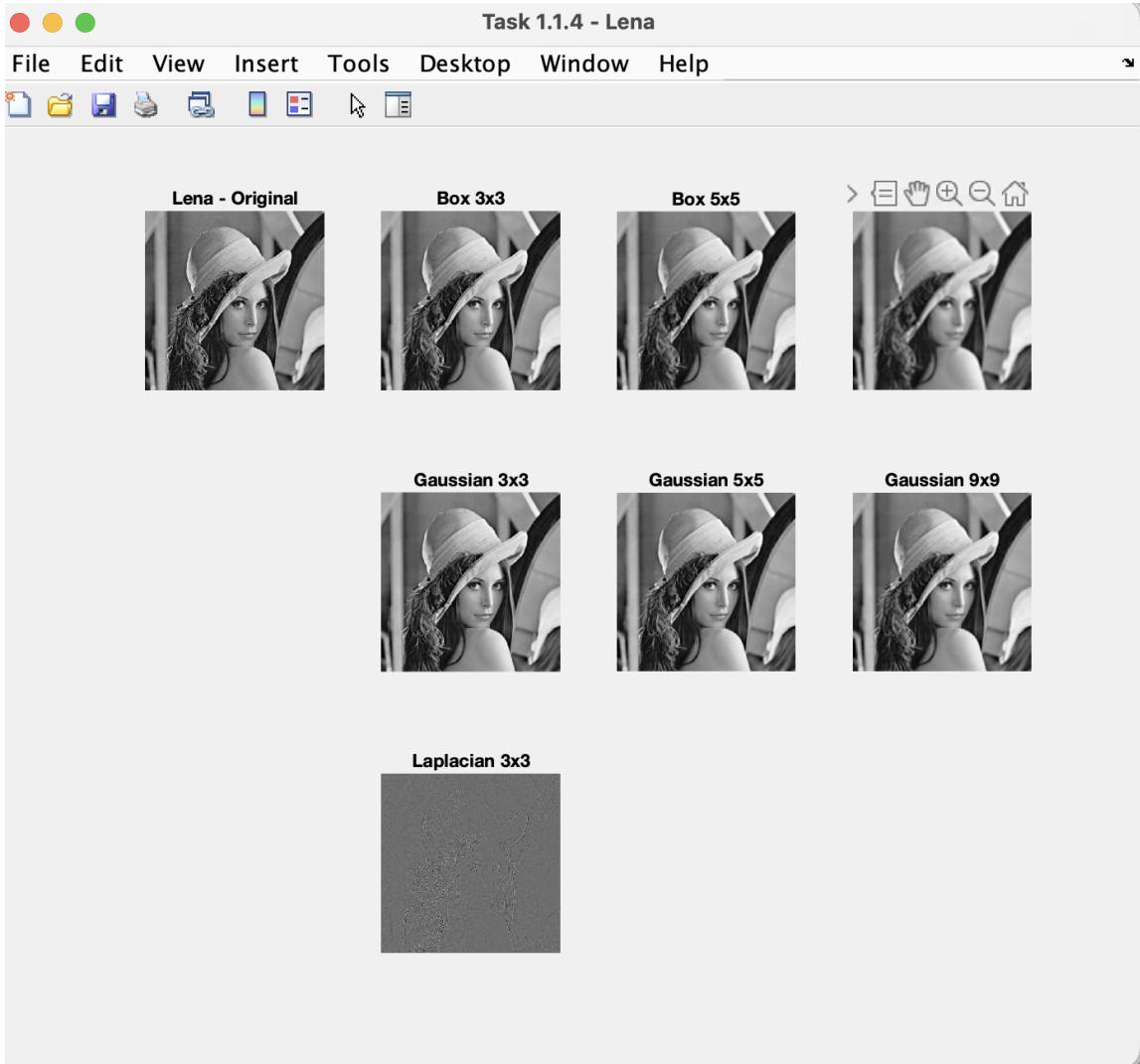


Figure 4: Blurring results using different convolution masks for the Lena image.

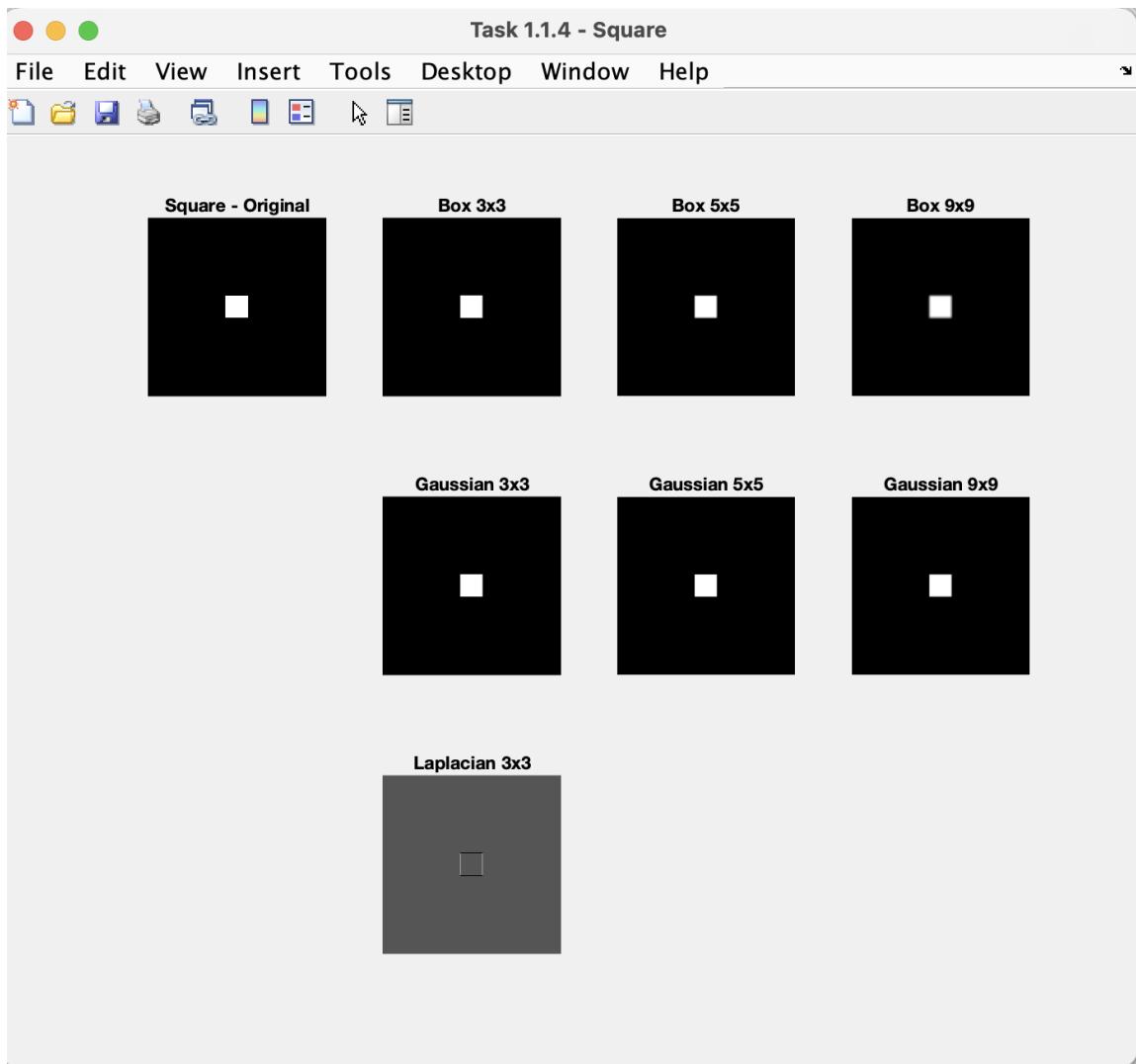


Figure 5: Blurring results using different convolution masks for the Square image.

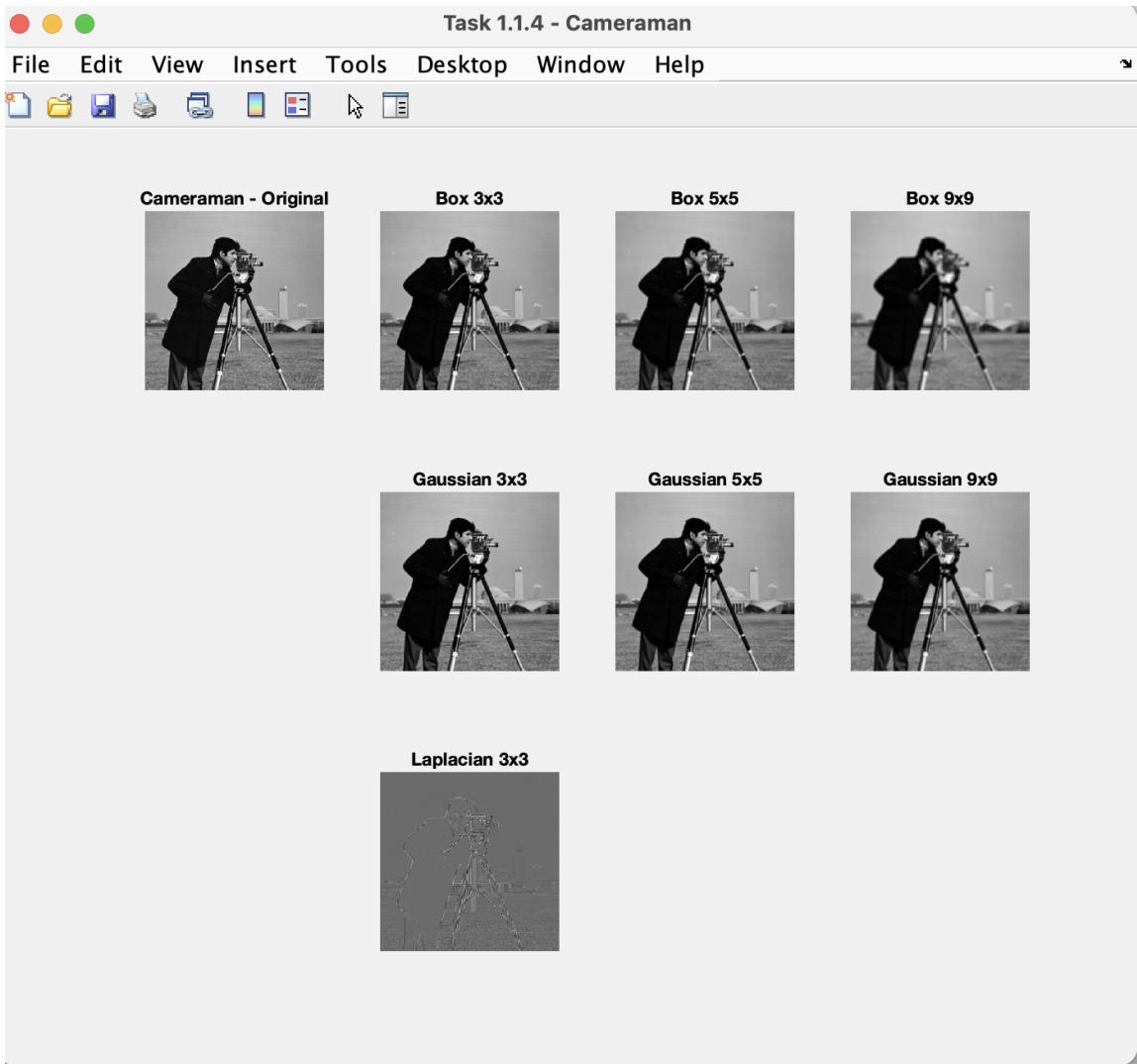


Figure 6: Blurring results using different convolution masks for the Cameraman image.

The box filter produces uniform smoothing but tends to blur edges strongly. The Gaussian filter yields smoother, more natural blurring. The Laplacian filter emphasizes edges and can also be used for sharpening when combined appropriately with the original image. Larger masks generally lead to heavier blurring and loss of detail.

2.5 Importance of Phase in 2D Fourier Transform (Tasks 1.2.x)

2.5.1 2D Fourier Transform and Spectra (Tasks 1.2.1–1.2.2)

The two-dimensional Fourier transform is computed using `fft2`, and the magnitude and phase spectra are obtained using `abs` and `angle`. The magnitude spectrum reflects the distribution of spatial frequencies, while the phase spectrum encodes structural information such as edges and spatial layout. Figure 7 illustrates the magnitude and phase spectra for the Lena and Cameraman images.

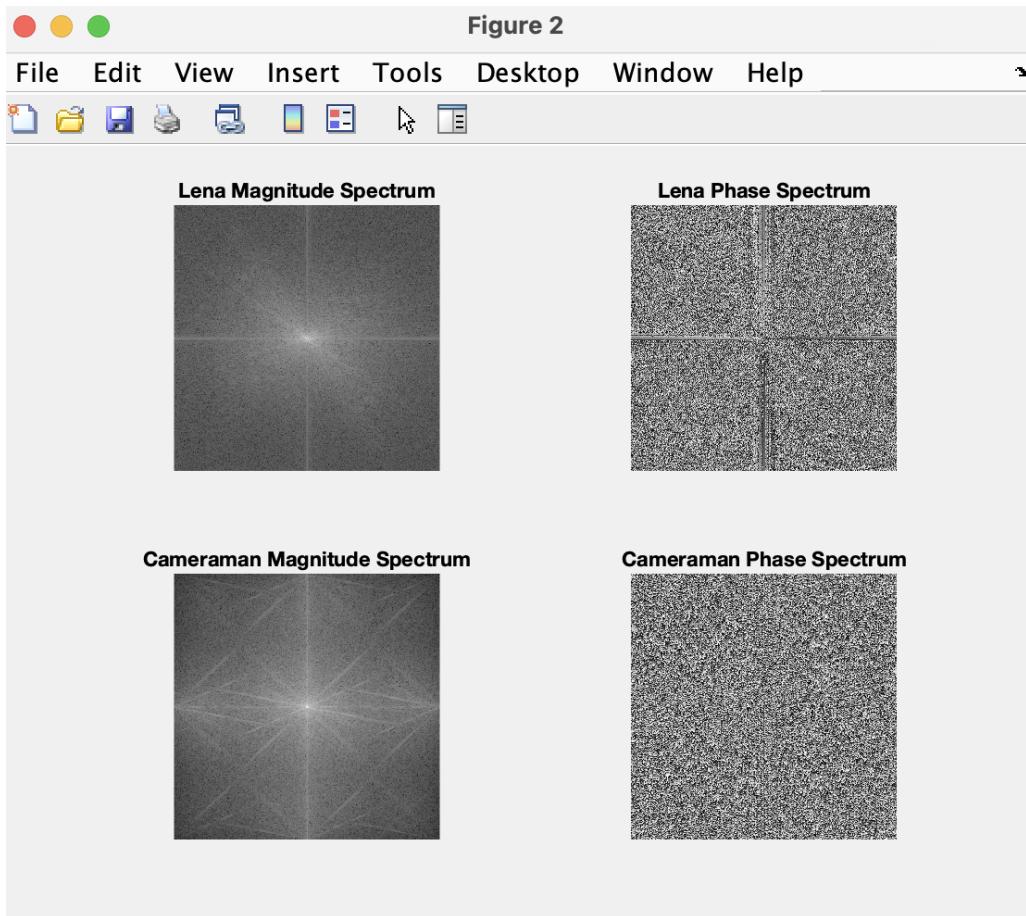


Figure 7: Magnitude and phase spectra of the Lena and Cameraman images.

2.5.2 Zero-Phase Reconstruction (Task 1.2.3)

A modified spectrum is formed by setting all phase values to zero, so that

$$X_1(\omega, \lambda) = |X(\omega, \lambda)|.$$

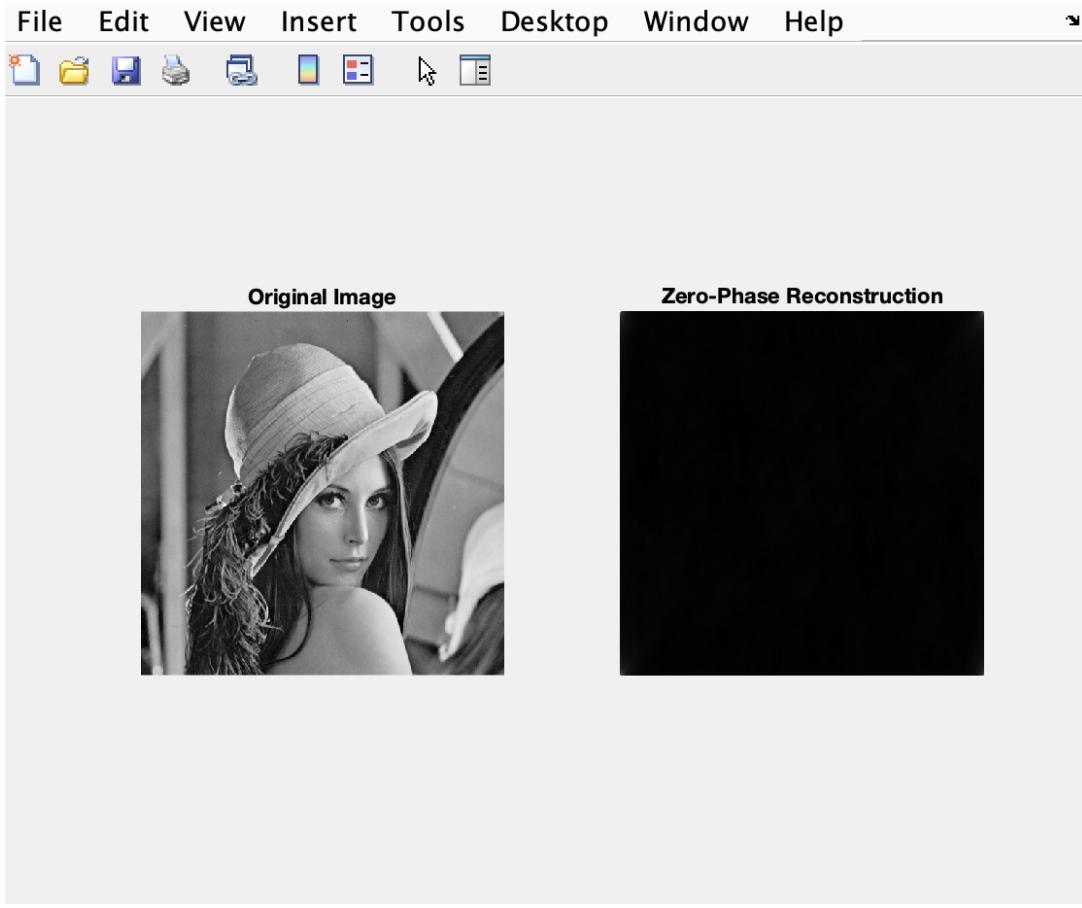


Figure 8: Zero-phase reconstruction of the Lena image.

Applying the inverse two dimensional Fourier transform to this spectrum yields a reconstructed image that contains almost no recognizable structural information. Only broad intensity variations remain, demonstrating that the phase component of the Fourier transform encodes the essential spatial structure of the image, while the magnitude primarily reflects global energy distribution.

2.5.3 Unit Magnitude Reconstruction (Task 1.2.4)

A modified spectrum is formed by preserving only the phase of the Fourier transform and forcing the magnitude to be identically one,

$$X_2(\omega, \lambda) = e^{j\angle X(\omega, \lambda)}.$$

Applying the inverse two-dimensional Fourier transform to this spectrum produces a reconstruction that retains most of the recognizable structural content of the image, but with highly distorted contrast and intensity. This behavior indicates that the phase component conveys the essential spatial structure, whereas the magnitude primarily controls the overall energy and brightness distribution.

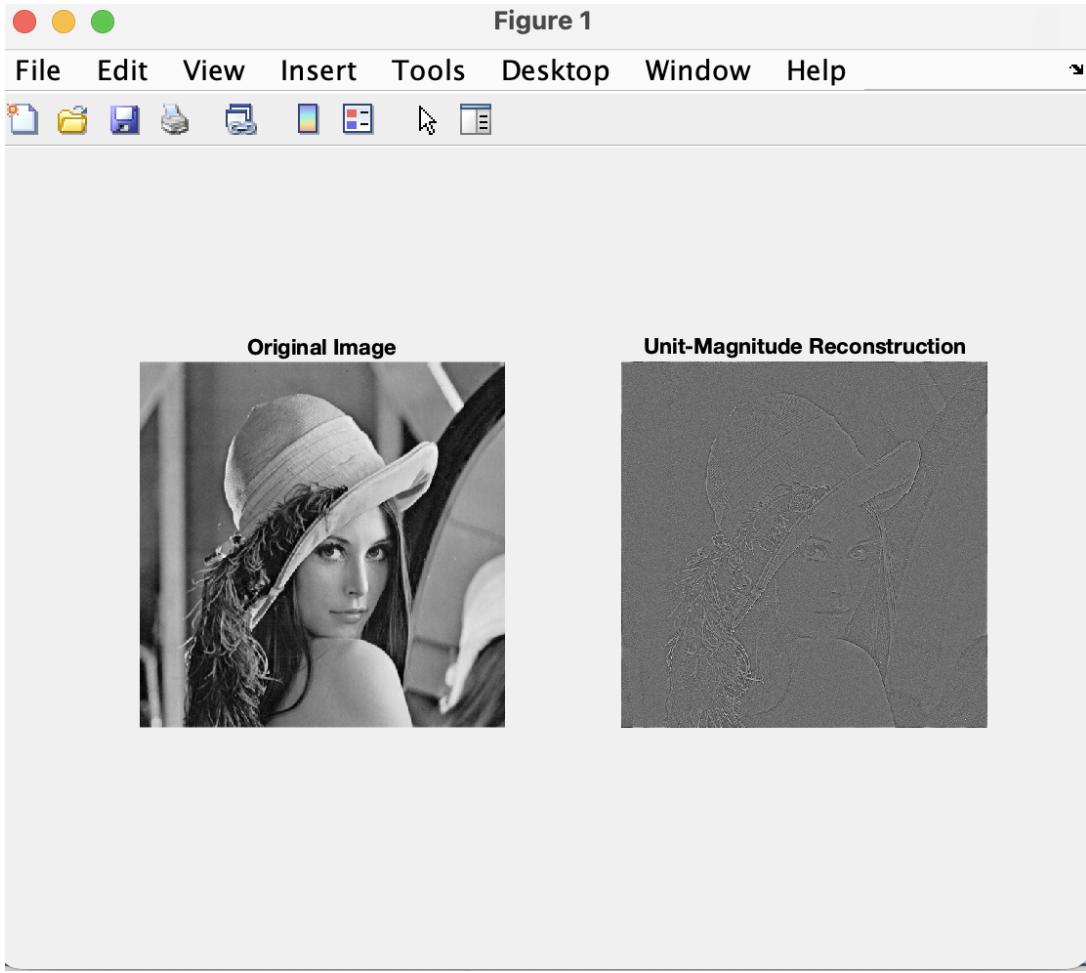


Figure 9: Reconstruction using unit magnitude and original phase (Task 1.2.4).

2.5.4 Phase Swapping Between Images (Task 1.2.5)

Phase information can be further examined by exchanging it between two images. Let $X_L(\omega, \lambda)$ and $X_C(\omega, \lambda)$ denote the Fourier transforms of the Lena and Cameraman images, respectively. Hybrid spectra are constructed by combining the magnitude of one image with the phase of the other:

$$X_{LC}(\omega, \lambda) = |X_L(\omega, \lambda)| e^{j\angle X_C(\omega, \lambda)}, \quad X_{CL}(\omega, \lambda) = |X_C(\omega, \lambda)| e^{j\angle X_L(\omega, \lambda)}.$$

Inverse Fourier transforms of these hybrid spectra yield reconstructions whose visual appearance closely follows the phase donor image, whereas the magnitude primarily influences contrast and overall brightness. This behavior demonstrates that phase plays the dominant role in determining the structural content of an image.

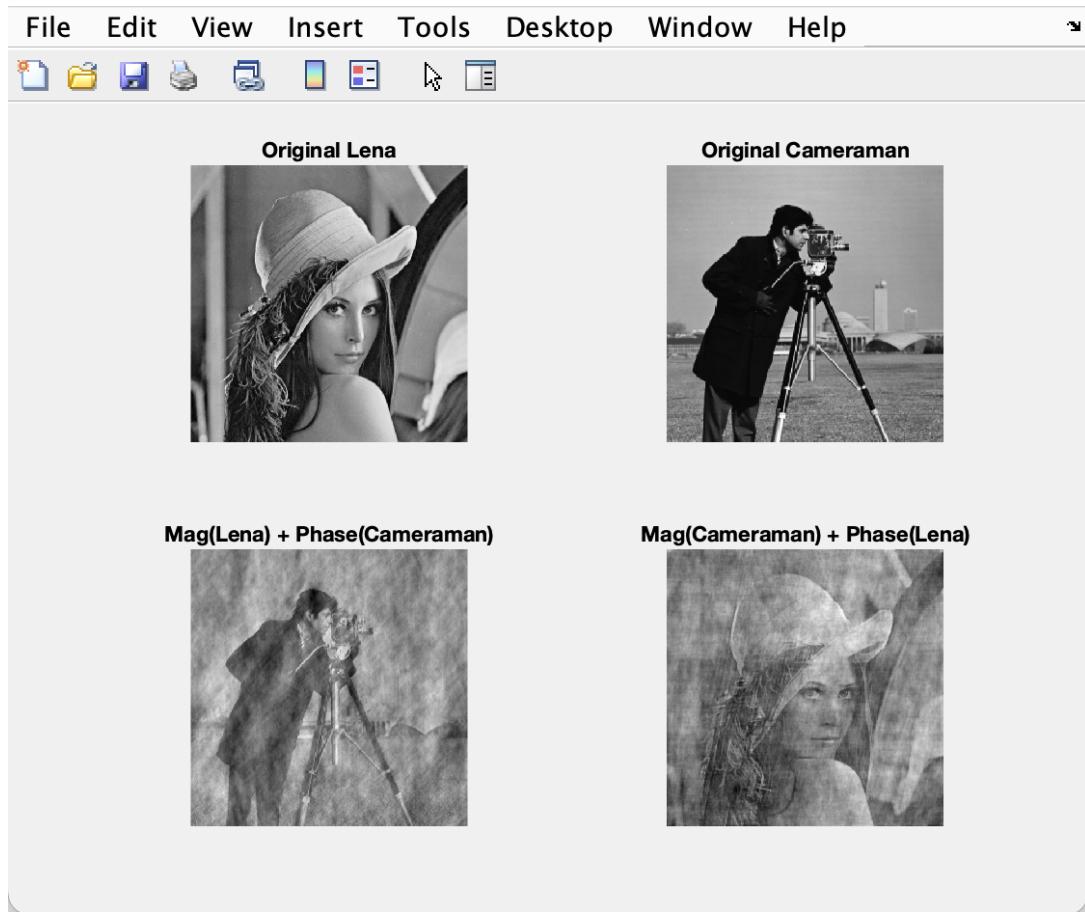


Figure 10: Phase-swapping results between Lena and Cameraman (Task 1.2.5).

2.6 Denoising in the Frequency Domain (Tasks 1.3.1,1.3.2,1.3.3)

2.6.1 Noisy Image and its Spectrum (Tasks 1.3.1–1.3.2)

The noisy image is shown in Figure 12. Its magnitude spectrum reveals distinct peaks corresponding to the additive cosine noise components.

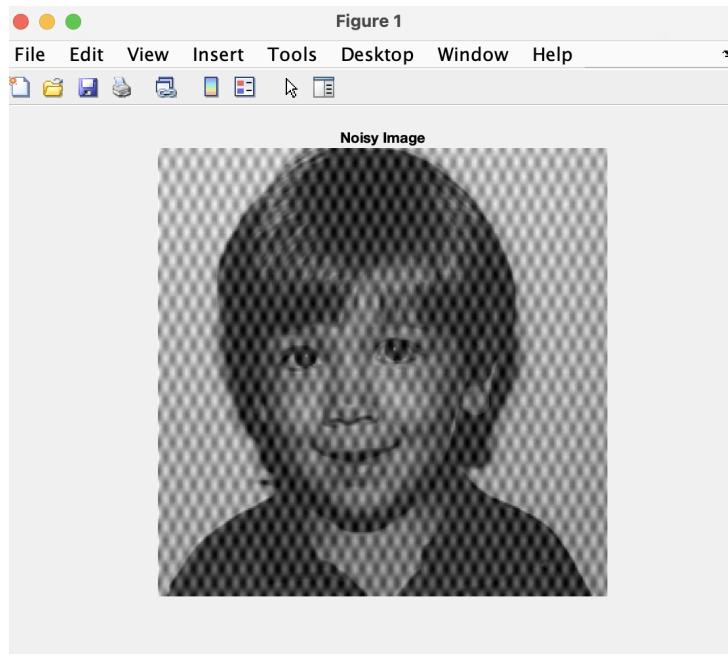


Figure 11: Noisy image provided in `noisy_image.mat`.

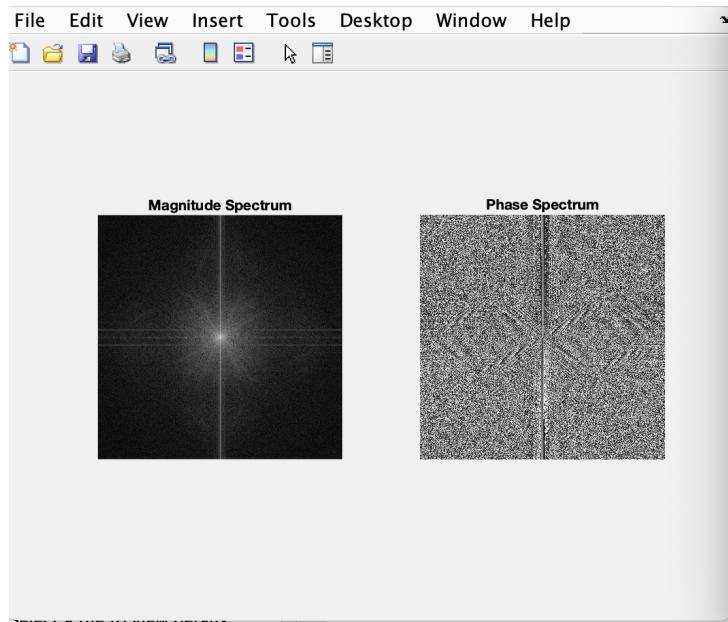


Figure 12: Magnitude and Phase `noisy_image.mat`.

2.6.2 Frequency-domain Filtering Procedure (Task 1.3.3)

A frequency-domain band-stop filter is constructed to attenuate the range of spatial frequencies responsible for the periodic noise present in the image. By suppressing this specific frequency band and retaining both the low-frequency image structure and the remaining high-frequency detail, the undesired periodic patterns are significantly reduced. After filtering in the frequency domain and applying the inverse two-dimensional Fourier transform, the resulting image exhibits substantially diminished noise while preserving the primary visual features. Figure 13.

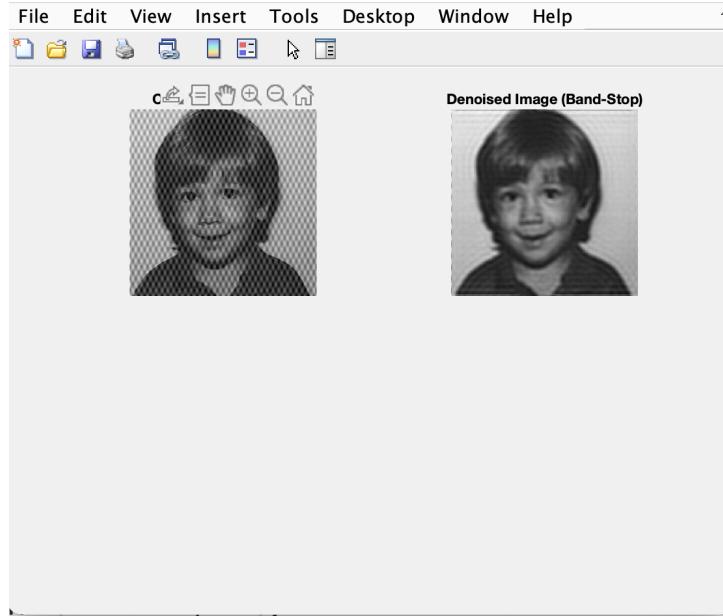


Figure 13: Denoised image obtained via frequency domain band Stop filtering.

3 Processing of Touch-tone (DTMF) Signals

3.1 DTMF System Overview

In the DTMF signaling scheme, each key is represented by the sum of a low frequency and a high frequency sinusoid. The frequencies are summarized in Table 1. The continuous-time signal model is

$$x(t) = A \sin(2\pi f_1 t) + A \sin(2\pi f_2 t), \quad (6)$$

where f_1 is one of the row frequencies and f_2 is one of the column frequencies.

Table 1: DTMF keypad frequencies.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

3.2 Signal Generation and Spectral Analysis (Tasks 2.1–2.2)

Row vectors $d0$ through $d9$ are generated to represent the DTMF signals for digits 0–9 at a sampling rate of 8192 Hz. The spectra of the signals corresponding to digits “2” and “9” are obtained using a 2048-point FFT. The resulting magnitude and squared-magnitude spectra exhibit distinct peaks at the frequencies prescribed by the DTMF standard, confirming the expected dual-tone structure of each digit.

3.3 Phone Number Construction and Noisy Signals (Tasks 2.3–2.4)

A seven-digit DTMF phone-number signal is formed by concatenating tone segments of 1000 samples, each separated by a 100-sample silent interval. Additive white Gaussian noise is introduced to obtain a specified signal-to-noise ratio (SNR), based on the average power of the

clean signal and the required noise variance. The resulting noisy waveform is displayed and subsequently rendered as audio using the `sound` function in MATLAB.

3.4 General DTMF Decoding Procedure (Task 2.5)

This section describes a general procedure for identifying a sequence of DTMF touch-tones encoded in an audio vector. Each touch-tone consists of 1000 samples and consecutive tones are separated by 100 samples of silence. The decoding method is based on segmentation and frequency-domain peak detection.

1. Signal Segmentation

Let $N_{\text{tone}} = 1000$ and $N_{\text{sil}} = 100$. If the phone number contains N_d digits, the starting index of the k -th tone segment is

$$n_{\text{start}}(k) = (k - 1)(N_{\text{tone}} + N_{\text{sil}}) + 1, \quad k = 1, \dots, N_d,$$

and the corresponding tone samples are extracted as

$$x_k[n] = x(n_{\text{start}}(k) + n), \quad n = 0, \dots, N_{\text{tone}} - 1.$$

This produces N_d isolated tone frames, each containing a single DTMF digit.

2. Fourier Transform of Each Tone

For each frame $x_k[n]$, a discrete Fourier representation is obtained using an N -point FFT (typically $N = 2048$):

$$X_k[m] = \text{FFT}\{x_k[n]\}, \quad m = 0, \dots, N - 1.$$

Only the positive-frequency portion of the magnitude spectrum,

$$|X_k[m]| = |X_k[m]|, \quad m = 0, \dots, N/2 - 1,$$

is required for frequency identification.

3. DTMF Frequency Candidates

The DTMF protocol specifies four “row” frequencies and four “column” frequencies:

$$f_{\text{low}} \in \{697, 770, 852, 941\} \text{ Hz}, \quad f_{\text{high}} \in \{1209, 1336, 1477, 1633\} \text{ Hz}.$$

Given sampling rate $F_s = 8192$ Hz and FFT size N , the associated FFT frequency grid is

$$f[m] = \frac{m}{N} F_s, \quad m = 0, \dots, N/2 - 1.$$

For each DTMF candidate frequency f_c , the nearest FFT bin index is determined by

$$m_c = \arg \min_m |f[m] - f_c|.$$

4. Row and Column Frequency Selection

For the k -th tone frame, the magnitude of the spectrum at the four row frequencies is evaluated:

$$E_{\text{row},r}(k) = |X_k[m_{\text{low},r}]|, \quad r = 1, \dots, 4.$$

Similarly, the magnitude at the column frequencies is computed:

$$E_{\text{col},c}(k) = |X_k[m_{\text{high},c}]|, \quad c = 1, \dots, 4.$$

The dominant row and column indices are then obtained as

$$r^* = \arg \max_r E_{\text{row},r}(k), \quad c^* = \arg \max_c E_{\text{col},c}(k).$$

5. Keypad Mapping

The pair (r^*, c^*) uniquely identifies a key on the 4×4 DTMF keypad:

$$\text{keypad} = \begin{bmatrix} 1 & 2 & 3 & A \\ 4 & 5 & 6 & B \\ 7 & 8 & 9 & C \\ * & 0 & # & D \end{bmatrix}.$$

The decoded digit for the k -th frame is given by

$$d_k = \text{keypad}(r^*, c^*).$$

6. Reconstruction of the Phone Number

Applying the segmentation, FFT analysis, peak detection, and keypad lookup for all $k = 1, \dots, N_d$, produces an estimated digit sequence

$$\{d_1, d_2, \dots, d_{N_d}\},$$

which corresponds to the phone number encoded in the audio vector. Because DTMF tones occupy well-separated spectral locations, this procedure remains reliable over a broad range of SNR conditions.

3.5 DTMF Decoder Implementation and Performance Evaluation (Task 2.6)

A general-purpose DTMF decoder function, denoted `ttdecode`, is developed to implement the procedure derived in Task 2.5. The function accepts as input a touch-tone signal sampled at 8192 Hz, assumes that each digit consists of 1000 samples followed by a 100-sample silence interval, and returns a 7-digit vector representing the decoded phone number. The function first partitions the input signal into seven consecutive tone segments, each of length 1000 samples, then computes a 2048-point FFT for each segment. By evaluating the spectral magnitudes at the eight DTMF candidate frequencies (four row and four column frequencies), the decoder identifies the dominant row–column pair and maps it to a keypad digit via the standard 4×4 DTMF lookup table.

To evaluate the performance of the decoder, a synthetic 7-digit phone-number signal is generated by concatenating seven tone segments and inserting 100-sample silent intervals between them. Additive white Gaussian noise is then introduced at different signal-to-noise ratios (SNRs) to simulate channel degradation. For an SNR value SNR_{dB} , the noise variance is chosen as

$$\sigma^2 = \frac{P_{\text{sig}}}{10^{\text{SNR}_{\text{dB}}/10}},$$

where P_{sig} denotes the average power of the clean touch-tone signal. The decoder is applied to increasingly noisy versions of the same phone-number signal, and the smallest SNR level at which the decoded sequence first deviates from the true sequence is recorded as the “break point” of the decoder. This break point provides an empirical measure of the robustness of the FFT-based DTMF decoding algorithm.

3.6 Decoding Provided Touch-tone Signals (Task 2.7)

The file `touchFS25.mat` contains two touch-tone recordings, denoted `x1` and `x2`, each corresponding to an unknown 7-digit phone number. The decoder developed in Task 2.6 is applied directly to these signals. Under the same assumptions regarding sampling rate, tone duration, and silence interval structure, the function `ttdecode` segments each recording into seven tone

frames and performs FFT-based frequency detection on each segment. The dominant row and column frequencies of each frame are identified and converted to keypad digits via the DTMF lookup table.

Let the resulting decoded sequences be

$$d^{(1)} = \{d_1^{(1)}, d_2^{(1)}, \dots, d_7^{(1)}\} \quad \text{and} \quad d^{(2)} = \{d_1^{(2)}, d_2^{(2)}, \dots, d_7^{(2)}\},$$

representing the phone numbers contained in x_1 and x_2 , respectively. These results demonstrate that the decoder generalizes to touch-tone signals not generated within the project and successfully extracts the corresponding telephone numbers.

4 Discussion and Conclusion

The Honors Option presented an examination of two-dimensional convolution methods and Fourier-domain techniques for image processing, together with the synthesis and interpretation of DTMF signaling. The results confirm that phase information is essential for maintaining structural components in images, whereas magnitude primarily influences overall contrast. They also show that frequency-domain filtering is effective in suppressing structured noise when the interference occupies identifiable regions of the spectrum.

In the context of DTMF processing, the investigation demonstrated that a simple FFT-based decoding procedure can accurately identify dialed symbols even under varying noise conditions. This outcome underscores the inherent robustness of the DTMF frequency allocation scheme and its suitability for reliable tone-based communication.

5 References

- ECE 366 Course Materials FS25, Dr. Panos, Michigan State University.
- A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed., Pearson, 2010. (Material on 1D/2D convolution and Fourier transforms)
- R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed., Pearson, 2018. (2D convolution, spatial filtering, and 2D Fourier transform)
- A. V. Oppenheim, *Signals and Systems*, 2nd ed., Prentice Hall, 1996. (Continuous-time and discrete-time convolution fundamentals)
- Bell Laboratories, “Dual-Tone Multi-Frequency (DTMF) Signaling Specifications,” Technical Documentation, 1960s–1970s. (Original design of DTMF frequencies)
- ITU-T Recommendation Q.23, “Technical Features of Push-Button Telephone Sets,” International Telecommunication Union, 1988. (Official DTMF signaling standard)
- J. A. Cadzow, “DTMF Detection by the Discrete Fourier Transform,” *IEEE Transactions on Communications*, vol. 31, no. 2, pp. 365–371, 1983. (FFT-based decoding of DTMF tones)
- Z. D. Zilic and K. R. Gadre, “Implementation of DTMF Receivers Using the Goertzel Algorithm,” *Bell Labs Technical Journal*, vol. 5, no. 2, 2000. (Efficient real-time decoding methods)