Video Processing
Lab 01
Inés Vergés Vega (u213900)
Daniela Quimis (u199759)

Task 1

We aim to sample $\psi(x, y) = cos((5a + 1)y)cos(10(5 - b)x)sin(5(a + b)(x + y))$ with a lattice matrice defined as: V = [1/40 0; 0 1/20]. 1/40 represents the sampling interval in the x-axis, while 1/20 represents the sampling interval in the y-axis.

In this report, a = 6 and b = 7, as they correspond to the last digit of the NIAs of the authors of this report. Thus, we can rewrite the function as follows:

$$\psi(x, y) = cos(31y)cos(- 20x)sin(60(x + y))$$

a) Using sampling matrix V, define a sample grid and use it to obtain the sampled signal $\psi_{sampled}$ $(\boldsymbol{n})$ = ψ $(\boldsymbol{Vn})$.

The lattice space can be thought of as the set of all sampled points that result from transforming a continuous signal, $\psi(x, y)$, with the sampling matrix V, at regular intervals.

Once we defined the sampling matrix and the function, we created the sampling grid consisting of 20 samples, which we deemed sufficiently informative about the signal.

Next, we located the samples on the grid b substituting in the giving formula. We find that the sampled signal $\psi_{sampled}$ $(\boldsymbol{n})$ is equal to the continuous signal evaluated at all points of the grid (thus including all samples).

Finally, we proceed to plot the grid and the sampled signal.

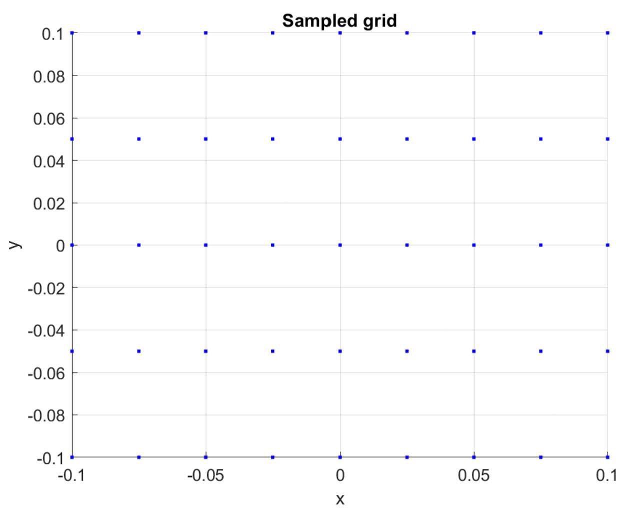


Figure 1.1. Sampled grid

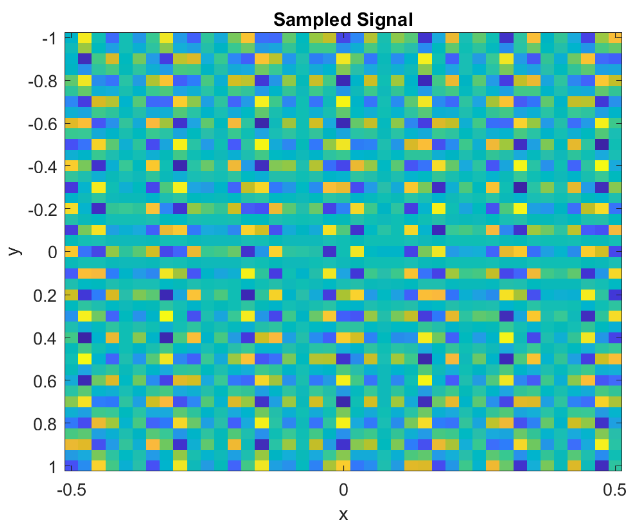
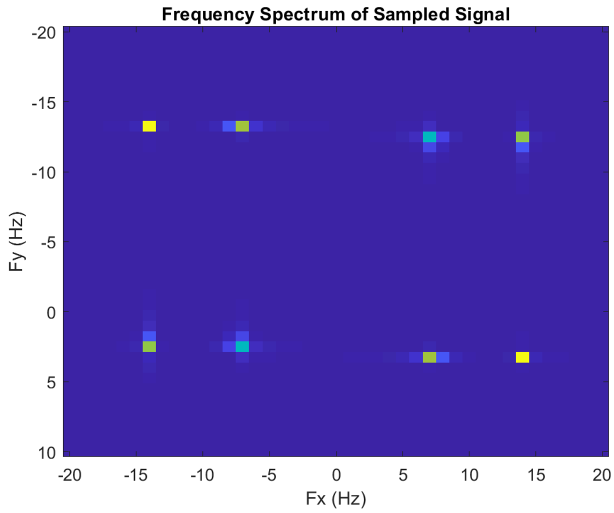


Figure 1.2. Sampled Signal

b) Calculate the spectrum of the sampled signal obtained in 1.a.

The signals' frequency spectrum is calculated with the discrete space Fourier Transform (DSFT), in 2 dimensions, as $\psi(x, y)$ depends both on *x* and *y*. When plotting the transform, we focus on the magnitude to distinguish the phase from the amplitude, which enhances visualisation and frequency analysis, as discussed in the next question.



We can observe in this figure (Figure 1.3) that there are eight 'points'. These are deltas that arise from the combinations of all the frequency components. By decomposing $\psi(x, y)$ with exponentials, we would see it clearly.

Figure 1.3. Plot of the sampled signal's frequency spectrum.

c) Discuss whether the signal obtained in 1.a suffers from aliasing or not. Provide a clear justification for your answer.

For a signal not to suffer from aliasing, its frequency support must be contained in the Voronoi's cell, U = (U_1, U_2).

This cell is formed by uniting the midpoints of every possible combination between U_1 and U_2. In other words, the distance between the origin and the calculated point must be less or equal to any point of the reciprocal lattice. The matrix is computed as $U = (V^T)^{-1}$.

Once the midpoints are computed, we form two arrays, the one containing the vertical points and the one containing the horizontal ones. Then, we plot the frequency spectrum of the continuous signal and on top, the cell.

We can observe that the frequencies of $\psi(x, y)$ are not within the Voronoi cell. Thus, this signal suffers from aliasing.

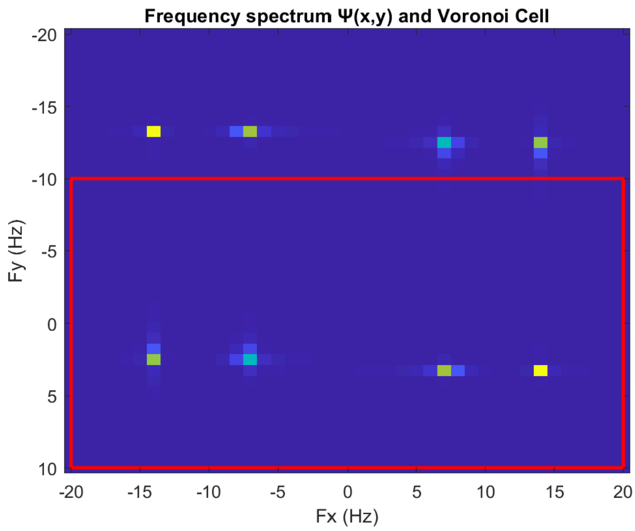


Figure 1.4: Frequency spectrum of the signal and the Voronoi cell

Task 2

a) Analyze the temporal frequencies of the pixel at position (5 × c, 5 × d), where a and b are the same ones defined in Task 1.

“c” and “d” correspond to the sum of all the digits of the NIA of the first student and the second, respectively. That is: ‘’c” = 2+5+4+6+1+7= 25 and ‘’d”=2+6+6+8+4+6= 32. The pixel at position (x,y) = (125,160) is the one we had to analyze. The video we are working with is six seconds long, has 720 frames and a frame rate of 120 fps. The frame size is 317 x 423 so 317 pixels in height and 423 pixels in width. This confirms that the video is indeed grayscale, and the pixel intensities are represented as a 2D matrix with dimensions `(317, 423)` for each frame.

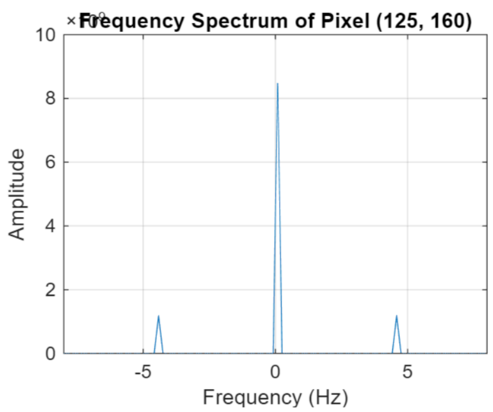
After defining the parameters, an array `intensities` is preallocated to store the intensity values of the chosen pixel for all frames in the video. Consequently, the loop reads each frame of the video, extracts its intensity value and stores it in the array at the pixel (160,125).

```
intensities = zeros(1,video_obj.NumFrames);

for i=1:video_obj.NumFrames
    vidframe = readFrame(video_obj,'native');
    intensities(i) = vidframe.cdata(pixel_y,pixel_x);
end
```

Figure 2.1: Code of the array initialization.

Then, we computed the Fourier Transform. A Fast Fourier Transform (FFT) is applied to the intensity signal to analyze its frequency components. The `fftshift` function centres the zero-frequency component for easier interpretation. The squared magnitude (`magnitude`) represents the power spectrum. The corresponding frequency axis is computed based on the total number of frames and the signal duration (6 seconds).




```
Peak Frequencies:
    0.0833    4.4167

maxFreq = 4.4167
```

Figure 2.2. Frequency spectrum of the pixel (125,160)

We used MATLAB's `findpeaks` function to identify the frequencies of the spectrum (See Figure 2.2). Out of both, the maximum frequency corresponds to the highest peak, 4.416 Hz.

Task 2.b: Based on your analysis from (2.a), indicate what would be the minimum sampling frequency that you could use so that you can recover correctly the fundamental frequency for the specific pixel of the pixel selected in Task 2.a

To determine the minimum sampling frequency required to correctly recover the fundamental frequency at the pixel, we must use the Nyquist-Shannon Sampling Theorem. The maximum frequency present in the signal is $F_{max} = 4,4167\ Hz$. According to the theorem

$F_{sampling} \geq 2 \cdot F_{max}$ so $F_{sampling} \geq 8,8334\ Hz$

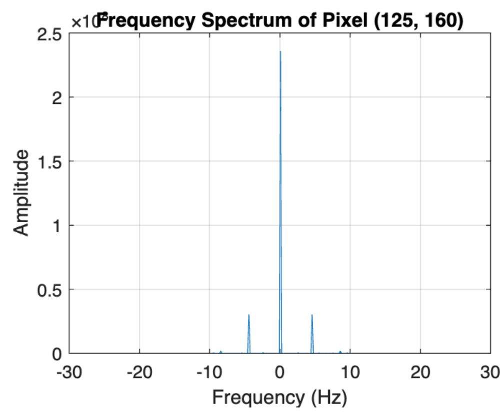
Sampling at this frequency ensures there is no aliasing**.** In our case , the sampling rate is 120 Hz (much higher than the Nyquist requirement), reducing it to at least 8.8334 Hz will still allow us to accurately recover the fundamental frequency of the pixel signal at 4.4167 Hz.

Task 2.c: Select a frequency close to (but slightly higher than) the minimum frequency that you determined in 2.b and generate a new video = the original video that was provided but now sampled at the frequency that you chose(*). Repeat the analysis of Task 2.a with this new video to confirm whether your result is correct and you can still recover the fundamental frequency of the pixel (5 × c, 5 × d).

We selected a 20 Hz frequency which ensures that the Nyquist criterion is satisfied, to downsample the video to 20 fps, we keep one frame for every six frames (120÷20=6).

Once obtained the new video, we analysed its frequencies as task2.a . By changing the sampling rate and accessing our 'videoReduce.avi' we ended up obtaining the same graph in task2.a, as seen in Figure 2.

A peak at 4.4167 Hz, confirms that the fundamental frequency is preserved in the downsampled video at 20 Hz. This validates that the downsampled video at 20 Hz sampling frequency is sufficient to recover the fundamental frequency of the pixel at position (125, 160).




```
Peak Frequencies:
    0.0833    4.4167

Max Frequency: 4.4167 Hz
```

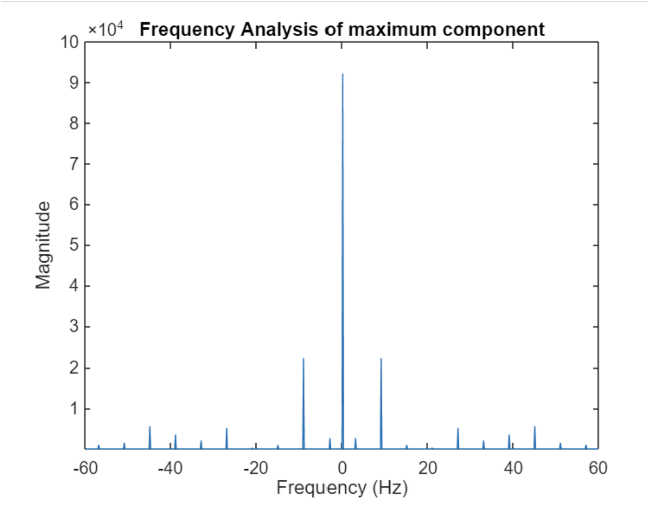
Figure 2.3: Frequency spectrum of the downsampled video

Task2.d: Now extend the analysis of temporal frequencies to the whole video and indicate what is the minimum frame rate at which we could sample so that we do not have aliasing that affects the fundamental frequencies at any part of the video.

Equivalently, so that the perceived rotation speed is the same as in the original video, for all pixels.

Instead of doing this for every frequency, we only considered the positive ones, as the spectrum is symmetric, to reduce the execution time.

The idea was to analyze the frequency of all pixels and frames and extract the highest one. Then, we could know the maximum sampling rate by the Nyquist criterion




```
Peak Frequencies:
    0.0833    8.9167    9.0833   44.9167   45.0833
```

Figure 2.4: Frequency spectrum of the original video

In Figure 2.4 we can observe that the maximum frequency present in the signal is 45.0833 Hz. However, we're not going to be considering it, neither will the 44.9167 frequency, because they are artifcacts, as their magnitude is relatively lower than 9.0833 Hz. Thus, the minimum sample frequency used to avoid aliasing is f_s = 2* 9.0833Hz ≃ 18 Hz.

We created a video named 'videotask2d' to see that there was no apparent aliasing when the sampling frequency was 20 Hz. 20 because it's a rate close to 18 but divisible by 120.

Task2.e: **Select a frequency that is approximately 40% of the minimum frequency that you determined in 2.d and generate a new video = the original video that was provided but now sampled at the frequency that you chose. Inspect the generated video to identify elements that suggest the type of aliasing that would be expected from the theory. Report your findings in a clear and concise manner**

The **minimum frequency** is likely the smallest **non-zero significant peak**, which is **8.9167Hz**. Calculating the 40% of $8,9167 \cdot 2 \simeq 18\,Hz$ is:

$$F_{selected} = 0.4 \cdot 18 = 7,2\,Hz$$

To simulate aliasing, we'll change the sampling rate of the video to 7,2 Hz. The motion of elements in the video appears significantly slower than normal, which can be attributed to temporal aliasing in the time domain. This results in an inability to capture smooth transitions between frames, causing objects to move unnaturally. Additionally, some objects seem to reverse direction which is characteristic of the "wagon-wheel effect" commonly associated with spatial-temporal aliasing. This artifact occurs when an object's motion exceeds the sampling capability, causing it to appear as if it is moving backwards or erratically.

Furthermore, the video shows signs of spatial aliasing, as some objects appear distorted or misaligned, suggesting that the sampling rate in space is not sufficient to capture their true position. The overall effect is a noticeable irregularity in both the speed and path of moving objects, which aligns with the theoretical expectations for spatial-temporal aliasing at this sampling rate of 7,2 Hz.