

## Tarea 2

### Instrucciones generales

- La tarea se realiza en grupos de máximo 3 personas. Cada grupo debe escribir el nombre de los integrantes del grupo en la siguiente dirección electrónica:

[https://docs.google.com/spreadsheets/d/1e-biLZ4gGOG5v\\_TioklGjgjlnoKavsA2](https://docs.google.com/spreadsheets/d/1e-biLZ4gGOG5v_TioklGjgjlnoKavsA2)

El número del grupo está indicado en la primera columna del documento.

- Todos los archivos de esta tarea se encuentran en la carpeta de *One Drive* del curso.
- Los archivos computacionales implementados en GNU Octave y Python deben estar correctamente comentados. Por cada archivo que no este documentado correctamente, se restaran 5 puntos de la nota final. **Si alguna función o archivo computacional está incompleto o genera error al momento de compilar, entonces pierde el 75% del puntaje de la pregunta asignada.**
- El documento escrito para cada parte de la tarea debe tener su respectiva bibliografía. **En caso de no presentar la bibliografía, se le restará 5 puntos por cada documento que no contenga referencias bibliográficas.**
- Los archivos que dan solución a la tarea deben estar en una carpeta principal con nombre **Tarea 2 - Grupo #**, donde # es el número de cada grupo. Dentro de esta carpeta debe existir dos carpetas con nombres **Parte 1** y **Parte 2**. En cada una de estas carpetas estarán todos los archivos necesarios para el desarrollo de las preguntas mencionadas anteriormente.
- La solución de la tarea que se encuentra en la carpeta **Tarea 2 - Grupo #** debe comprimirse en un archivo **.zip** y subirlo al formulario que se encuentra en el siguiente enlace:

<https://forms.gle/ufnZ6dGWXCniv5T78>

#### Observaciones:

- Se necesita tener una cuenta de **gmail** para llenar el formulario.

- **Fecha y hora máxima de entrega: Domingo 29 de Setiembre del 2023, a las 11:59 pm**
- Las entregas tardías se realizarán al correo penalizarán con una reducción de la nota obtenida con un 10% por cada hora de atraso. A las tareas que excedan el plazo de entrega en 10 horas o más después de la hora límite, se les asignará la nota de 0. **SOLAMENTE las entregas tardías se realizarán al correo [jusoto@tec.ac.cr](mailto:jusoto@tec.ac.cr).**

# Parte 1: Restauración de Imágenes

## Descripción General

- La primera parte de la tarea consiste en la técnica de restauración de imágenes (*inpainting*, en inglés).
- El algoritmo a desarrollar se encuentra en el artículo *Fast Digital Image Inpainting*. Este algoritmo se basa en la convolución de matrices.
- La implementación computacional de esta parte se debe realizar en GNU Octave.
- La parte escrita debe estar en un documento pdf con nombre `tarea2.parte1`.

## Preguntas

- **Pregunta 1:** En GNU Octave, si se desea obtener las entradas que cumplen una condición booleana o acceder a algunas de las entradas de una matriz para cambiar su valor, no se recomienda utilizar un doble bucle `for`, ya que el tiempo de ejecución se incrementa significativamente. Para evitar este problema en esta parte de la tarea, se recomienda usar la estructura de programación que tiene GNU Octave. Esta pregunta tiene la finalidad de explicar como utilizar esta estructura. Para eso, considere la matriz de 3 dimensiones  $A \in \mathbb{R}^{4 \times 4 \times 3}$ , donde

$$A(:, :, 1) = \begin{pmatrix} 5 & 10 & 15 & 20 \\ 4 & 8 & 12 & 16 \\ 3 & 6 & 9 & 12 \\ 2 & 4 & 6 & 8 \end{pmatrix}, \quad A(:, :, 2) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \\ 1 & 4 & 7 & 10 \end{pmatrix}, \quad A(:, :, 3) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 \\ 10 & 10 & 10 & 10 \\ 15 & 15 & 15 & 15 \end{pmatrix}.$$

- (a) **[Valor 5 puntos]:** Leer y ejecutar el archivo '`obtener_indices_matriz.m`'. Utilizando la matriz  $A$ , implemente un *script* en GNU Octave que permita obtener una matriz de 2 dimensiones  $B \in \mathbb{R}^{4 \times 4}$ , tal que  $B(i, j) = 1$  si  $A(i, j, 1) > 5$ ,  $A(i, j, 2) < 5$  y  $A(i, j, 3) \geq 10$ ; y  $B(i, j) = 0$  en caso contrario. El nombre del archivo debe ser '`parte2.p1a.m`'. **Observación:** No debe utilizar ningún bucle para llegar al resultado final.
- (b) **[Valor 5 puntos]:** Leer y ejecutar el archivo '`evaluar_matrices.m`'. Utilizando la matriz  $A$ , implemente un *script* en GNU Octave que permita obtener una matriz de 3 dimensiones  $C \in \mathbb{R}^{4 \times 4 \times 3}$ , tal que  $C(i, j, k) = -30$  si  $A(i, j, k) \leq 10$  y  $C(i, j, k) = 30$  si  $A(i, j, k) > 10$ . El nombre del archivo debe ser '`parte2.p1b.m`'. **Observación:** No debe utilizar ningún bucle para llegar al resultado final.
- **Pregunta 2 [Valor 10 puntos]:** Leer el documento '`restauracion_de_imagen.pdf`' y desarrollar una infografía del tema de restauración de imágenes.

**Sugerencias:** (1) Leer los documentos '`infografia1.pdf`' e '`infografia2.pdf`' para poder crear una infografía. (2) Para desarrollar una infografía en internet, pueden usar la aplicación que se encuentra en la página <https://www.visme.co/>.

- **Pregunta 3 [Valor 30 puntos]:** Implemente computacionalmente en GNU Octave el algoritmo presentado en el artículo científico *Fast Digital Image Inpainting*, considerando la siguiente información:

- La imagen original se encuentra en el archivo '**paisaje.jpg**' [Ver Figura 1(a)].
- La región a restaurar se encuentra en la imagen binaria '**marca.jpg**', donde las partes en color blanco son las regiones a restaurar [Ver Figura 1(b)].
- Ambas imágenes tienen el mismo tamaño ( $320 \times 662$ ).
- El código desarrollado en el archivo '**generar\_imagen\_restaurar.m**' genera la imagen que combina la imagen '**paisaje.jpg**' y '**marca.jpg**' [Ver Figura 1(c)].
- El objetivo de esta pregunta es realizar la restauración de la imagen de la Figura 1(c), utilizando el algoritmo del artículo científico *Fast Digital Image Inpainting* [Ver Figura 1(d)].
- La implementación computacional no debe utilizar bucles para obtener el resultado final. Para eso, usar la idea explicada en la Pregunta 1.
- Utilizar el índice de calidad de imagen SSIM para ver la calidad de la imagen reconstruida.



Figura 1: Imágenes de la Pregunta 3 - Parte 1

- **Pregunta 4 [Valor 10 puntos]:** El algoritmo propuesto en el artículo *Fast Digital Image Inpainting* también permite eliminar ciertos objetos de una imagen [Ver Figura 2].

Cada grupo debe escoger tres imágenes a color tomadas en el TEC y donde aparezca en cada imagen al menos uno de los integrantes del grupo. Además, en cada imagen debe aparecer un objeto seleccionado arbitrariamente (un lápiz, un anillo, una cuerda, entre otros). El objetivo de esta pregunta es eliminar ese objeto de la imagen, utilizando el algoritmo propuesto en el artículo *Fast Digital Image Inpainting*. **Sugerencia:** El objeto debe identificarse claramente en la imagen, y no debe ser un objeto que abarque mucho espacio de la imagen.



Figura 2: (a) Imagen Original. (b) Imagen Binaria del Objeto a Eliminar. (c) Imagen Reconstruida sin el Objeto.

## Parte 2: DFT para Imágenes a Color

### Descripción General

- La segunda parte de la tarea consiste en implementar una versión de la transformada discreta de Fourier (DFT) para imágenes a color. Esta transformada fue desarrollada en el artículo *Complex and hypercomplex discrete Fourier transforms based on matrix exponential form of Euler's formula*, desarrollado por S. J. Sangwin y T. A. Ell. Las implementaciones computacionales se deben realizar en **Python**.
- Sea  $A$  una imagen a color de tamaño  $m \times n$ . Sea  $A(x, y)$  un pixel en la posición  $(x, y)$  de la imagen  $A$ , tal que  $A(x, y) = [A_r(x, y), A_g(x, y), A_b(x, y)]$ , donde  $A_r(x, y)$ ,  $A_g(x, y)$  y  $A_b(x, y)$  son los valores del pixel en los canales rojo, verde y azul, respectivamente.
- La DFT-2D hipercompleja de una imagen a color  $A$  se define como la matriz  $F$  de tamaño  $m \times n$ , donde cada una de sus entradas  $F(u, v)$  es una matriz de tamaño  $4 \times 4$ , para todo  $u = 1, 2, \dots, m$  y  $v = 1, 2, \dots, n$ . Cada entrada  $F(u, v)$  se calcula a partir de la siguiente fórmula:

$$F(u, v) = \frac{1}{\sqrt{mn}} \sum_{r=0}^{m-1} \sum_{s=0}^{n-1} E(r, u, m) \mathcal{F}[A(r, s)] E(s, v, n),$$

donde:

- $\mathcal{F}$  transforma un pixel a color  $A(x, y) = [A_r(x, y), A_g(x, y), A_b(x, y)]$  en una matriz de tamaño  $4 \times 4$ , a través de la siguiente fórmula:

$$\mathcal{F}[A(x, y)] = \begin{bmatrix} 0 & -A_r(x, y) & -A_g(x, y) & -A_b(x, y) \\ A_r(x, y) & 0 & -A_b(x, y) & A_g(x, y) \\ A_g(x, y) & A_b(x, y) & 0 & -A_r(x, y) \\ A_b(x, y) & -A_g(x, y) & A_r(x, y) & 0 \end{bmatrix}.$$

- $E(p, q, r)$  es una matriz de tamaño  $4 \times 4$  que generaliza el contexto de la fórmula de Euler, es decir:

$$E(p, q, r) = e^{-2\pi J \frac{pq}{r}} = I_4 \cos\left(2\pi \frac{pq}{r}\right) - J \sin\left(2\pi \frac{pq}{r}\right),$$

donde  $J \in \mathbb{C}^{4 \times 4}$  tal que  $J^2 = -I_4$ , donde  $I_4$  es la matriz identidad de tamaño  $4 \times 4$ .

**Nota:** La matriz  $J$  no es única.

### Preguntas

- **Pregunta 1 [Valor 10 puntos]:** Investigue y encuentre dos matrices  $J \in \mathbb{C}^{4 \times 4}$  tal que  $J^2 = -I_4$ . Una de las matrices debe tener entradas solo reales. Escriba las dos matrices en un archivo con nombre `parte2_p1.m`.

- **Pregunta 2 [Valor 30 puntos]:** Implemente computacionalmente en Python la DFT-2D hiper-compleja para determinar la frecuencia/espectro de la imagen `lena.jpg` (ver Figuras 5(a) y 5(b)). **Utilice como matriz  $J$  la matriz de entradas reales obtenida en la Pregunta 1.** Para obtener la frecuencia/espectro que se encuentra en la Figura 5(b), utilice la representación logarítmica vista en clases, usando una norma matricial en lugar del valor absoluto para cada una de las entradas de la matriz que representa la DFT-2D hipercompleja. La implementación computacional se debe realizar en un archivo con nombre `parte2_p2.m`.

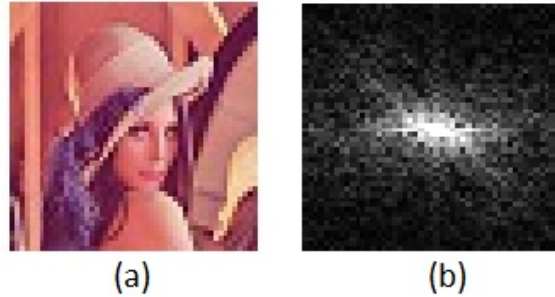


Figura 3: (a) Imagen Lena (b) Frecuencia/Espectro de la Imagen Lena