

## Tarea 2

### Instrucciones

- La tarea se realiza en grupos de máximo 3 personas. Los archivos relacionados a la tarea se encuentran en el TEC- Digital, en la sección de Documentos->Tareas ->Tarea 2.
- 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.

### Parte 1: Eliminación de Ruido del Tipo *Salt and Pepper* en un video

#### Descripción General

- Esta parte de la tarea se basa en eliminar el ruido de un video, usando dos tipos de filtros.
- La implementación computacional se debe realizar en GNU Octave.
- La parte escrita debe estar en un documento con nombre **Tarea 2 - Parte 1.pdf**. Todos los resultados numéricos y ejemplos imágenes del video deben aparecer en el documento, y explicado con detalle y claridad.

#### Preguntas

1. **[15 puntos]** Explique en que consisten los Algoritmos *Fast Median Filter Approximation* (ver Algoritmo 1) y IAMFA-I (verl Algoritmo 2). Para eso escriba el pseudocódigo de cada algoritmo algoritmo. Para explicar dichos algoritmos, pueden utilizar un dibujo como se muestran en las Figuras 2 y 3.
2. **[5 puntos]** Crear un video donde aparezcan todos los integrantes del grupo. Este video no debe durar menos de 5 segundos ni más de 10 segundos. El nombre del archivo debe ser `video_con_ruido.mp4`. Este video debe contener ruido del tipo *salt and pepper*. Para crear dicho ruido, deben usar el comando `imnoise` del paquete `image` de GNU Octave. Información del comando `imnoise` se puede encontrar en el siguiente enlace:

<https://octave.sourceforge.io/image/function/imnoise.html>

3. **[25 puntos]** Implemente computacionalmente en GNU Octave los Algoritmos *Fast Median Filter Approximation* y IAMFA-I, para eliminar el ruido del video `video_con_ruido.mp4`. El resultado final debe ser un video sin ruido con los nombres `video_sin_ruido_alg1.mp4` y `video_sin_ruido_alg2.mp4`, según corresponda a cada algoritmo implementado. El nombre del archivo donde se encuentran los algoritmos debe ser `filtros_mediana_modificada.m`.
4. **[10 puntos]**: El índice de similitud estructural (SSIM de sus siglas en inglés) es un método para determinar la similitud entre dos imágenes. El índice SSIM resultante entre dos imágenes es un valor decimal en el intervalo  $[0, 1]$ , y el valor 1 solo es accesible en el caso de dos imágenes idénticas, por lo tanto, indica una similitud estructural perfecta. Un valor de 0 indica que no hay similitud estructural entre las imágenes.
  - Utilice la función `ssim` para determinar cual de los dos métodos utilizados en la Pregunta 3 realiza una mejor limpieza del video `video_con_ruido.mp4`. Justifique su respuesta. Para eso, deben utilizar el video original sin ruido. El nombre del archivo de la implementación computacional debe ser `resultado_ssim.m`.

## Parte 2: 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 **Tarea 2 - Parte 2**.

### Preguntas

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.
2. **[7 puntos]:** Leer el documento '`restauracion_de_imagen.pdf`' y desarrollar una infografía del tema de restauración de imágenes.
  3. **[20 puntos]** Implemente computacionalmente en GNU Octave el algoritmo presentado en el artículo científico *Fast Digital Image Inpainting*, considerando la siguiente información:
    - (a) Los archivos a utilizar se encuentran en la carpeta con nombre **Parte 1 - P2**.
    - (b) La imagen original se encuentra en el archivo '`paisaje.jpg`' [Ver Figura 1(a)].
    - (c) 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)].
    - (d) Ambas imágenes tienen el mismo tamaño ( $320 \times 662$ ).
    - (e) 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)].
    - (f) 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)].
    - (g) La implementación computacional no debe utilizar bucles para obtener el resultado final. Para eso, usar la idea explicada en la Pregunta 1.



Figura 1: Imágenes de la Pregunta 3 - Parte 2

4. [8 puntos] El algoritmo propuesto en el artículo *Fast Digital Image Inpainting* también permite eliminar ciertos objetos de una imagen [Ver Figura 2].

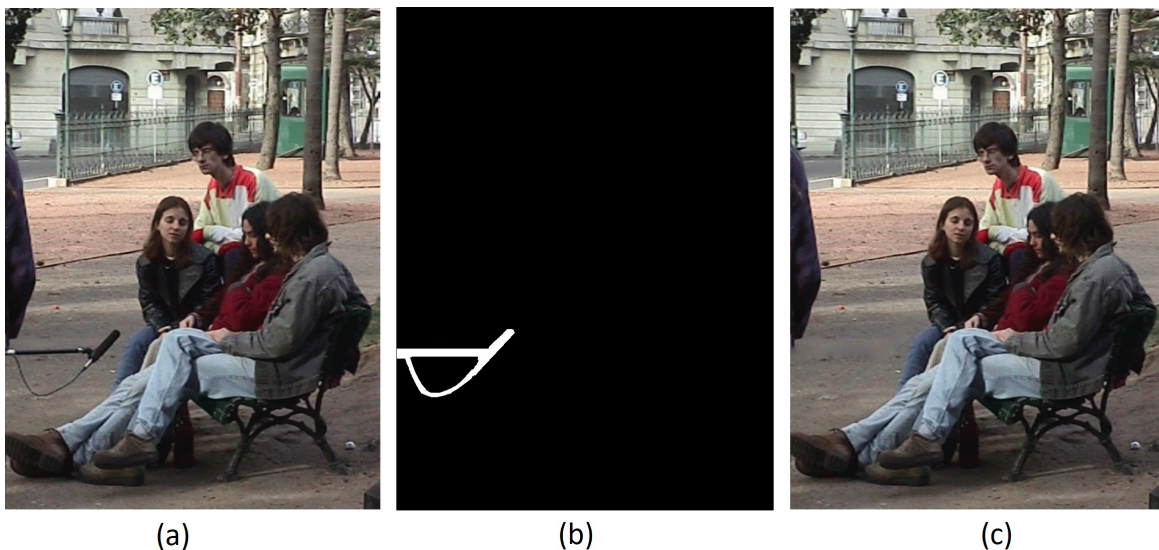


Figura 2: (a) Imagen Original. (b) Imagen Binaria del Objeto a Eliminar. (c) Imagen Reconstruida sin el Objeto.

**Cada grupo debe escoger una imagen a color donde aparezca, al menos, uno de los integrantes del grupo.** Además, en la 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.

## Información de la Entrega

- **Fecha y hora límite:** Miércoles 25 de Noviembre a las 11:59 pm.
- Los documentos deben estar guardados usando la siguiente estructura: Una carpeta principal con nombre **Tarea 2**. Dentro de esta carpeta debe existir tres 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.
- Deben enviar la carpeta **Tarea 2** en formato **zip** al correo **jusoto@tec.ac.cr**, con el encabezado **Entrega Tarea 2 - PAID**. En el cuerpo del correo deben indicar el nombre completo de los miembros del grupo.
- La entrega tardía de la tarea se penalizará con una reducción del 25% de la nota final, por día de atraso. A las tareas que excedan el plazo de entrega en 3 días o más después de la fecha límite, se les asignará la nota de 0.

## Defensa

- Cada grupo debe defender esta tarea frente al profesor. Para eso deben seleccionar un horario de la siguiente dirección electrónica:

<https://doodle.com/poll/4niymzdmuaqkgrtm>

- Deben escribir el nombre (sin apellidos) de todos los miembros del grupo y seleccionar uno de los horarios disponibles.
- Todos los miembros del grupo deben estar presentes para defender cada una de las preguntas. Si un estudiante no esta presente, entonces el estudiante perderá 35 puntos de la nota final.