

Tarea 3

Instrucciones

- La tarea se realiza en grupos de 3 o 4 personas. Los archivos relacionados a la tarea se encuentran en el TEC - Digital, en la sección de Documentos->Tareas->Tarea 3.
- 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: Croma o Clave de Color

- La primera parte de la tarea consiste en una técnica llamada croma o clave de color. La parte escrita debe estar en un documento pdf con nombre **Tarea 3 - Parte 1**. Todos los resultados numéricos e imágenes deben aparecer en el documento, y explicado con detalle y claridad. Las implementaciones computacionales se deben realizar en **GNU Octave**.
- La croma es una técnica audiovisual utilizada ampliamente tanto en cine, televisión y fotografía, que consiste en extraer un color de una imagen o vídeo (usualmente el verde) y reemplazar el área que ocupaba ese color por otra imagen o vídeo.
- Sea A una imagen a color de tamaño $m \times n$ usando el modelo RGB. La imagen A tiene un fondo de color verde [Ver Figura 1(a)]. Sean $A_r \in \mathbb{R}^{m \times n}$, $A_g \in \mathbb{R}^{m \times n}$ y $A_b \in \mathbb{R}^{m \times n}$ los canales rojo, verde y azul de la imagen A , respectivamente [Ver Figuras 1(b), 1(c) y 1(d)]. Como se observa en las imágenes de la Figura 1, un pixel (i, j) de A es de color verde si y solo si el pixel (i, j) de A_r y A_b son de tonalidad negra y el pixel (i, j) de A_g es de tonalidad blanca.
- Matemáticamente, lo anterior se puede representar de la siguiente manera:

$$A(i, j) = \text{■} \iff A_r(i, j) = 0 \wedge A_g(i, j) = 255 \wedge A_b(i, j) = 0,$$

para todo $i = 1, 2, \dots, m$ y $j = 1, 2, \dots, n$.

- Sea B una imagen a color de tamaño $m \times n$. La imagen B representa el nuevo fondo que se le quiere poner a la imagen A , en lugar del fondo en color verde [Ver Figuras 2(a) y 2(b)]. Sea C la imagen de tamaño $m \times n$ resultante de realizar el cambio de fondo en las partes verdes de la imagen A , utilizando la imagen B [Ver Figura 2(c)].
- Un pseudocódigo para obtener la nueva C es el siguiente:

```
for  $(i, j) \in \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$  do
    if  $A(i, j) = \text{■}$  then
        |  $C(i, j) = B(i, j)$ 
    else
        |  $C(i, j) = A(i, j)$ 
    end
end
```

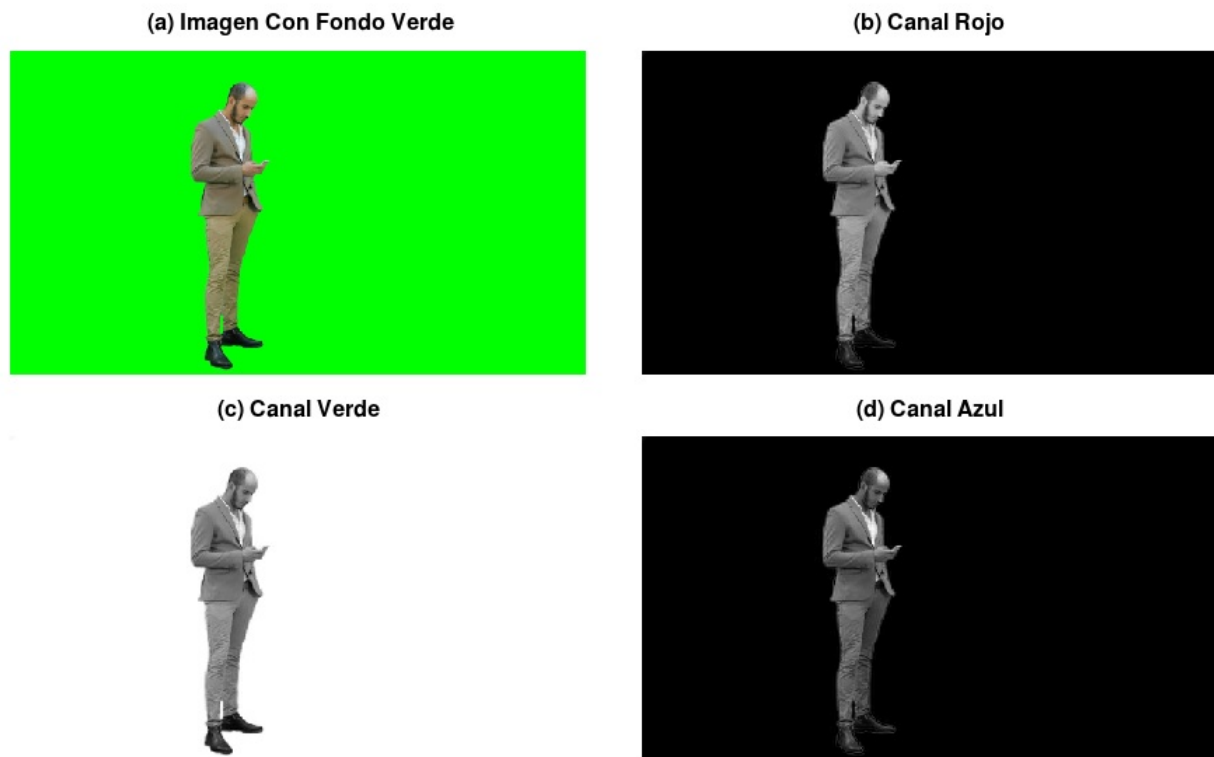


Figura 1: (a) Imagen con Fondo Verde. (b) Canal Rojo. (c) Canal Verde. (d) Canal Azul.

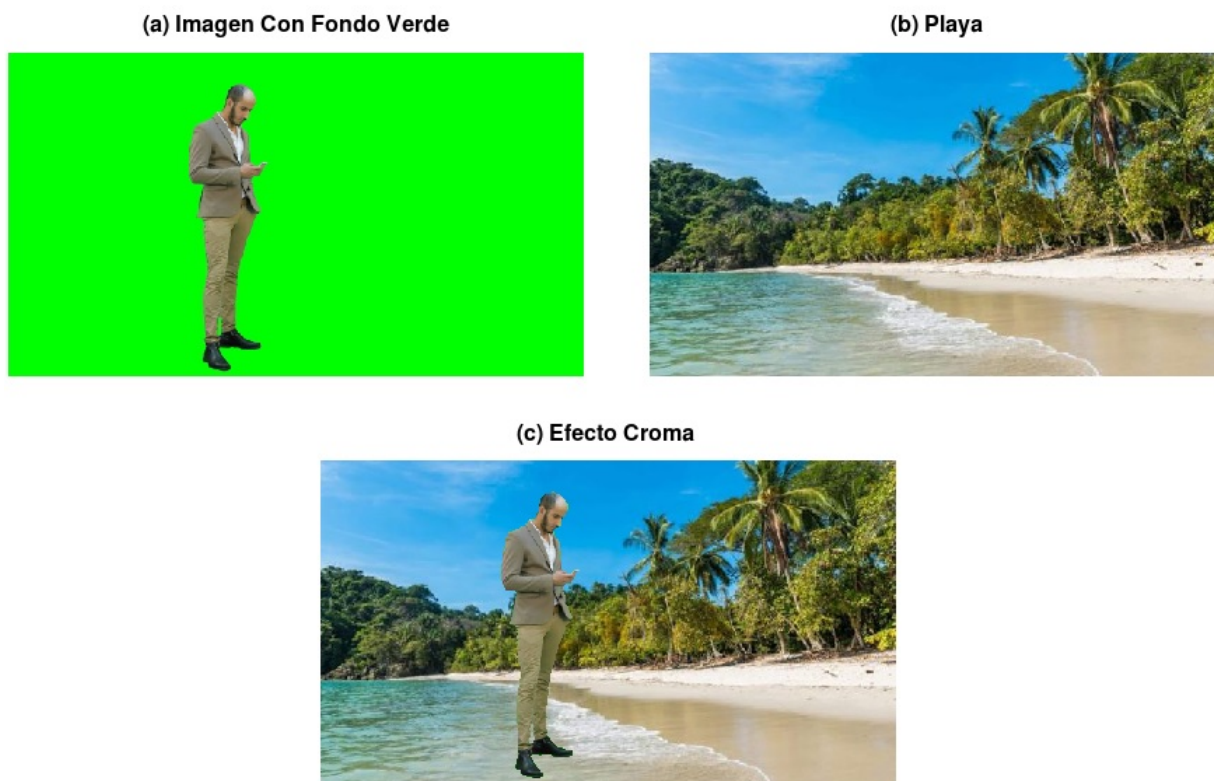


Figura 2: (a) Imagen con Fondo Verde. (b) Imagen del Nuevo Fondo. (c) Proceso de Croma.

Preguntas - Valor: 45 puntos

- **Pregunta 1 [Valor 20 puntos]:** Implemente computacionalmente en GNU Octave el proceso de croma para sustituir el fondo verde de la imagen 'fondo_verde.jpg' por la imagen 'playa.jpg' [Ver Figuras 2(a) y 2(b)]. El nombre del archivo debe ser 'parte1.p1.m'. **Observaciones:** (1) No deben utilizar ningún bucle para llegar al resultado final. (2) Utilicen un valor de tolerancia, ya que los valores negros y blancos en los canales RGB de la imagen 'fondo_verde.jpg' no siempre son 0 y 255, respectivamente.
- **Pregunta 2 [Valor 15 puntos]:** El resultado final de la imagen obtenida en la Pregunta 1 puede generar un color verde (no puro) en los bordes de la figura incrustada [Ver Figura 3(a)]. Esto se debe a la mezcla de colores en el borde entre el fondo verde y el objeto. Para reducir el color color verde del borde, se utiliza una imagen binaria que representa el borde de la imagen incrustada [Ver Figura 3(c)]. Luego, se aplica un proceso de restauración, utilizando como referencia la parte blanca de la imagen binaria [Ver Figura 3(b)].

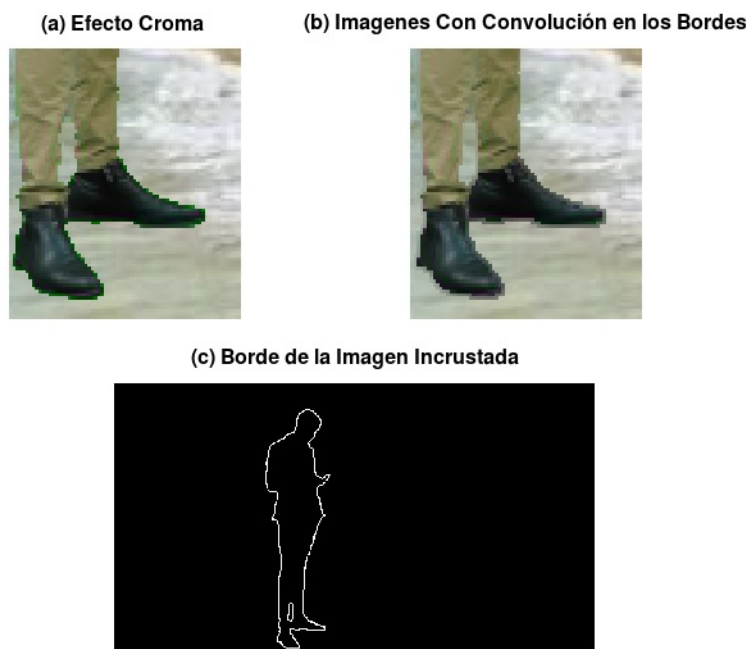


Figura 3: (a) Bordes de Color Verde después del Proceso de Croma. (b) Bordes de Color Verde Atenuados con Convolución. (c) Borde la Imagen Insertada.

En esta pregunta, deben modificar el código desarrollado en la Pregunta 2, de tal manera que se disminuya el color verde de los bordes de la imagen resultante. **Para eso, utilice el algoritmo desarrollado en el Parte 2 de la Tarea 2.** El nombre del archivo debe ser 'parte1.p2.m'. **Observaciones:** (1) No debe utilizar ningún bucle para recorrer las entradas de las matrices. Solo se permite un bucle para eliminar el color verde del borde. (2) Para obtener el borde de la imagen, como se muestra en la Figura 3(c), se recomienda obtener una imagen binaria del canal verde de la imagen 'fondo_verde.jpg', y luego utilizar algún operador morfológico para obtener solo el borde.

- **Pregunta 3 [Valor 10 puntos]:** Realizar el proceso de croma en un video, utilizando los siguientes archivos:
 - 'video_avion.mp4', el cual contiene un video de un avión con fondo verde [Ver Figura 4(a)].
 - 'video_cielo.mp4', el cual contiene un video del cielo [Ver Figura 4(b)].

El resultado debe ser un video con nombre 'video_croma.mp4', que contiene al avión desplazándose en el cielo [Ver Figura 4(b)]. El nombre del archivo debe ser 'parte1.p3.m'. **Observación:** Deben utilizar el código desarrollado en la Pregunta 2.

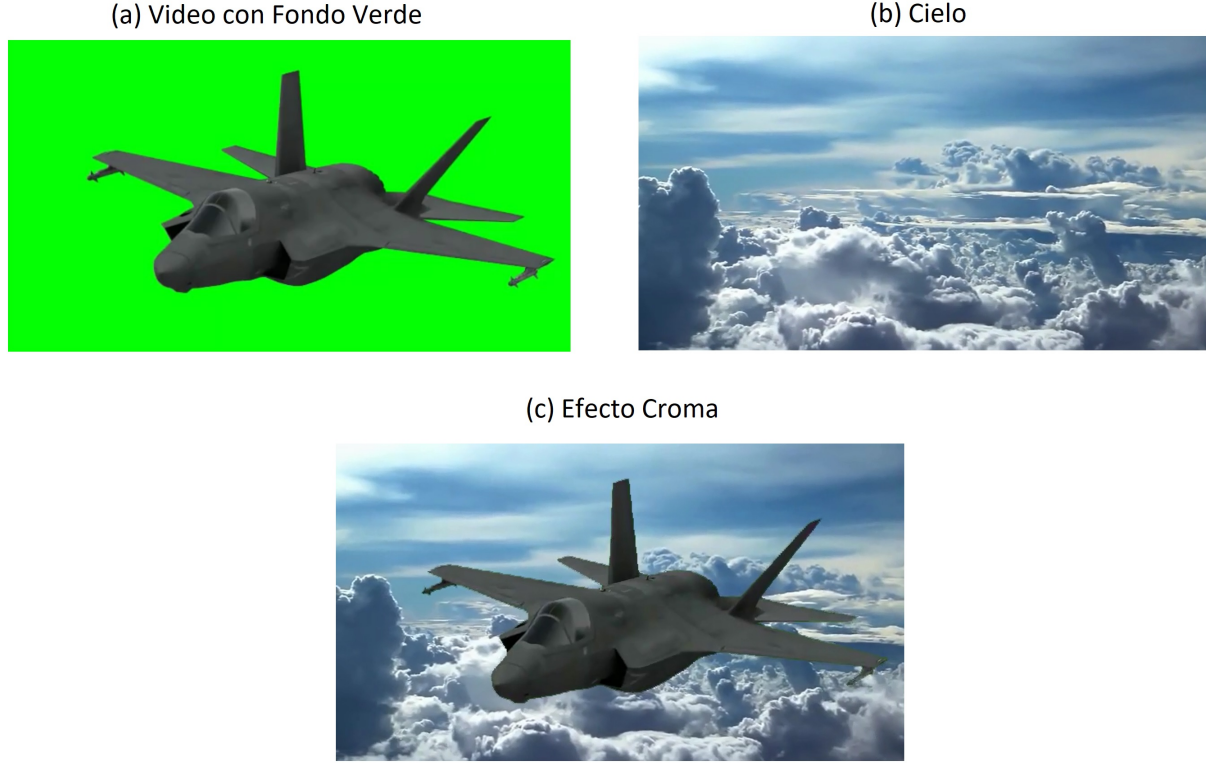


Figura 4: (a) Imagen del Video con Fondo Verde. (b) Imagen del Video con Nuevo Fondo. (c) Imagen del Proceso de Croma.

Parte 2: DFT para Imágenes a Color

- 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 **GNU Octave**.
- 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 4×4 , 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:

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

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 4×4 , 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 4×4 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 4×4 . **Nota:** La matriz J no es única.

Preguntas - Valor: 25 puntos

- **Pregunta 1 [Valor 5 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 20 puntos]:** Implemente computacionalmente la DFT-2D hipercompleja 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 5: (a) Imagen Lena (b) Frecuencia/Espectro de la Imagen Lena

Parte 3: Transformada Hough para Detectar Círculos

- La segunda parte de la tarea consiste en un algoritmo para detectar múltiples círculos en imágenes usando la transformada de Hough. Este algoritmo fue desarrollado en el artículo *On the Improvement of Multiple Circles Detection from Images using Hough Transform* por los investigadores W. O. Barbosa y A. W. Vieira.
- La implementación computacional se debe realizar en **Python**.

Preguntas - Valor: 35 puntos

- **Pregunta 1 [Valor 10 puntos]:** Desarrolle una infografía que explique de manera general el algoritmo propuesto en el artículo *On the Improvement of Multiple Circles Detection from Images using Hough Transform*. Para eso deben considerar la modificación que los autores realizan al Algoritmo 1, el cual se encuentra en la página 337.
- **Pregunta 2 [Valor 25 puntos]:** Implemente computacionalmente la transformada de Hough para detectar border circulares basado en el Algoritmo 1 del artículo *On the Improvement of Multiple Circles Detection*

from Images using Hough Transform. Para eso, deben considerar la modificación a dicho algoritmo que se encuentra en la página 337. Luego, utilice dicho algoritmo para detectar los bordes circulares de las imágenes `imagen1.jpg`, `imagen2.jpg` y `imagen3.jpg` (Ver Figuras 6). La implementación computacional se debe realizar en un archivo con nombre `parte3_p2.py`.



Figura 6: Imágenes de prueba para detectar bordes circulares

Información de la Entrega

- **Fecha y hora límite:** Miércoles 20 de Enero del 2021 a las 1:00 pm.
- Los documentos deben estar guardados usando la siguiente estructura: Una carpeta principal con nombre **Tarea 3**. Dentro de esta carpeta, deben estar dos carpetas con nombres **Parte 1**, **Parte 2** y **Parte 3**. En cada una de estas carpetas estarán todos los archivos necesarios para el desarrollo de las preguntas mencionadas anteriormente.
- Deben enviar la carpeta **Tarea 3** en formato **zip** al correo `jusoto@tec.ac.cr`, con el encabezado **Entrega Tarea 3 - PAID**. En el cuerpo del correo deben indicar el nombre completo de los miembros del grupo.
- **OBSERVACIÓN IMPORTANTE:** La entrega tardía de la tarea se penalizará con una reducción del 20% de la nota final, **POR CADA HORA DE ATRASO**.

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 está presente, entonces el estudiante perderá 35 puntos de la nota final.