

**CC3182 – Visión por Computadora**  
**Laboratorio 5**

## Instrucciones

- Esta es una actividad en grupos de no más de 3 integrantes.
  - Recuerden **unirse al grupo de canvas**
- No se permitirá ni se aceptará cualquier indicio de copia. De presentarse, se procederá según el reglamento correspondiente.
- Tendrán hasta el día indicado en Canvas.
  - No se confíen, aprovechen el tiempo en clase para entender todos los ejercicios y avanzar lo más posible.
- **NOTA:** Limiten el uso de IA generativa. Intenten primero buscar en fuentes de internet y si en verdad necesitan usarla, asegúrense de colcar el prompt que utilizar para cada task donde corresponda, así como una explicación de por qué ese prompt funcionó. A menos que las instrucciones indiquen algo diferente.

## Consideraciones Generales

En el workshop de esta semana, usted implementó un algoritmo de stitching capaz de unir imágenes mediante una Homografía. Aprendió que este método funciona perfectamente bajo la suposición de que la cámara rota sobre su propio eje (sin traslación), eliminando así el efecto de paralaje.

Sin embargo, en la visión por computadora moderna (y en la visión humana), el paralaje no es un error, sino una herramienta.

Durante esta práctica usted forzará el fallo de su algoritmo de panoramas mediante traslación lateral. Este fallo generará "artefactos" o "fantasmas" en la imagen. En lugar de descartar este resultado, usted medirá la separación de estos fantasmas para descubrir empíricamente la relación matemática entre el error de alineación y la distancia física de los objetos.

Reutilizará el código desarrollado en clase con una estricta configuración geométrica para evitar errores de coordenadas negativas recuerde:

1. **Configuración:** Asegúrese de que su código considere la Imagen Izquierda como Train (Destino/Base) y la Imagen Derecha como Query (Origen/A transformar).
2. **Validación:** Antes de salir a capturar datos, verifique que su código funciona con las imágenes de prueba del workshop.

## Task 1

Usted deberá capturar dos sets de imágenes utilizando la cámara de su celular. Para esto deberá crear un set de control y otro experimental. A cada uno deberá ser ejecutado con el código se creó en el workshop. Para cada uno considere:

### Set de Control: Rotación Pura (Validación)

- **Objetivo:** Demostrar que la homografía funciona cuando no hay traslación.
- **Procedimiento:** Desde una posición fija, tome dos fotos de una escena lejana rotando solamente el dispositivo.
- **Resultado Esperado:** Un panorama limpio y continuo.

### Set Experimental: Traslación y Profundidad (El Descubrimiento)

- **Escena:** Busque un entorno (pasillo, habitación o exterior) que tenga objetos a distancias claramente variadas.
  - **Objeto A (Cercano):** A aprox. 50 cm de la cámara (ej. una botella, una silla).
  - **Objeto B (Medio):** A aprox. 2 metros.
  - **Objeto C (Lejano/Fondo):** A más de 5 metros (pared, edificio).
- **Procedimiento:**
  1. Tome la **Foto Izquierda**.
  2. Desplácese lateralmente (dé un paso a la derecha) unos 15-30 cm. Mantenga la cámara apuntando al frente.
  3. Tome la **Foto Derecha**.
  - **Nota:** Asegúrese de que los objetos A, B y C aparezcan en ambas fotos.

## Task 2

Ejecute su algoritmo de stitching con el **Set Experimental**. Dado que hubo traslación, el algoritmo (usando RANSAC) alinearán probablemente el plano dominante (el fondo), pero será incapaz de alinear los objetos cercanos, creando "fantasmas" (duplicados semitransparentes).

### Task 2.1 – Visualización del Error

Solicite a su asistente de IA (ChatGPT/Claude/etc) que genere un script para resaltar estas discrepancias. Se le deja este prompt como sugerencia para su uso inicial:

*“Actúa como ingeniero de visión artificial. Tengo dos imágenes: la imagen base (izquierda) y la imagen warped (derecha transformada). Escribe un código en Python que genere una imagen compuesta (blending) donde la imagen base se muestre en el canal Rojo y la imagen warped en el canal Cian (Verde+Azul). Esto debe crear un efecto de anaglifo donde los píxeles perfectamente alineados se vean en escala de grises y los errores se vean como bordes de colores.”*

### Task 2.1 – Visualización del Error

Usted deberá medir la distancia en píxeles entre las dos versiones del mismo objeto en su imagen fallida.

1. Identifique el **Objeto A** (Cercano) en la imagen compuesta. Busque un punto distintivo (una esquina, una letra) y encuentre sus coordenadas  $(x_1, y_1)$  en la versión roja y  $(x_2, y_2)$  en la versión cian.
2. Identifique el **Objeto B** (Medio) y haga lo mismo.
3. Identifique el **Objeto C** (Fondo) y haga lo mismo.
4. Puede usar herramientas como matplotlib.pyplot.ginput o simplemente inspeccionar la imagen haciendo zoom en su notebook.

Construya una tabla de resultados como la que se muestra a continuación:

Objeto	Distancia Real Estimada	Coordenada X Fantasma 1	Coordenada X Fantasma 2	Disparidad ( $ x_1 - x_2 $ ) en pixeles
A (Cercano)				
B (Medio)				
C (Fondo)				

## Task 3

Construya un reporte, donde responda a las siguientes preguntas basándose **exclusivamente en sus datos**:

1. Observe la columna de "Disparidad". ¿Existe una tendencia clara? ¿Cómo se comporta el valor de la disparidad a medida que aumenta la distancia real del objeto?
2. Si definimos  $Z$  como profundidad y  $d$  como disparidad. Basado en su abla, ¿la relación se parece más a una función lineal ( $d = k \cdot Z$ ) o a una inversa ( $d = k/Z$ )? Grafique sus 3 puntos si es necesario para visualizarlo.
3. Usted acaba de experimentar con los principios de la **Estereoscopía**. Explique por qué el algoritmo de Panoramas (Homografía) falló en alinear los objetos cercanos. ¿Qué información tridimensional estaba "escondida" en ese fallo?

## Evaluación

1. [1.75 pt] Task 1
2. [2 pt] Task 2
3. [1.25 pt] Task 3

Total 5 pts