

CC3182 – Visión por Computadora
Laboratorio 4

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.

Consideraciones Generales

- El lenguaje sugerido para este laboratorio es Python, con las librerías de OpenCV y Numpy
- Se debe entregar tanto Jupyter Notebook (.ipynb) como la versión PDF del mismo, en este debe contener el código, las visualizaciones y principalmente el análisis escrito justificando cada decisión
- Usted es responsable de importar las librerías, leer las imágenes y estructurar su solución

Task 1

El objetivo es evaluar la compresión de la geometría proyectiva y la manipulación algebraica de coordenadas homogéneas. Entonces, responda las siguientes preguntas demostrando el desarrollo matemático. No se aceptan respuestas puramente textuales sin respaldo algebraico.

1. Una homografía H es una matriz de 3×3 . Explique matemáticamente por qué, aunque tiene 9 elementos, solo posee 8 grados de libertad (GDL)
 - a. Adicionalmente, responda. Si tuviéramos una cámara que solo rota sobre su eje óptico (sin traslación ni cambio de perspectiva), ¿la matriz de transformación sigue teniendo 8 GDL o se reduce? Demuestre la estructura de dicha matriz simplificada
1. En el algoritmo DLT (Direct Linear Transform), convertimos el problema $x' = Hx$ en un sistema de la forma $Ah = 0$. Explique por qué buscamos el vector singular asociado al menor valor singular de A en lugar de simplemente invertir la matriz. ¿Qué representa geométricamente ese "menor valor singular" cuando los datos tienen ruido?
1. Si usted selecciona 4 puntos para calcular H , pero 3 de ellos son colineales (están en la misma línea recta), el algoritmo fallará. Explique algebraicamente qué le sucede a la matriz A del sistema DLT en este caso y por qué no tiene solución única.

Task 2

El objetivo de esta parte es implementar el pipeline de alineación sin depender de cajas negras. Por ello considere que **no deben de usar cv2.findHomography o cv2.RANSAC**. Además para este laboratorio necesitará crear su propio dataset, por ello tome 3 fotografías propias de una escena planar (i.e. una pancarta en una pared, un cuadro, o una fachada de edificio lejana) con ángulos y perspectivas drásticamente diferentes. Con esto realice:

1. Detección y Macheo
 - a. Utilice SIFT u ORB (permitido usar OpenCV aquí) para detectar puntos de interés y descriptores.
 - b. Realice un emparejamiento de fuerza bruta (Brute-Force Matcher).
 - c. Requisito: Visualice los matches antes de filtrar. Debe verse una cantidad considerable de ruido/errores.
2. Algoritmo DLT
 - a. Escriba una función `calcular_homografia_dlt(puntos_src, puntos_dst)` que reciba exactamente 4 pares de puntos.
 - b. Debe construir la matriz A de tamaño 8×9 .
 - c. Debe resolver el sistema usando SVD (`numpy.linalg.svd`).
 - d. **Nota:** Debe normalizar los puntos antes de entrar al DLT (restar la media y dividir por la desviación estándar) para estabilidad numérica, y des-normalizar la matriz H resultante.
3. RANSAC Manual
 - a. Implemente la función `ransac_homografia(matches, umbral, prob_exito)`.
 - b. **Cálculo de N:** Su código debe calcular dinámicamente el número de iteraciones N basado en la fórmula de probabilidad vista en clase. No "hardcodee" el número 1000.
 - c. **Bucle:**
 - i. Seleccione 4 matches aleatorios.
 - ii. Llame a su función DLT.
 - iii. Proyecte todos los puntos fuente usando `H_test`.
 - iv. Calcule el error de reproyección (distancia Euclíadiana) y cuente los inliers.
 - a. **Refinamiento:** Una vez encontrado el mejor conjunto de inliers, recalcule H final usando *todos* los inliers (no solo los 4 iniciales) mediante SVD

La salida esperada es una imagen “stitched” (panorama) mostrando la alineación correcta.

Task 3

En esta parte lo que se busca es evaluar el criterio profesional ante situaciones adversas y trade-offs de diseño. Para ello realice y responda lo siguiente

1. Ejecute su algoritmo RANSAC variando el parámetro de umbral de error (threshold) en pixeles (ej. 1px, 5px, 20px)
 - a. Genere una gráfica: Eje X = Umbral, Eje Y = Número de Inliers encontrados.
 - b. Discusión: Como ingeniero, ¿qué riesgo corre si establece un umbral demasiado estricto (ej. 0.5px)? ¿Qué pasa con la matriz final si el umbral es muy laxo (ej. 50px)?
2. Imagine que usted es el Lead Computer Vision Engineer de una empresa de drones. Deben alinear imágenes térmicas de paneles solares tomadas desde el aire para detectar fallos.
 - a. **Problema:** El drone vuela a 50 metros de altura. El terreno no es perfectamente plano (hay colinas suaves), pero los paneles sí son planos.
 - b. **Pregunta A:** ¿Es válido usar una Homografía global para unir todo el mapa del terreno? ¿Por qué sí o por qué no?

- c. **Pregunta B:** Su algoritmo RANSAC está tardando demasiado (3 segundos por frame) en la computadora a bordo del drone (Raspberry Pi). La telemetría indica que el 90% de los matches iniciales son outliers debido al pasto y árboles repetitivos.
- Proponga una estrategia concreta para reducir el tiempo de ejecución sin cambiar el hardware. (Pista: Piense en la fórmula de N o en pre-filtrado geométrico)

Evaluación

- [1.0 pt] Task 1
 - Pregunta 1 (0.4)
 - Pregunta 2 (0.3)
 - Pregunta 3 (0.3)
- [2.5 pt] Task 2
 - Detección y macheo (0.5)
 - Algoritmo DLT (0.8)
 - RANSAC Manual (1.2)
- [1.5 pt] Task 3
 - Pregunta 1 (0.7)
 - Pregunta 2 (0.8)

Total 5 pts