

Lab 4

Integrantes:

- Francis Aguilar,
- César López,
- Jose Marchena, 22398

Repo: https://github.com/MarchMol/lab4_vision

Task 1

El objetivo es evaluar la comprensión 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)

R// Esto es por el valor de escala que se le agrega a las coordenadas al convertirlas de espacio euclidiano a homogéneo. Matemáticamente, está la equivalencia de $x \sim kx$ que nos indica que este valor de escala no afecta/no se toma en cuenta en el caso de proyecciones 3D lo que nos elimina el grado de libertad, o restricciones que implica tomar en cuenta este valor en los cálculos

a. Adicionalmente, responde. 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

R//

Si en lugar de tomar en cuenta 3 posibles direcciones a las cuales rotar, tomamos solo 1 posible movimiento de rotación correspondiente al ángulo sobre el eje óptico, reducimos 3 GDL \rightarrow 1 GDL. Asimismo, si eliminamos también traslación por completo reducimos 3 GDL (correspondiente a las coordenadas x, y, z) a 0, pues no los debemos tomar en cuenta. Lo mismo aplica al cambio de perspectiva, si no aplica a nuestro caso no contamos los GDL

Por tanto pasamos de 3 GDL de rotación, 3 de traslación y 3 de perspectiva (9 valores en H) a 1 GDL correspondiente a la rotación en el eje óptico.

La matriz se vería así, la cual es la matriz 3D estándar para definir rotaciones en espacio normalizado

$$M = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. 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 geoméricamente ese "menor valor singular" cuando los datos tienen ruido?

R// la idea de invertir la matriz A no aplica a nuestro caso porque es una ecuación homogénea. Es decir, estamos igualando a 0 por tanto no hay nada que invertir del otro lado de la ecuación, especialmente, porque estamos buscando soluciones no triviales. También hay otros problemas al pensar en sacar la inversa de la matriz, por ejemplo, A no siempre será cuadrada.

Por último, el menor valor singular de A , geoméricamente, nos indica la dirección del espacio nulo, es decir, donde todos los valores se hacen 0, pero en la realidad, el ruido hace que no podamos solucionar por completo una transformación. No buscamos realmente $Ah = 0$ porque el ruido va a hacer que nunca sea 0, sino que tratamos de MINIMIZAR Ah dada la restricción de que la magnitud de h es 1. Por tanto aproximamos la solución con SVD.

3. 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 necesitara 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 Euclidiana) y cuente los inliers.
 - d. Refinamiento: Una vez encontrado el mejor conjunto de inliers, recalculé 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.

i. 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)