

Laboratorio 1

Repositorio en GitHub: <https://github.com/ArielaMishaanCohen/LAB1.git>

Usted ha sido contratado como Ingeniero de Visión Computacional Junior en "AutonoVision", un startup que desarrolla sistemas de navegación para robots de almacén.

El equipo de hardware ha instalado nuevas cámaras, pero las imágenes llegan con mucho ruido térmico debido a las condiciones de luz del almacén. El módulo actual de detección de obstáculos está fallando: detecta la textura del suelo de concreto como si fueran "bordes" de obstáculos, frenando el robot innecesariamente.

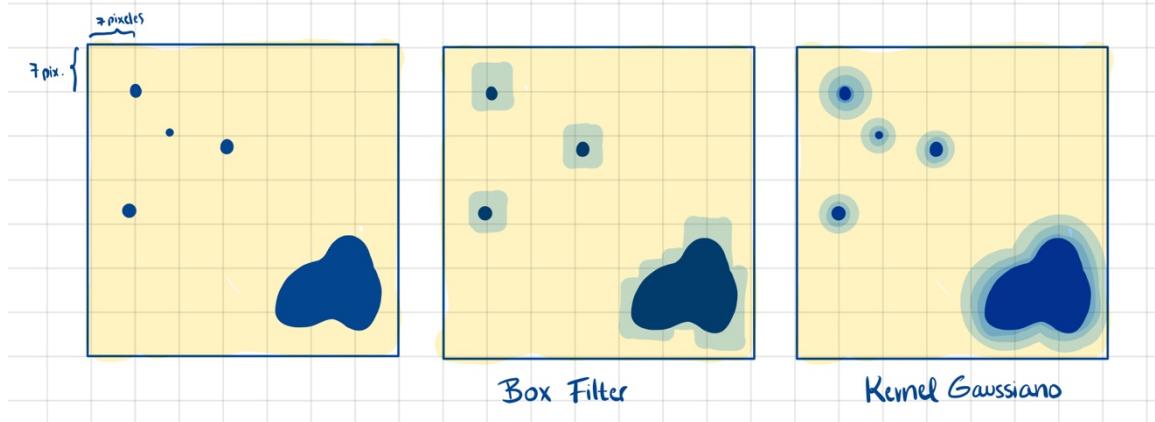
Su director de proyecto le ha asignado la tarea de construir un pipeline de pre-procesamiento robusto desde cero para entender el problema a nivel matemático y ajustar los parámetros óptimos para el despliegue.

Task 1 – Análisis teórico y analítico

1. **Su jefe sugiere usar un filtro de media (Box Filter) de 7x7 para eliminar el ruido rápido. Usted cree que es una mala idea. Explique matemáticamente y con un diagrama visual (dibujado) por qué un Box Filter de ese tamaño es perjudicial para la detección precisa de la posición de un obstáculo comparado con un filtro Gaussiano del mismo tamaño.**

Un box filter de tamaño 7x7 es perjudicial para la detección precisa de la posición de un obstáculo. Esto se debe a que el box filter asigna el mismo peso a todos los pixeles del vecindario. Esto provoca:

- Desenfoque fuerte (se mezclan intensidades de ambos lados del borde)
- Desplazamiento del borde (mala localización del obstáculo)
- Pérdida de información fina.



Los detalles que se perderían con el box filter son muy importantes para la detección de bordes e identificación de objetos.

Como se puede ver en el diagrama, en la imagen modificada con el box filter, todos los píxeles dentro del kernel 7x7 contribuyen con el mismo peso, lo que termina en una difuminación cuadrada que desplaza y ensancha los bordes. Por el otro lado, el filtro Gaussiano asigna más peso a los píxeles centrales y menos a los lejanos. Esto ayuda a identificar de mejor manera el obstáculo (matemáticamente: el filtro Gaussiano aplica una “campana de Gauss” sobre la imagen, dando más importancia a los pixeles del centro, mientras el box filter solo hace un promedio) (Awaits, 2020)¹.

2. Al realizar la convolución en los bordes de la imagen (por ejemplo, en el píxel 0,0), el kernel "se sale" de la imagen.

a. Si el robot navega por pasillos oscuros con luces brillantes al final, ¿por qué el Zero-Padding podría generar falsos positivos de bordes en la periferia de la imagen?

El zero padding consiste en poner 0s en las orillas en las que no se puede utilizar el kernel completo (porque “se sale”). Cuando existen regiones brillantes cercanas al borde, se genera un “borde” entre estos píxeles y la luz de la imagen (0 es representado por el color negro). Al seguir con el proceso y usar el identificador de contornos, como por ejemplo el de Sobel, este identificaría en las orillas contornos que en realidad no son bordes de objetos, sino que fueron artificialmente creados por el padding. Esto genera confusión e identificación incorrecta de bordes.

b. ¿Qué estrategia de padding (Reflect, Replicate, Wrap) recomendaría para evitar esto y por qué?

Para evitar el problema de la confusión del padding con la luz, en vez de llenar los píxeles inalcanzables con 0s, la mejor solución es reflejar los píxeles cercanos. Esto mantiene la intensidad de la región de la imagen (la luz brillante), y no crea regiones artificiales que se pueden malinterpretar como bordes falsos (Vermulapati, 2023).

3. Dada la siguiente sub-imagen I de 3x3 y el kernel K:

$$I = \begin{bmatrix} 10 & 10 & 10 \\ 10 & 0 & 10 \\ 10 & 10 & 10 \end{bmatrix}, \quad K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

¹ Para este ejercicio se utilizó la ayuda de IA generativa, con el objetivo de validar el resultado obtenido. Se le proporcionó la imagen realizada y se dio el siguiente prompt: “Mira mi dibujo. ¿Tiene sentido lo que está ahí? Hice como una difuminación cuadrada para el box y una como uniforme para el gaussiano.” La respuesta del LLM ayudó a aclarar la idea respecto al tema; aclaró que el gaussiano no es “uniforme”, sino “radialmente decreciente”, e indicó que el dibujo estaba correcto.

a. Calcule el valor del píxel central resultante de la convolución

$$\begin{aligned} \text{Pixel central} = & 10 \cdot 0 + 10 \cdot 1 + 10 \cdot 0 + 10 \cdot 1 + 0 \cdot (-4) + 10 \cdot 1 + 10 \cdot 0 + 10 \cdot 1 \\ & + 10 \cdot 0 = 40 \end{aligned}$$

b. ¿Qué tipo de estructura detecta este filtro K (conocido como Laplaciano)?

El filtro laplaciano sirve para resaltar regiones en las que hay un cambio de intensidad rápido en los pixeles (detecta bordes de objetos). Matemáticamente representa la segunda derivada espacial de la imagen. El kernel asigna un peso negativo al pixel central y pesos positivos (menores) a sus pixeles vecinos inmediatos, permitiendo detectar discontinuidades locales de intensidad. Este cambio de signo entre le centro y los pixeles adyacentes es lo que permite identificar bordes y transiciones dramáticas (Lini, 2023)².

Task 2 – Práctica

Ver código en el github.

Task 3 – Evaluación de ingeniería y criterios

Experimento A – Análisis

Al analizar las tres imágenes de bordes obtenidas, se observa que cuando no se aplica suavizado previo, el operador Sobel detecta fuertemente la textura del suelo y el ruido presente en la imagen, generando numerosos falsos bordes que podrían ser interpretados erróneamente como obstáculos. En contraste, al utilizar un suavizado Gaussiano con un valor de σ alto, los bordes finos y detalles pequeños como grietas del suelo se atenúan o desaparecen debido a la eliminación de altas frecuencias, permaneciendo únicamente los bordes asociados a estructuras grandes. Desde un punto de vista de ingeniería, el uso de un Gaussiano con σ elevado resulta la mejor opción para detectar pallets grandes en un entorno industrial, ya que reduce significativamente el ruido y la textura del suelo, mejorando la robustez del sistema de detección y evitando falsas detecciones que afectarían la navegación del robot.

Experimento B – Preguntas

Al aplicar un umbral simple a la magnitud del gradiente, se observa que las líneas de los bordes tienden a romperse, especialmente en zonas donde la intensidad del gradiente disminuye ligeramente. Esto ocurre porque un único valor de umbral no puede adaptarse a las variaciones

² En este ejercicio se usó ChatGPT para verificar la respuesta con el prompt “Mira mi respuesta del 3 y dime si está bien: [respuesta]”. La respuesta que se había puesto estaba correcta, solo se hizo un cambio en la formulación. La respuesta dada reemplazó a: “Como se ve en la matriz, el filtro le da al pixel central el mayor valor y los cuatro pixeles adyacentes con menor intensidad, para identificar si se trata de un área con mucho contraste o no (esa es la razón del cambio de signo entre el pixel del centro y sus 4 pixeles vecinos)”.

Alina Carías, Daniel Machic y Ariela Mishaan

Visión por Computadora

locales de iluminación y contraste a lo largo de un mismo borde, provocando discontinuidades visibles. Como resultado, los bordes pierden continuidad, lo que dificulta la correcta identificación de objetos completos en la imagen.

Un umbral de corte simple nunca alcanza la efectividad del método de histéresis utilizado en el algoritmo de Canny porque emplea un único valor de decisión que no considera la continuidad ni la estructura de los bordes. En escenarios reales, como un robot en movimiento, las vibraciones y los cambios leves de iluminación provocan variaciones locales en la magnitud del gradiente a lo largo de un mismo borde, haciendo que algunas secciones caigan por debajo del umbral y se pierdan. La histéresis resuelve este problema al utilizar dos umbrales y un criterio de conectividad, permitiendo que los bordes débiles se conserven únicamente si están conectados a bordes fuertes, lo que garantiza continuidad espacial y una detección de obstáculos más robusta y estable durante la navegación del robot.

Referencias

Awaits. (11 de noviembre de 2020). [CV] 2. *Image Processing Basic: Gaussian and Median Filter, Separable 2D filter.* Obtenido de Medium: <https://medium.com/jun94-devpblog/cv-2-gaussian-and-median-filter-separable-2d-filter-2d11ee022c66>