

PRACTICA 2 DETECCION DE BORDES

Visión computacional Aplicada a la Robótica

Kelly Del Moral Adame

Introducción

La presente práctica se redacta con la finalidad de demostrar el funcionamiento de un filtro Canny. Se programó un filtro Canny por un algoritmo multietapa desarrollado por Jonh F. Canny en 1986, con la finalidad de comprender cada paso del algoritmo y así poder modificar los parámetros de este. Se realizaron pruebas con el filtro en una escena y se comprobó el funcionamiento de la detección de bordes del filtro. Se pretende usar el filtro programado posteriormente para la detección de diferentes características de una escena.

Objetivo

Desarrollar y comprender el algoritmo del filtro Canny para detectar los bordes de un vídeo.

Marco teórico

Un detector de bordes es un operador o filtro encargado de encontrar las regiones de una imagen digital en las cuales esta cambia su brillo drásticamente (o sea, sus bordes). Dado que tiene aplicaciones de tiempo real, es necesario que los algoritmos que lo implementan sean eficientes.

El detector o algoritmo de Canny es un operador que usa un algoritmo multietapa para detectar bordes en una imagen, desarrollado por John F. Canny en 1986. También fue el fundador de la teoría computacional de la detección de bordes para explicar por qué su algoritmo funciona.

Según esta teoría, los puntos más importantes que debe tener un detector de bordes para ser correcto son:

- Detección con bajos niveles de errores. Es decir, el algoritmo debe detectar con precisión tantos bordes en la imagen como sea posible.
- Los puntos de borde detectados deben estar localizado en el centro del borde. Cada borde debe ser detectado una y solo una vez, y de ser posible, el ruido no debe crear falsos bordes.

El algoritmo de Canny posee 4 etapas, donde la primera toma la imagen original, y se van pasando los resultados de cada etapa como entrada de la siguiente, hasta la última que genera el resultado final.

- 1.Reducción de ruido
2. Gradiente de Intensidad
3. Supresión de No-Máximos
4. Doble Umbral [1]

Desarrollo

1.Reducción de ruido

Primero se realizó la reducción de ruido, por medio del filtro gaussiano. Este primer paso se hace debido a que el ruido en una imagen puede ocasionar la detección de falsos bordes, por eso, es necesario aplicar un filtro para reducir el ruido como primera etapa. En la fig1 se puede observar la imagen original y en la fig 2 se observa el resultado del suavizado.

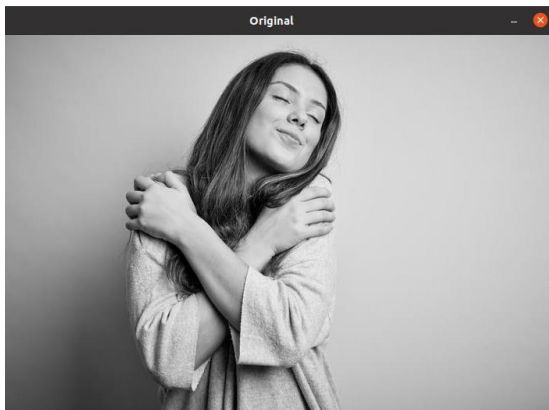


Fig 1. Imagen original

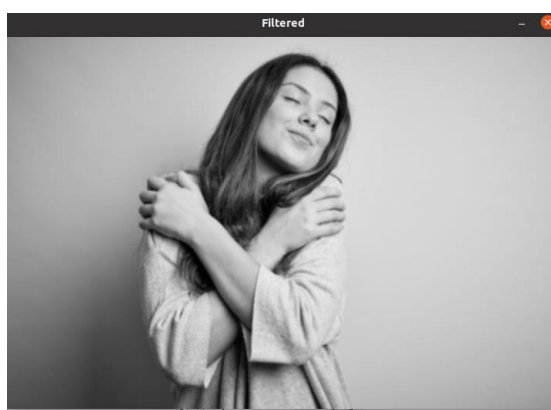


Fig 2. Imagen filtrada

Una prueba del funcionamiento del suavizado de la imagen por medio del filtro gaussiano se puede observar en la fig. 3 y fig. 4.

En las cuales se muestra el resultado final de aplicar un filtro Canny, pero en la imagen de la izquierda el filtro que se le aplicó a la imagen original es de tamaño menor al de la derecha es decir la imagen izquierda tiene más ruido como consecuencias se pueden observar más bordes falsos.



Fig 3 . Imagen sin suavizado



Fig 4 . Imagen con suavizado

2. Gradiente de intensidad

Una vez filtrada la imagen por medio del filtro gaussiano se obtiene el gradiente de intensidad, aplica un filtro sobel en ambas direcciones vertical y horizontal. Una vez obtenidos los gradientes vertical y horizontal se aplicaron las siguientes fórmulas para la obtención de la magnitud del gradiente(G_m) y la dirección del ángulo (G_a) .

$$G_{m_{i,j}} = \sqrt{G_{x_{i,j}}^2 + G_{y_{i,j}}^2}$$
$$G_{a_{i,j}} = \text{atan2}(G_{y_{i,j}}, G_{x_{i,j}})$$

Fig 5. Ecuaciones para obtención de gradiente y ángulo

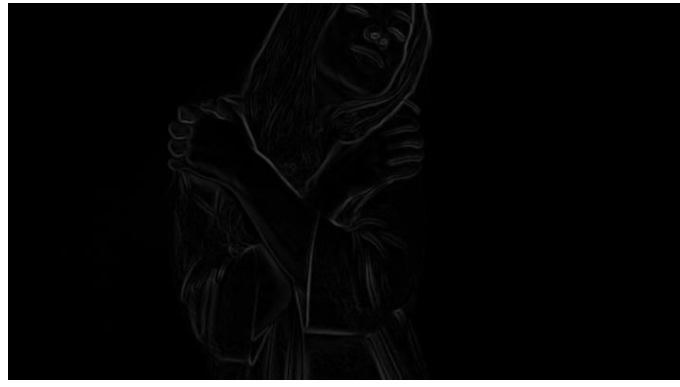


Fig 6. Resultado del gradiente de la imagen

3. Supresión de no máximos

Este paso es utilizada para afinar los bordes encontrados en el paso anterior. Consiste en escanear la imagen para eliminar los píxeles que no formen parte de los bordes. Para esto se compara el valor de cada píxel con sus vecinos cercanos en la dirección del gradiente (perpendicular al borde). Si el valor del píxel en cuestión es mayor que sus píxeles vecinos, entonces este es considerado un máximo local y el algoritmo lo acepta. De lo contrario, si el píxel resulta no ser un máximo local, entonces es suprimido. El resultado de la supresión se puede observar en la fig. 7.

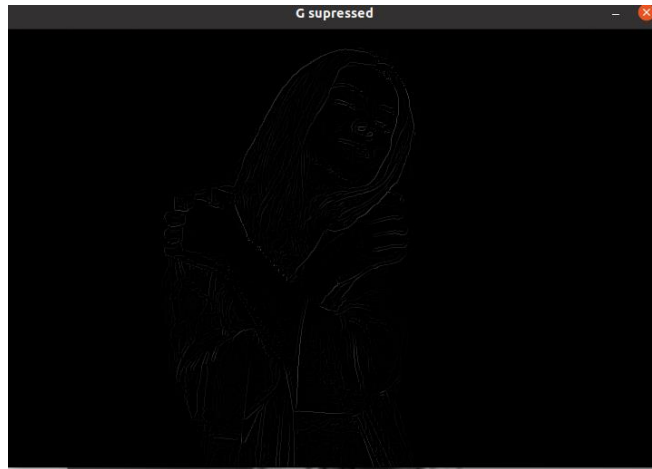


Fig 7. Resultado de supresión de no máximos

4. Doble umbral

Como ultimo paso se seleccionaron dos umbrales, uno máximo y otro mínimo con base a los gradientes obtenidos. Una vez elegidos los umbrales se procedió a clasificar los pixeles en tres tipos:

Fuertes: $\text{Pixel} \geq \text{umbral máximo}$

Débiles: $\text{Pixel} \geq \text{umbral mínimo}$ $\text{Pixel} \leq \text{umbral máximo}$

Suprimidos: $\text{Pixel} \leq \text{umbral mínimo}$

Los pixeles que se encontraban por debajo del umbral mínimo eran suprimidos como su nombre lo dice.

Los fuertes se mantenían en la imagen final

La permanencia de los débiles dependía de los pixeles vecinos, si alguno de los pixeles vecinos era un pixel fuerte entonces el pixel débil permanecía en la imagen de lo contrario se suprimía.

Una prueba importante de este paso fue la selección de los umbrales, ya que entre más pequeño fuera el valor máximo, aparecían más bordes innecesarios y entre más grande era desaparecían bordes de la imagen, en la imagen de la izquierda podemos observar el primer caso y en la derecha el ultimo.

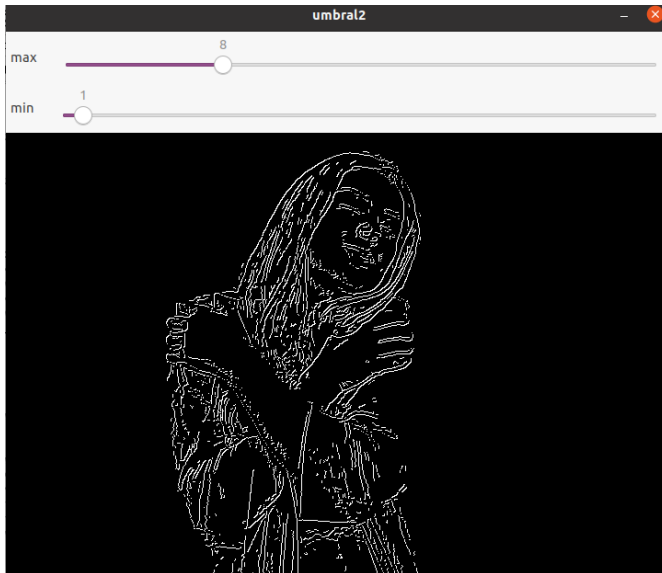


Fig 8. Resultado de escoger un umbral máximo alejado al valor máximo del gradiente



Fig 9. Resultado de escoger un umbral máximo acercado al valor máximo del gradiente



Fig 9. Resultado final

Conclusiones

El filtro Canny es muy efectivo detección de bordes finos. Es importante realizar un suavizado o quitar ruido a nuestra imagen o video para evitar bordes falsos. También la detección de umbrales es importante ya que de esto depende la cantidad de bordes que el filtro Canny detectara por ejemplo si se escoge un umbral demasiado bajo al valor máximo de la magnitud del gradiente nos

puede generar bordes falsos, así que dependiendo de las características que se busquen extraer se pueden definir los umbrales del filtro.

Referencias

[1] Juan Martin Leiva. optimización de un detector Canny. Organización del Computador II Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires.2017

[2] Universidad de Jaén. Detección de bordes en una imagen. Área de Ingeniería de Sistemas y Automática. Curso 2005/2006

[3] Yingke Feng, Jinmin Zhang, and Siming Wang , "A new edge detection algorithm based on Canny idea", AIP Conference Proceedings 1890, 040011 (2017) <https://doi.org/10.1063/1.5005213>