Aplicación del método de Canny para la detección de bordes de una imagen digital

Walter Alejandro Moreno Ramírez
Departamento de Estudios Multidisciplinarios
Universidad de Guanajuato
Yuriria, Guanajuato
Correo: wa.morenoramirez@ugto.mx

***Abstract*—This article describes what are the edges of an image, as they can be detected by Canny method. Its advantages and applications will also be stated.**

***Index Terms*—Pixel, píxeles, convolución, canny, smoothing, suavizado, gradiente, segmentación ,primera derivada, función, C++, OpenCV, ventana, máscara, vecindario.**

I. Introducción

Existen varios algoritmos para detectar los bordes de los objetos en una imagen digital, entre ellos, las matrices de convolución son una muy buena técnica para la detección de bordes; donde se puede utilizar una matriz de convolución ya sea de primera o segunda derivada. Ya que dichas matrices se basan en la razón de cambio o un cambio muy drástico en los tonos de grises en los vecinos de un pixel central.

Para esta práctica se utiliza el algoritmo de Canny para la detección de bordes. Fue desarrollado por John F. Canny en 1986 y se basa en una secuencia de técnicas que se aplican sobre una imagen. Este algoritmo tiene como objetivo principal satisfacer tres criterios principales:

- **Baja tasa de error**: esto se refiere a una buena detección de los bordes existentes.
- **Buena localización**: la distancia entre los pixeles de borde detectados y los pixeles de borde reales es reducida.
- **Respuesta mínima**: sólo un detector responde por borde.

Podemos visualizar dicho algoritmo como una cadena, donde el resultado de cada técnica pasa al siguiente eslabón, teniendo como resultado final una imagen con los bordes de los objetos presentes en la imagen de entrada.

El algoritmo de Canny para la detección de bordes se basa en cuatro técnicas o componentes:

- **Suavizado**: la finalidad de suavizar la imagen es la de atenuar el ruido presente en la imagen. Esto se logra ya sea con una matriz de mediana, promedio o con una matriz gaussiana.
- **Gradiente y angulos**: una vez la imagen fue suavizada, se procede a obtener el gradiente y el ángulo para cada pixel de la imagen.
- **Supresión de no-máximos**: se eliminan los píxeles cuyo gradiente sea de menor longitud con respecto al de sus vecinos que se encuentren en la dirección positiva y negativa del ángulo central de una ventana de 3x3.
- **Histéresis**: este último paso ubica los píxeles en tres secciones delimitadas por dos umbrales. Estas secciones categorizan a los píxeles como “débiles”, “medios” o “fuertes” y lo que hace es formar una continuidad entre los píxeles “fuertes” ya que representan los bordes de la imagen.

II. Metodología

Suavizado de la imagen:

Como primer paso del algoritmo se suaviza la imagen. Debido a que los bordes de una imagen pueden ser fácilmente confundidos por ruido, al suavizar la imagen se desea atenuar el ruido para poder prevenir la detección de falsos bordes.
Para realizar este paso se utilizó un filtro gaussiano de 5x5 con $\sigma = 1,5$, lo que da como resultado la matriz mostrada en la Ecuación (1).

$$B = \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \end{bmatrix} \tag{1}$$

Una vez se definio el filtro para el suavizado y, dada una imagen A de entrada, se realiza la convolución entre dicho filtro y la imagen.

$$imageOut = A * B \tag{2}$$

De acuerdo a la Ecuación (2), nos muestra que el resultado de la convolución es una imagen que se pasa al siguiente

eslabón de la cadena que es el algoritmo de Canny.

Obtener bordes:

La siguiente fase del algoritmo es la detección de los bordes de la imagen, para ello se utilizó un kernel, mas específicamente, los kernel de Sobel para detectar la razón de cambio tanto en el eje x como en el eje y de la imagen. Por lo anterior, los kernel o filtros de Sobel son de primera derivada, dichos kernel se muestran en la Ecuación (3).

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{3}$$

Para obtener los bordes es necesario realizar una convolución entre cada kernel con la imagen suavizada, lo que da como resultado el gradiente de la imagen en ambas direcciones, x y y.
Para unificar estos bordes y, de acuerdo a la Ecuación (4), se obtiene la magnitud del gradiente de la imagen.

$$G = \sqrt{G_x^2 + G_y^2} \tag{4}$$

También se obtiene el ángulo de acuerdo a la Ecuación (5).

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{5}$$

Ya que el resultado de la función $atan2$, que esta incluida en la librería $cmath$ para C++, arroja un resultado en radianes, es necesario hacer una conversión de radianes a grados multiplicando por 180 y dividiendo entre π. Después se discretiza el ángulo resultante para cada pixel, en un multiplo de $\pi/4$. Para dicha discretización los ángulos se tomarán de acuerdo a la Ecuación (6).

$$\theta(p) = \begin{cases} 0° \leftarrow 157{,}5 \leq p \leq 22{,}5 \\ 45° \leftarrow 22{,}5 \leq p < 67{,}5 \\ 90° \leftarrow 67{,}5 \leq p < 112{,}5 \\ 135° \leftarrow 112{,}5 \leq p < 157{,}5 \end{cases} \tag{6}$$

Supresión de no-máximos:

Una vez se hizo la discretización de las direcciones del gradiente, el siguiente paso del algoritmo es eliminar los píxeles cuya magnitud, con respecto a sus vecinos, es menor. Para esto, se toma el pixel central de una ventana de 3x3 y se verifica que la magnitud de su gradiente sea mayor que el mismo valor pero de los píxeles que se encuentren en la dirección positiva y negativa de acuerdo a la dirección del gradiente del mismo pixel central, si esta condición se cumple, el pixel central se considera como un borde, de lo contrario se elimina.

La evaluación con los píxeles vecinos se realiza de acuerdo a las direcciones mostradas en la Figura 1.

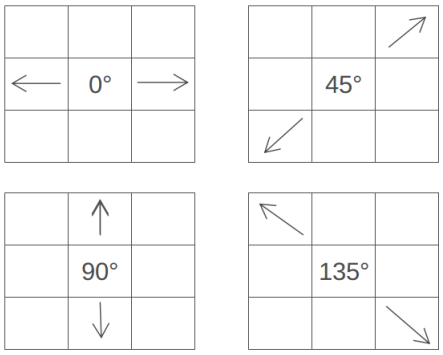

Figura 1: Direcciones para realiza la supresión de las magnitudes no-máximas, esto para eliminar píxeles que no son bordes.

Una característica muy importante de este procedimiento es que como resultado se obtendrán unos bordes con un grosor de un pixel, lo que da una muy buena localización de los bordes obtenidos, con respecto a los bordes reales.

Una vez que se eliminaron los píxeles que no pueden considerarse bordes teniendo como parámetro de discriminación la magnitud de su gradiente, se procede al último paso.

Histéresis:

Hasta el momento se tienen los bordes de una imagen con una muy buena localización, pero aún existen bordes provenientes del ruido que es necesario eliminar y existen un conjunto de píxeles no contiguos, los cuales pueden ser parte de un borde. Para ello es necesario realizar un seguimiento de dichos píxeles para formar los bordes.

Al aplicar la histéresis es necesario utilizar una doble umbralización y, de acuerdo a los dos umbrales, se agrupan los píxeles en tres categorías:

- **Fuertes**: son aquellos píxeles cuyo valor es mayor o igual al umbral alto.
- **Medios**: se considera a aquellos píxeles cuyo valor esta entre el umbra alto y el umbra bajo.
- **Débiles**: son todos los píxeles cuyo valor es menor o igual al umbra bajo.

Tomando como referencia la gráfica de la Figura 2., T_1 y T_2 son los umbrales bajo y alto respecticamente, a partir de estos dos umbrales comienza la categorización de los píxeles.

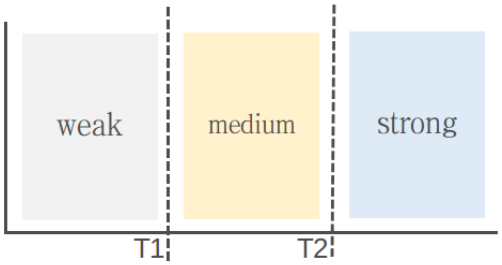

Figura 2: Gráfica de la doble umbralización y como deben separarse los píxeles dependiendo de sus valores.

Una vez se tiene una categorización de los píxeles, el siguiente paso es eliminar aquellos píxeles que sean “débiles”, mantener los que son “fuertes” y, en el caso de que sean píxeles “medios”, se realiza el siguiente procedimiento:

Utilizando una ventana de 3x3 donde el pixel central es un pixel medio, se analiza si tiene al menos un vecino que sea “fuerte”, para estos existen dos casos.

- Se cuenta con al menos un pixel “fuerte” en el vecindaro: el pixel “medio” se vuelve “fuerte”.
- No se tienen vecinos “fuertes”: el pixel “medio” se elimina o se vuelve cero.

Para este paso es muy importante realizar los cambios en la misma imagen sobre la cual se está haciendo el análisis y categorización de los píxeles, esto para producir un seguimiento de los píxeles que formarán los bordes.

III. Resultados

Para poder llevar a cabo las pruebas fue necesario elegir una imagen que tuviera una buena cantidad de bordes; los bordes se pueden encontrar en casas, ventanas, carros y objetos que estén conformados por figuras bien definidas como podrían ser círculos, cuadrados, rectangulos, triángulos, etc.
La Figura 3., es una muestra de las imágenes que se utilizó para realizar las pruebas.

Figura 3: Foto de un SuDoKu, la cuale se utilizará para probar la implementación del algoritmo de Canny.

El primer paso de la implementación es hacer un suavizado en la imagen, el resultado de esta técnica se muestra en la Figura 4.

Figura 4: Difuminado o suavizado que es el resultado de aplicar la Ecuación (2).

El paso anterior es simplemente para atenuar el ruido presente en la imagen ya que este se puede confundir como un bordes por el detector. Como se atenua y no se elimina completamente, pueden pasar píxeles de ruido en el siguiente paso del algoritmo, para disminuirlo o eliminarlo se aplican otras técnicas.

Como siguiente paso se procede a obtener los bordes de la imagen, aplicando los kernel de convolución de la Ecuación (3), como resultado tenemos la imagen de la Figura 5.

Figura 5: Se obtienen el gradiente de la imagen con un filtro o kernel de primera derivada, las derivadas se realizan tanto en x como en y.

En base al resultado se puede observar que los bordes mas dominantes se encuentran con un tono de gris mayor, que los bordes que no son dominantes. Este resultado es a causa de utilizar un filtro de derivada. Si, se obtienen los

bordes analizando como $[-1, 1]$, como resultado tendríamos unos bordes más delgados lo cual es algo bueno, aunque el siguiente paso sería también realiza lo mismo.

Una vez que se obtienen los bordes, el siguiente paso es la supresión de los no-máximos mostrado en la Figura 6.

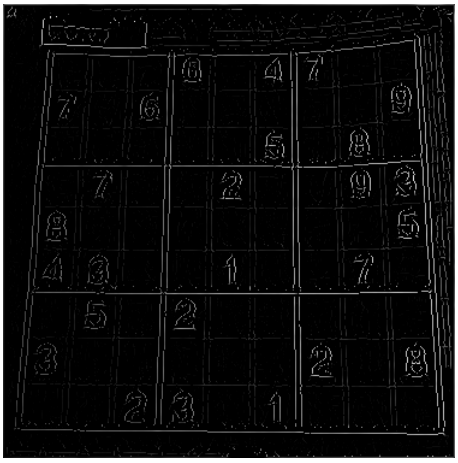

Figura 6: Con la supresión de los no-máximos podemos eliminar los bordes que no tienen relevancia en la imagen y un poco de ruido.

Como resultado, los bordes se adelgazan a un grosor de un pixel, esto también provoca que se diferencie aún más entre los bordes con una mayor relevancia en la imagen que aquellos cuyor valor hace parecer como bordes "débiles".

El último paso es aplicar la histéresis sobre la imagen resultante de la supresión de los píxeles no-máximos. Para realiza la histéresis, se utilizan una doble umbralización y, de acuerdo al valor de cada pixel y la sección en la que se categoricen de acuerdo a la grafica de la Figura 2., se eliminan aquellos píxeles "débiles" y los píxeles "medios" que no tengan un pixel "fuerte" como vecino, esto a su vez, realiza un seguimiento para poder completar las lineas o bordes teniendo como resultado la imagen mostrada en la Figura 7.

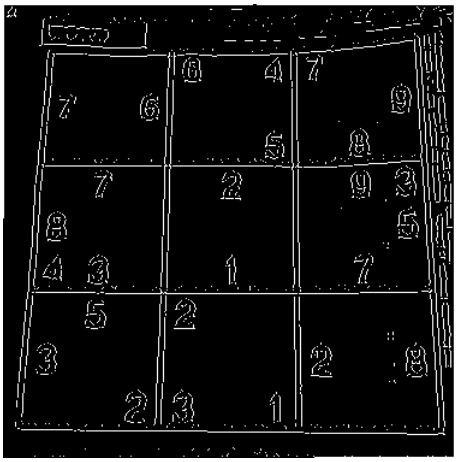

Figura 7: Resultado de aplicar la histéresis sobre la imagen.

Cabe mencionar que la doble umbralización se utiliza con un umbral alto y un umbral bajo. El valor del umbral bajo se tomó como $t_1 = 20$ y el umbral alto se calcula a partir del valor del umbral bajo, utilizando un ratio de $r = 2$ o $r = 3$, y se calcula como $highThreshold = lowThreshold * r$.

IV. Conclusiones

El algoritmo de Canny es uno o posiblemente el mejor algoritmo para detectar los bordes de una imagen. Cada técnica o paso del algoritmo tiene un fundamento matemático del porque se aplica y cual es el resultado esperado teórico, lo cual le agrega una robustez al algoritmo. Otra característica importante es que es un algoritmo permite modificar parámetros en cada paso, esto da la oportunidad de agregar una matriz de convolución distinta en valores y dimensiones. También la posibilidad de modificar tanto el umbral bajo, ratio y, por ende, el umbral alto para adecuarlo a la imagen y sus propiedades.