

Detección de bordes mediante el algoritmo de Canny

Jorge Valverde Rebaza

Escuela Académico Profesional de Informática

Universidad Nacional de Trujillo

Resumen

En el presente trabajo se expone de manera concisa el fundamento básico para detectar los bordes de una imagen siguiendo el algoritmo de Canny, el cual es considerado como uno de los mejores para este tipo de trabajo. Además se presentan algunos resultados obtenidos al aplicar este algoritmo en imágenes reales.

1. Introducción

En el área de procesamiento de imágenes, la detección de los bordes de una imagen es de suma importancia y utilidad, pues facilita muchas tareas, entre ellas, el reconocimiento de objetos, la segmentación de regiones, entre otras.

Se han desarrollado variedad de algoritmos que ayudan a solucionar este inconveniente. El algoritmo de Canny es usado para detectar todos los bordes existentes en una imagen. Este algoritmo está considerado como uno de los mejores métodos de detección de contornos mediante el empleo de máscaras de convolución y basado en la primera derivada. Los puntos de contorno son como zonas de píxeles en las que existe un cambio brusco de nivel de gris. En el tratamiento de imágenes, se trabaja con píxeles, y en un ambiente discreto, es así que en el algoritmo de Canny se utiliza máscaras, las cuales representan aproximaciones en diferencias finitas.

El resto de este trabajo está organizado de la siguiente manera: En la sección 2 se muestran los criterios en los que se basa el algoritmo de Canny. El algoritmo de Canny es expuesto en la sección 3. En la sección 4 se presentan algunos resultados obtenidos al aplicar el algoritmo de Canny. Finalmente en la sección 5 se presentan las conclusiones.

2. Criterios del Algoritmo de Canny

En 1986, Canny propuso un método para la detección de bordes, el cual se basaba en tres criterios, estos son:

- Un criterio de **detección** expresa el hecho de evitar la eliminación de bordes importantes y no suministrar falsos bordes.
- El criterio de **localización** establece que la distancia entre la posición real y la localizada del borde se debe minimizar.
- El criterio de **una respuesta** que integre las respuestas múltiples correspondientes a un único borde.

3. Algoritmo de Canny para la detección de bordes

Uno de los métodos relacionados con la detección de bordes es el uso de la primera derivada, la

que es usada por que toma el valor de cero en todas las regiones donde no varía la intensidad y tiene un valor constante en toda la transición de intensidad. Por tanto un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada [1], característica que es usada para detectar un borde, y en la que se basa el algoritmo de Canny.

El algoritmo de Canny consiste en tres grandes pasos:

- Obtención del gradiente: en este paso se calcula la magnitud y orientación del vector gradiente en cada píxel.
- Supresión no máxima: en este paso se logra el adelgazamiento del ancho de los bordes, obtenidos con el gradiente, hasta lograr bordes de un píxel de ancho.
- Histéresis de umbral: en este paso se aplica una función de histéresis basada en dos umbrales; con este proceso se pretende reducir la posibilidad de aparición de contornos falsos.

3.1 Obtención del gradiente

Para la obtención del gradiente, lo primero que se realiza es la aplicación de un filtro gaussiano a la imagen original con el objetivo de suavizar la imagen y tratar de eliminar el posible ruido existente. Sin embargo, se debe de tener cuidado de no realizar un suavizado excesivo, pues se podrían perder detalles de la imagen y provocar un pésimo resultado final. Este suavizado se obtiene promediando los valores de intensidad de los píxeles en el entorno de vecindad con una mascara de convolución de media cero y desviación estándar σ . En la figura 1 se muestran dos ejemplos de máscaras que se pueden usar para realizar el filtrado gaussiano.

Una vez que se suaviza la imagen, para cada píxel se obtiene la magnitud y módulo (orientación) del gradiente, obteniendo así dos imágenes. El algoritmo para este primer paso se describe a continuación.

(a)

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

(b)

$$\frac{1}{115}$$

2	4	5	4	2
4	9	12	9	4
5	12	15	12	5
4	9	12	9	4
2	4	5	4	2

Figura 1. Máscaras de convolución recomendadas para el obtener el filtro gaussiano. La máscara (a) fue obtenida de [5], mientras que la máscara (b) fue obtenida de [6].

3.2 Supresión no máxima al resultado del gradiente

Las dos imágenes generadas en el paso anterior sirven de entrada para generar una imagen con los bordes adelgazados. El procedimiento es el siguiente: se consideran cuatro direcciones identificadas por las orientaciones de 0° , 45° , 90° y 135° con respecto al eje horizontal. Para cada píxel se encuentra la dirección que mejor se aproxime a la dirección del ángulo de gradiente.

Algoritmo: Obtención de Gradiente

Entrada: imagen I

máscara de convolución H , con media cero y desviación estándar σ .

Salida: imagen E_m de la magnitud del gradiente

imagen E_o de la orientación del gradiente

1. Suavizar la imagen I con H mediante un filtro gaussiano y obtener J como imagen de salida.
2. Para cada píxel (i, j) en J , obtener la magnitud y orientación del gradiente basándose en las siguientes expresiones:

El gradiente de una imagen $f(x,y)$ en un punto (x,y) se define como un vector bidimensional dado por la ecuación:

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}$$

siendo un vector perpendicular al borde, donde el vector G apunta en la dirección de variación máxima de f en el punto (x,y) por unidad de distancia, con la magnitud y dirección dadas por:

$$|G| = \sqrt{G_x^2 + G_y^2} = |G_x| + |G_y|,$$

$$\phi(x, y) = \tan^{-1} \frac{G_y}{G_x}$$

3. Obtener E_m a partir de la magnitud de gradiente y E_o a partir de la orientación, de acuerdo a las expresiones anteriores.

Posteriormente se observa si el valor de la magnitud de gradiente es más pequeño que al menos uno de sus dos vecinos en la dirección del ángulo obtenida en el paso anterior. De ser así se asigna el valor 0 a dicho píxel, en caso contrario se asigna el valor que tenga la magnitud del gradiente.

La salida de este segundo paso es la imagen I_n con los bordes adelgazados, es decir, $E_m(i, j)$ después de la supresión no máxima de puntos de borde.

3.3 Histéresis de umbral a la supresión no máxima

La imagen obtenida en el paso anterior suele contener máximos locales creados por el ruido. Una solución para eliminar dicho ruido es la histéresis del umbral.

El proceso consiste en tomar la imagen obtenida del paso anterior, tomar la orientación de los puntos de borde de la imagen y tomar dos umbrales, el primero más pequeño que el segundo. Para cada punto de la imagen se debe localizar el siguiente punto de borde no explorado que sea mayor al segundo umbral. A partir de dicho punto seguir las cadenas de máximos locales

conectados en ambas direcciones perpendiculares a la normal del borde siempre que sean mayores al primer umbral. Así se marcan todos los puntos explorados y se almacena la lista de todos los puntos en el contorno conectado. Es así como en este paso se logra eliminar las uniones en forma de Y de los segmentos que confluyen en un punto

Algoritmo: Supresión no máxima

Entrada: imagen E_m de la magnitud del gradiente
imagen E_o de la orientación del gradiente

Salida: imagen I_n

Considerar: cuatro direcciones d_1, d_2, d_3, d_4 identificadas por las direcciones de $0^\circ, 45^\circ, 90^\circ$ y 135° con respecto al eje horizontal

1. Para cada píxel (i, j) :
 - 1.1. Encontrar la dirección d_k que mejor se aproxima a la dirección $E_o(i, j)$, que viene a ser la perpendicular al borde.
 - 1.2. Si $E_m(i, j)$ es más pequeño que al menos uno de sus dos vecinos en la dirección d_k , al píxel (i, j) de I_n se le asigna el valor 0, $I_n(i, j) = 0$ (supresión), de otro modo $I_n(i, j) = E_m(i, j)$.
2. Devolver I_n

3.4 Un cuarto paso

Frecuentemente, es común que un cuarto y último paso se realice en el algoritmo de Canny, este paso consiste en cerrar los contornos que pudiesen haber quedado abiertos por problemas de ruido.

Un método muy utilizado es el algoritmo de Deriche y Cocquerez. Este algoritmo utiliza como entrada una imagen binarizada de contornos de un píxel de ancho. El algoritmo busca los extremos de los contornos abiertos y sigue la dirección del máximo gradiente hasta cerrarlos con otro extremo abierto.

El procedimiento consiste en buscar para cada píxel uno de los ocho patrones posibles que delimitan la continuación del contorno en tres direcciones posibles. Esto se logra con la convolución de cada píxel con una máscara específica. Cuando alguno de los tres puntos es ya un píxel de borde se entiende que el borde se ha cerrado, de lo contrario se elige el píxel con el valor máximo de gradiente y se marca como nuevo píxel de borde y se aplica nuevamente la convolución. Estos pasos se repiten para todo extremo abierto hasta encontrar su cierre o hasta llegar a cierto número de iteraciones determinado.

Algoritmo: Histéresis de umbral a la supresión no máxima

Entrada: imagen I_n obtenida del paso anterior
imagen E_o de la orientación del gradiente
umbral t_1
umbral t_2 , donde $t_1 < t_2$

Salida: imagen G con los bordes conectados de contornos

1. Para todos los puntos de I_n y explorando I_n en orden fijo:
 - 1.1. Localizar el siguiente punto de borde no explorado previamente, $I_n(i, j)$, tal que $I_n(i, j) > t_2$
 - 1.2. Comenzar a partir de $I_n(i, j)$, seguir las cadenas de máximos locales conectados en ambas direcciones perpendiculares a la normal de borde, siempre que $I_n > t_1$.
 - 1.3. Marcar todos los puntos explorados y, salvar la lista de todos los puntos en el entorno conectado encontrado.
2. Devolver G formada por el conjunto de bordes conectados de contornos de la imagen, así como la magnitud y orientación, describiendo las propiedades de los puntos de borde.

4. Resultados

A continuación se presentan algunos resultados obtenidos de la literatura y otros de la experiencia propia.

El primer ejemplo de los resultados de aplicar el algoritmo de Canny se puede observar en la figura 2, obtenida de [3]. El segundo ejemplo fue obtenido de [5] y se muestra en la figura 3.

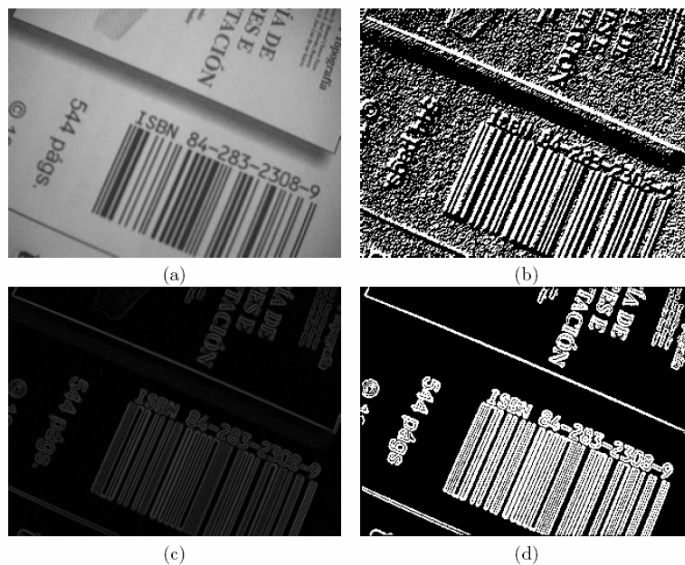


Figura 2. Resultado de aplicar el detector de bordes de Canny: (a) imagen original; (b) orientación; (c) supresión no máxima; (d) histéresis de umbral.

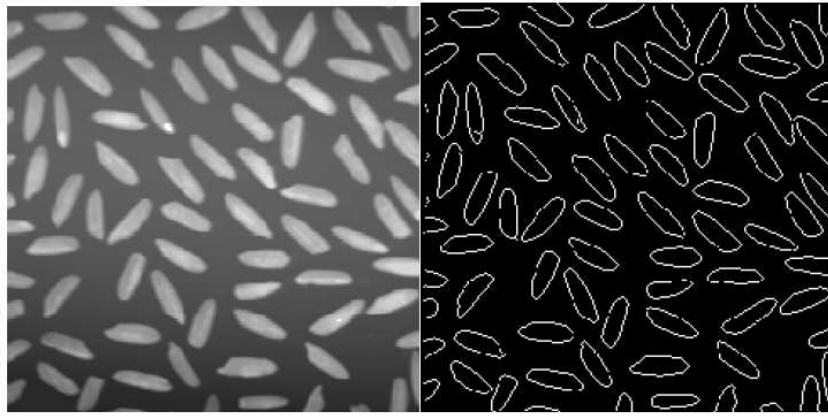


Figura 3. Resultado de la detección de bordes mediante el algoritmo de Canny.

Siguiendo los pasos que he descrito en las secciones anteriores, personalmente implemente el algoritmo de detección de bordes de Canny, en la figura 4 y figura 5 muestro los resultados obtenidos.

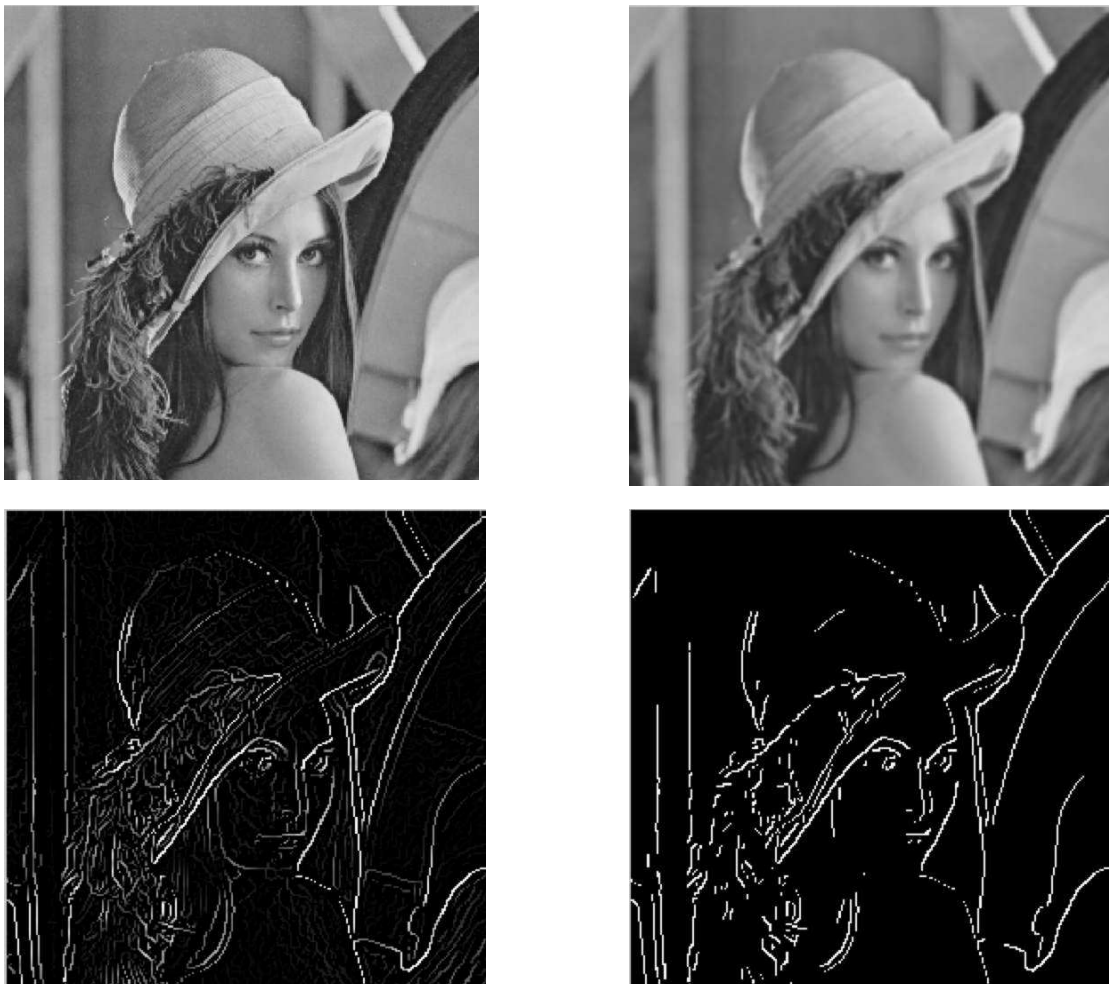


Figura 4. La figura superior izquierda muestra la imagen original de Lena. La figura superior derecha muestra la imagen de Lena después de haber pasado un filtro gaussiano en el que se usó la mascarilla presentada en la figura 1a. La figura inferior izquierda muestra la imagen de Lena después de haber pasado por la supresión de no máximos. La figura inferior derecha muestra la imagen de Lena después del proceso de histeresis, y en los que se usó umbrales de 100 y 130.



Figura 5. Se muestran otras dos figuras en las que se aplicó el algoritmo de detección de bordes de Canny con umbrales diferentes al aplicar la histéresis. En la figura izquierda se usó umbrales con los valores 110 y 150. En la figura de la derecha se usó umbrales con los valores 120 y 180.

5. Conclusiones

En este trabajo se ha presentado un estudio informativo en el que se detalla todos los pasos necesarios para detectar los bordes de una imagen mediante el algoritmo de Canny, el cual tiene como principal ventaja su gran adaptabilidad para poder ser aplicado a diversos tipos de imágenes, además de no disminuir su performance ante la presencia de ruido en la imagen original. Sin embargo, algunas de las desventajas que pude identificar al implementar este algoritmo se encuentran en el suavizamiento, puesto que, si aumentamos el σ de la mascarilla logramos reducir el ruido pero difuminamos los bordes y perdemos calidad al momento de calcular la orientación.

Además, como se ha podido observar, los resultados obtenidos en la aplicación de este algoritmo fueron muy satisfactorios, dado que, como se presento en la sección de Resultados, el algoritmo de Canny funcionó correctamente al implementarla en imágenes reales, algunas de las cuales no eran de buena definición (por ejemplo, la imagen presentada en la figura 3).

De esta manera concluyó el presente trabajo indicando que el algoritmo de Canny es uno de los mejores métodos para la detección de bordes, el cual aplica métodos de diferencias finitas basado en la primera derivada y cuya popularidad se debe, además de sus buenos resultados, a su sencillez, la cual que permite una gran velocidad de procesamiento al ser implementado. Sin embargo, se debe tener en cuenta que existen muchos algoritmos de detección de bordes, los cuales no deben de ser descartados sin antes observar sus resultados sobre las imágenes que queremos procesar.

Referencias

- [1] Gonzalo Pajares, Jesús M. de la Cruz. Visión por Computador. Imágenes digitales y aplicaciones. Ra-Ma, 2001.
- [2]. Canny. A Computational Approach to Edge Detection, IEEE Trans. on Pattern Analysis and Machine Intelligence, 8(6), pp. 679-698 (1986).

- [3] Vicente Castelló Martínez. Localización y decodificación de códigos de barras en imágenes digitales. Universitat Jaume I, 2005.
- [4] Alfonso Pérez, Hugo Solís. Métodos para seguimiento de dedo en tiempo real. 2005.
- [5] F. Escolano, O. Colomina, M.A.Cazorla. Visión Artificial: Extracción de Características I, 2006.
- [6] Hill Green. Canny Edge Detection Tutorial. 2002.
- [7] Alonso Ramírez Manzanares. Reporte de Búsqueda, detección y conteo de objetos. Centro de Investigación en Matemáticas. México. 2002.
- [8] Miguel A. Jaramillo, J. Alvarado Fernández, E. Martínez de Salazar. Implementación del detector de Bordes de Canny sobre Redes Neuronales Celulares. Departamento de Electrónica e Ingeniería Electromecánica – Universidad de Extremadura. España.