DETECCIÓN DE BORDES DE UNA IMAGEN USANDO MATLAB

Víctor Manuel Sánchez Zorrilla
Instituto Tecnológico de Oaxaca
vic.zorrilla.san@gmail.com

Franco Gabriel Caballero Julián
Instituto Tecnológico de Oaxaca
francogcaballero@gmail.com

Miguel Ángel Pérez Solano
Instituto Tecnológico de Oaxaca
mapsolano@gmail.com

Martín Vidal Reyes
Instituto Tecnológico de Oaxaca
martinvid@gmail.com

Enrique Rodríguez Calvo
Instituto Tecnológico de Oaxaca
enrique.rodriguez@itoaxaca.edu.mx

Resumen

El procesamiento digital es una técnica que se emplea en imágenes para la extracción de parámetros por medio de técnicas de detección de bordes; en su desarrollo convergen conocimientos propios de diferentes áreas como las matemáticas, la computación y el diseño de algoritmos (u operadores); de gran importancia en la manipulación de imágenes por la diversidad de aplicaciones en el campo de la robótica y visión artificial.

El operador o algoritmo de Canny es uno de los mejores métodos para detección de bordes que existen en la actualidad, haciendo que los contornos abiertos los tome como contornos cerrados, generando una mayor definición de

reconocimiento de objetos que serían útiles en aplicaciones de robótica (detección, reconociendo, seguimiento o localización).En este trabajo se desarrollan aplicaciones de detección de bordes en dos versiones utilizando el software Matlab; primero por medio del operador de Sobel y segundo por medio del operador de Canny.

En el desarrollo de las dos aplicaciones el programa se codifica de la siguiente manera: lee la imagen original en formato jpg en color con un tamaño de 725x313 pixeles, detecta y reconoce sus elementos básicos como figuras, líneas y formas, visualiza sus partes principales, define geométricamente la posición de sus elementos y por ultimo compara la primera imagen con la segunda imagen (teniendo en cuenta que se usan dos imágenes con las mismas características, pero en diferente posición).

Palabras Claves: Algoritmo de Canny, bordes, imagen.

Abstract

The digital processing is a technique used in imaging for edge detection in order to improve their quality; in its application converge own knowledge of different areas like mathematics, computing and design of algorithms (or operators); with great importance in image manipulation by the diversity of applications in the field of robotics and artificial vision.

The operator or Canny edge detector is one of the best methods for detecting edges that exist today, making open contours take them as closed contours, generating a greater definition of recognition of objects that would be useful in robotics applications (detection recognizing, monitoring or positioning).In this paper edge detection applications are developed in two versions using the Matlab software; first by Sobel operator and second by Canny operator.

In the development of the two applications the program is coded as follows: Read the original image in jpg format color with a size of 725x313 pixels, detects and recognizes their basic elements such as shapes, lines and shapes, displays its main parts, geometrically defines the position of its elements and finally compares

the first image with the second image (considering that two images are used with the same characteristics but in different position).

Keywords: *Canny algorithm, border, image.*

1. Introducción

El procesamiento digital de imágenes es un campo de investigación abierto. El constante progreso en esta área no ha sido por sí mismo, sino en conjunto con otras áreas con las cuales está relacionada como las matemáticas, la computación, y el conocimiento cada vez mejor de ciertos órganos del cuerpo humano que intervienen en la percepción y en la manipulación de las imágenes. Aunado a esto, la inquietud del hombre por imitar y usar ciertas características del ser humano como apoyo en la solución de problemas. El avance del Procesamiento Digital de Imágenes se ve reflejado en la medicina, la astronomía, geología, microscopía, etc. La información meteorológica, la transmisión y despliegue agilizado de imágenes por Internet tienen sustento gracias a estos avances (Escalante, 2006).

Los bordes extraídos de imágenes de segunda dimensión a partir de escenas de tercera dimensión se pueden clasificar según el punto de vista en: dependientes o independientes. Un borde independiente del punto de vista refleja propiedades inherentes de objetos tridimensionales, tales como marcas y formas en las superficies de los mismos. Un borde dependiente del punto de vista puede cambiar si se cambia el punto de vista, y refleja la geometría de la escena, tales como objetos que se tapan unos a otros (Rodríguez y Sossa, 2012).

Un borde típico puede ser la arista entre un bloque rojo y uno amarillo que se encuentran uno a continuación del otro. En contraste, una recta puede ser un pequeño número de pixeles de un color diferente a un fondo que nunca cambia. Para la recta, usualmente se detectan dos bordes, uno a cada lado de la misma (González y Woods, 2011). Para lo cual se han desarrollado variedad de algoritmos que ayudan a solucionar la detección de bordes. Algunos de estos algoritmos (u operadores) son:

- Operador de Roberts. El operador obtiene buena respuesta ante bordes diagonales, figura 1. Ofrece buenas prestaciones en cuanto a localización. El gran inconveniente de este operador es su extremada sensibilidad al ruido y por tanto tiene pobres cualidades de detección (Hermosilla et al., 2006).

0	1
-1	0

Gy

1	0
0	-1

Gx

Figura 1 Máscara del operador de Roberts.

- Operador de Prewitt. El operador Prewitt, al igual que el de Sobel, expande la definición del gradiente en una máscara de 3x3 para ser más inmune al ruido utilizando la misma ecuación que Sobel, pero con una constante c=1. A diferencia de Sobel, este operador no otorga una importancia especial a los pixeles cercanos al centro de la máscara. Se puede apreciar la máscara del operador Prewitt en la figura 2 (Muñoz, 2007).

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

Figura 2 Máscara del operador de Prewitt.

- Operador de Sobel. El operador de Sobel es la magnitud del gradiente calculado mediante la ecuación 1.

$$M = \sqrt{sx^2 + sy^2} \quad (1)$$

Donde las derivadas parciales son calculadas por ecuaciones 2.

$$s_x = (a_2 + c\,a_3 + a_4) - (a_0 + c\,a_7 + a_6)$$
$$s_y = (a_6 + c\,a_5 + a_4) - (a_0 + c\,a_1 + a_2) \quad (2)$$

Con la constante c=2.

En el caso de las etiquetas de pixeles de los operadores de Sobel y de Prewitt trabajan en forma diagonal dando una mejor detección como se muestra en la figura 3.

a_0	a_1	a_2
a_7	a_8	a_3
a_6	a_5	a_4

Figura 3 Matriz de la máscara.

De igual manera que el operador Roberts, Sx y Sy pueden implementarse usando mascaras de convolución. Este operador pone una especie de énfasis en pixeles cercanos al centro de la máscara. Es uno de los operadores comúnmente utilizados en la detección de bordes, se muestra en la figura 4 (Muñoz, 2007).

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Figura 4 Máscara del operador de Sobel.

- Operador Frei-Chen (Isotrópico). El operador de Prewitt proporciona una buena detección para bordes verticales y horizontales, mientras que el operador de Sobel es mejor en la detección de bordes diagonales. El operador isotrópico intenta llegar a un equilibrio entre ellos aplicando las máscaras de la figura 5.

-1	0	1
$-\sqrt{2}$	0	$\sqrt{2}$
-1	0	1

-1	$-\sqrt{2}$	-1
0	0	0
1	$\sqrt{2}$	1

Figura 5 Máscara del operador Frei-Chen (Isotrópico).

- Operador LoG. Uno de los métodos más utilizados es la suavización por medios de una Gaussiana. Que consiste en:
  - ✓ Convolucionar la imagen original con un filtro gaussiano.

- ✓ Calcular las derivadas sobre la imagen visualizada.

Como este operador es lineal podemos cambiar la operación de distintas formas:

- ✓ Suavizado de la imagen y cálculo de la segunda derivada.
- ✓ Convolucion de la imagen original utilizando el Laplaciano del gaussiano (operador LoG).

El operador LoG es también sensible al ruido, pero los efectos de ruido pueden ser reducidos si se ignoran los cruces por cero producidos por pequeños cambios en la intensidad de la imagen. Este operador nos da la información de la dirección de los ejes, determinada mediante la dirección del cruce por cero (Hertle y Zisserman, 2003).

- Operador Laplaciano. Todos los operadores anteriores tienen una aproximación hacia derivadas de primer orden sobre el valor de píxeles en una imagen.

  Existen métodos que utilizan detectores de borde basados en derivadas de segundo orden. Uno de los más utilizados es el operador Laplaciano mostrados en la ecuación 3.

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \qquad (3)$$

  Aunque el operador Laplaciano responde a la transmisión de intensidad, rara vez se utiliza en la práctica para detección de bordes (Hartley y Zisserman, 2003):

  - ✓ Los operadores basados en la primera derivada son sensibles al ruido de la imagen. El operador Laplaciano es aún más sensible.
  - ✓ Genera bordes dobles.
  - ✓ No existe información directa de los ejes detectados.

- Operador de Canny. Uno de los métodos más importantes para realizar una detección global de bordes sobre una imagen es el conocido como método de Canny. Esta técnica, que se caracteriza por estar optimizada para la detección de bordes diferenciales, consta de tres etapas principales:

filtrado, decisión inicial, e histéresis. La primera etapa consiste en un filtrado de convolución de la derivada primera de una función gaussiana normalizada discreta sobre la imagen, realizada en dos direcciones: horizontal y vertical. La función gaussiana posee dos parámetros fundamentales, valor medio y desviación típica. En este caso, el valor medio es nulo, siendo la ecuación que define el filtro gaussiano de la forma como se aprecia en la ecuación 4 (Rodriguez y Sossa, 2012).

$$g(x) = k\, exp\left(-\frac{x^2}{2\sigma^2}\right) \qquad (4)$$

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 (Valverde, 2010).

2. Método

El programa consta de varias partes que son de vital y suma importancia para el resultado satisfactorio:

- Leer la imagen de referencia: Por medio de google imágenes pudimos obtener nuestra imagen de referencia la cual es un billete de dólar que cuenta con distintas características: en formas, tamaño y bordes, etc. En la figura 6 se muestra la imagen inicial de este artículo.
- Detectar y reconocer elementos básicos de la imagen: Una vez que fue ingresada la imagen al programa en Matlab, sigue el reconocimiento de elementos básicos o principales de dicha imagen tanto de la imagen original

como la imagen del operador o algoritmo de Canny (detectando color, dimensión, posición, etc.), como se muestra en las figuras 7 y 8.

Figura 6 Imagen de un billete de cien dólares.

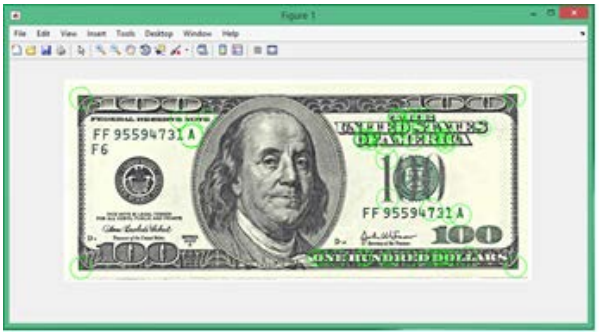
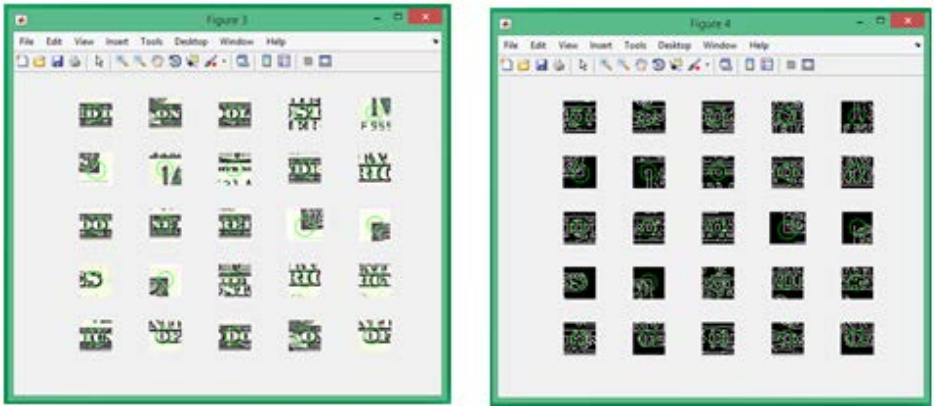


Figura 7 Detección de características de la imagen.

Figura 8 Detección de características usando el algoritmo de Canny.

- Visualizar 25 fragmentos principales de la imagen: Ya que fueron detectados y reconocidos los elementos básicos de la imagen tanto es su versión original como en el algoritmo de Canny, el programa visualizará la imagen en 25 fragmentos, en los cuales se muestra la forma y posición de esos fragmentos, como se muestra en la figura 9.

- Lee una segunda imagen (la cual será comparada con la primera imagen): Para este siguiente paso que es el reconocimiento de la segunda imagen, de igual forma que el primer punto tendremos una segunda imagen de referencia la cual fue obtenida de google imágenes (figura 10), la cual será muy útil para la comparación con la primera imagen, la cual nos ayudará a detectar su similitud.



Fragmentos de imagen original. Fragmentos del algoritmo canny.

Figura 9 Veinticinco fragmentos de la imagen.

Figura 10 Imagen de varios billetes de cien dólares.

- Muestra la segunda imagen y hace la detección de bordes en la imagen: De igual forma que se hace con la primera imagen se hace para la segunda, se detectará los elementos de la imagen como la forma y su contorno (figura 11), en donde se detectan y muestran dichas áreas tanto en la imagen original como en la del algoritmo.
- Detecta las características principales de la imagen: Se hace una primera prueba donde compara las imágenes tanto la primera como la segunda, en esta primera comparación para la detección, no es muy sensible porque detecta otras áreas que tienes muy poco de similitud. En la figura 12, se

hace la comparación con la imagen original (original-original) y con la imagen del algoritmo (Canny-Canny), para la figura 13 se hace una combinación de imágenes original-Canny y Canny-original, esto se hace con la finalidad de obtener una mejor y mayor eficiencia en detección de bordes.

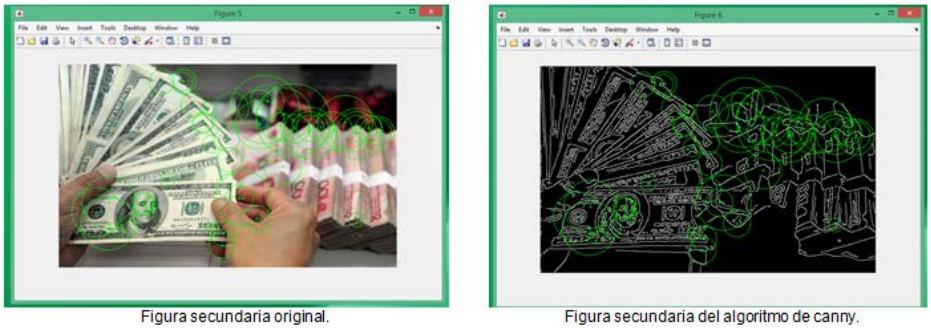
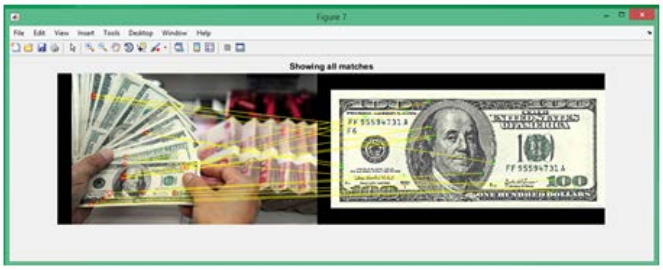
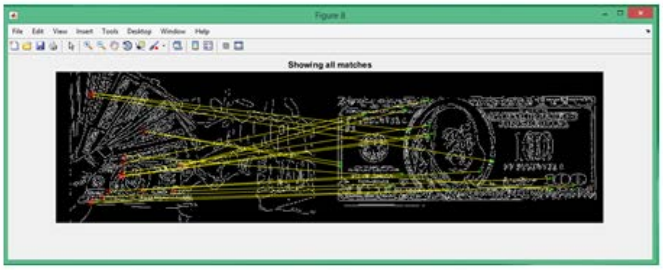


Figura secundaria original. Figura secundaria del algoritmo de canny.

Figura 11 Detección de características de la segunda imagen.



Primera comparación de imágenes (original-original).

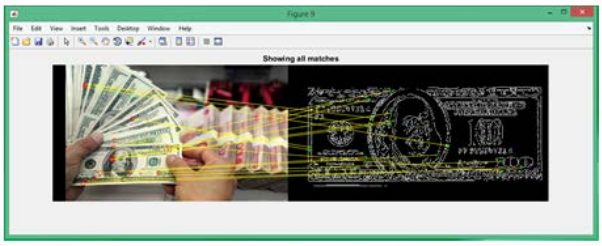


Primera comparación de imágenes (canny-canny).

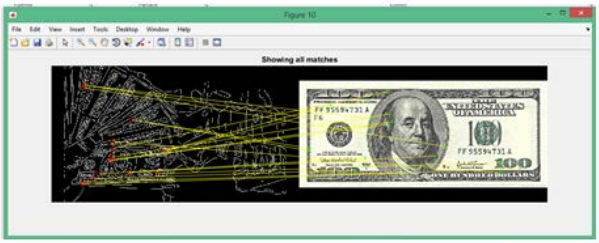
Figura 12 Comparación de imágenes original-original y Canny-Canny.

- Define geométricamente la posición de semejanza de ambas imágenes: En este siguiente punto del programa la comparación entre ambas imágenes ya es más precisa detectando solamente las áreas que cuentan con la misma similitud. Observe la figura 14 donde la segunda comparación se

hace entre las imágenes original-original, como también se hace la de Canny a Canny. En figura 15 se tiene otra operación donde se comparan pero entre original a Canny y de Canny a original, esto se hace para tener una mayor y confiable área de similitud.

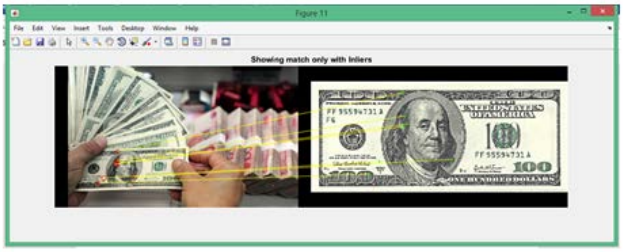


Primera comparación de imágenes (original-canny).

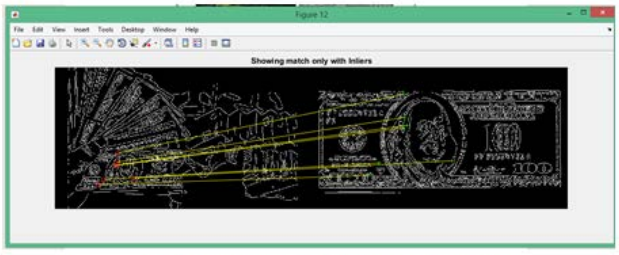


Primera comparación de imágenes (canny-original).

Figura 13 Comparación de imágenes original-Canny y Canny-original.



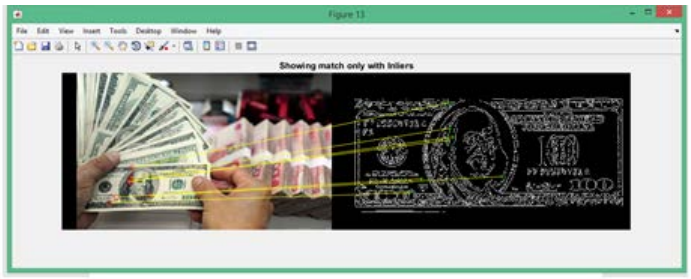
Segunda comparación de imágenes (original-original).



Segunda comparación de imágenes (canny-canny).

Figura 14 Segunda Comparación de imágenes original-original y Canny-Canny.

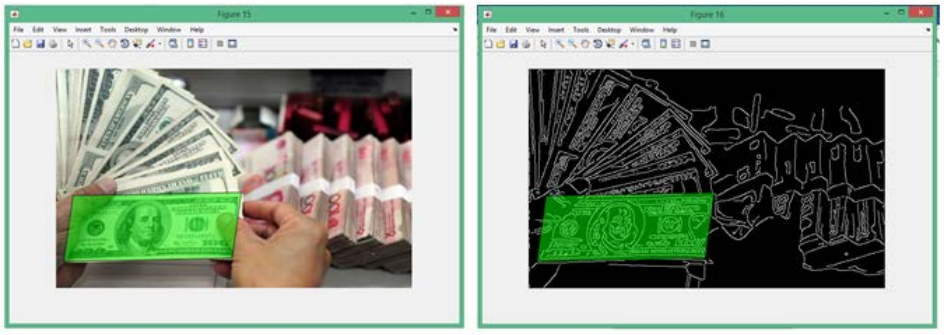
- Se comparan las imágenes y se muestra la zona de la similitud: Una vez obtenido lo anterior llegamos a nuestro último paso del programa para el cual es la zona de similitud, donde la segunda imagen mostrará el área donde es igual a la primera imagen, mostrando en la imagen inferior de la figura 16 un recuadro de color verde el cual nos indica que esa es el área de similitud en la imagen original y en la imagen del algoritmo de Canny.



Segunda comparación de imágenes (original-canny).

Segunda comparación de imágenes (canny-original).

Figura 15 Segunda comparación original-Canny y Canny-original.



Resultado de la zona de similitud. Resultado de la zona de similitud con el algoritmo canny.

Figura 16 Zona de similitud.

Una vez obtenido todos estos pasos podemos llegar a nuestro resultado de obtener el algoritmo y comparación exacta de la imagen.

3. Resultados

El operador de Sobel es usado en procesamiento de imágenes, especialmente en detección de bordes. Este operador formula de forma conjunta máscaras de convolución. El problema que presenta el operador de Sobel es que los bordes de la imagen no son definidos en su totalidad, lo que tiene el operador es que solo reconoce los bordes primarios de la imagen tal como se observa en la figura 17.

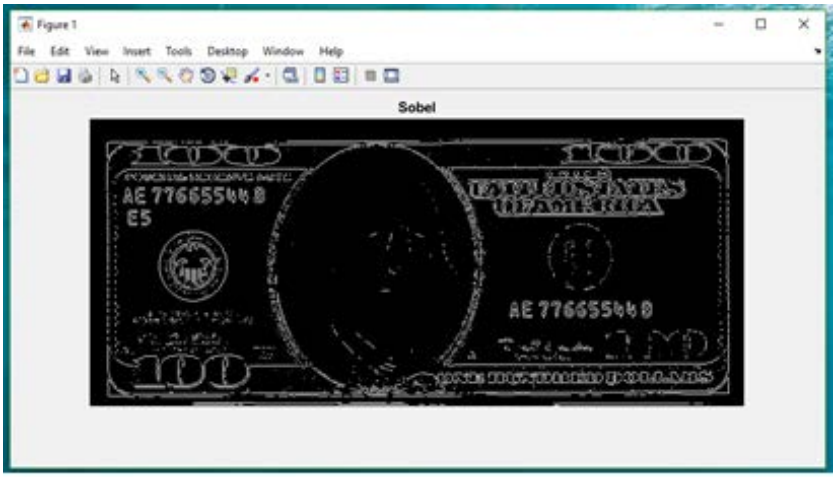


Figura 17 Aplicación del operador de Sobel.

En comparación con la imagen original (figura 6) se muestra claramente como los bordes del operador de Sobel no son los suficientemente buenos para el reconocimiento de ciertas partes de la imagen.

El operador de Canny es uno de los principales y mejores algoritmos que existe en detección de bordes, detectando un mayor número de bordes en la imagen que da una mejor definición y apreciación de características de la imagen.

A diferencia del operador de Sobel, el algoritmo de Canny es más eficiente y preciso en la detección de bordes ya que cubre y rellena los espacios vacíos de la imagen cerrando su contorno. Como se aprecia en la figura 18, en esta imagen podemos ver más detección de bordes haciendo una imagen de mejor calidad.

Viendo en la imagen original (figura 6) podemos observar claramente como es definido con mayor precisión en sus bordes, siendo así que el operador de Canny es de mejor definición y detalla más contornos.

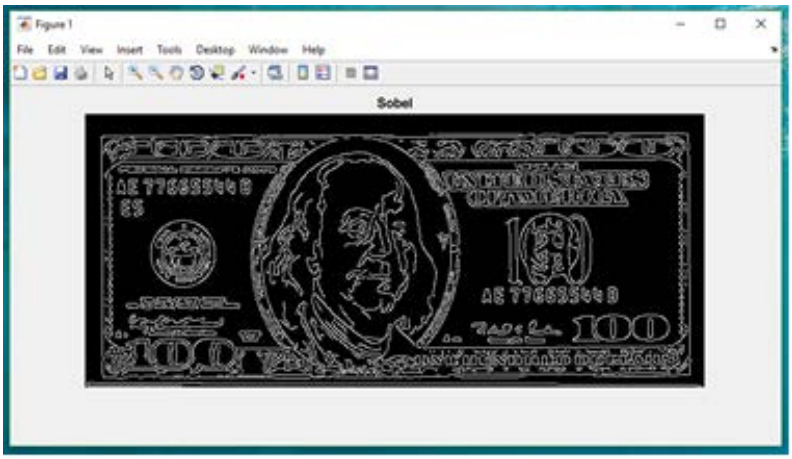


Figura 18 Aplicación del operador de Canny.

El trabajo solamente consiste en obtener resultados visuales desde la observación del analista humano, para hacer el comparativo entre los operadores de Sobel y Canny; un trabajo numérico más complejo está fuera del alcance de este estudio.

4. Discusión

A lo largo del trabajo se expone y se mencionan las características con las que cuenta el algoritmo de Canny, el cual tiene un gran impacto en el procesamiento digital de imágenes para el reconocimiento y la detección de bordes que nos ayuda a detectar los bordes de una imagen ya sea números, rasgos, características o como el ejemplo que se ha expuesto a lo largo del trabajo que es capaz de detectar los bordes de un billete de cien dólares.

Una de las aplicaciones que podría tener en robótica e inteligencia artificial, es que sea capaz de reconocer objetos, las características con las que cuanta, determinando similitudes, la detección de formas y figuras.

5. Bibliografía y Referencias

[1] Escalante, B. (2006). Procesamiento Digital de Imágenes. http://verona.fi-p.unam.mx/boris/teachingnotes/Introduccion.pdf.

[2] Gonzalez, R., y Woods, R. (2011). Digital Image Processing Using MATLAB. Second edition. Boston MA: Addison_Wesley.

[3] Ghanshyam, R, y Vipin,T. (2016).Texture image retrieval using adaptive tetrolet transforms. .University of engineering and Technology Guna, India. Digital Signal Processing 48(2016) 50-57 Elseiver.

[4] Hermosilla, T., Bermejo, E., Balaguer, A., y Ruiz L.A. (2006). Detección de bordes con precisión subpíxel en imágenes digitales: Interpolación lineal frente a esquemas de tipo no lineal.

[5] Hartley, R., & Zisserman, A. (2003). Multiple View Geometry in computer vision. Second edition. Combridge University Press.

[6] Jaramillo, M.A., Fernández, J.A, Martínez, E. (s. f.). Implementación del detector de bordes de Canny sobre redes neuronales celulares. http://eii.unex.es/profesores/jalvarof/pdf/canny00.pdf.

[7] Muñoz, J. (2007). Descriptaciones de características. http://www.uma.es /munozlp/pi_cap8.pdf.

[8] Rodriguez, R., y Sossa, J. (2012). Procesamiento y Análisis Digital de Imágenes. México: Alfaomega Grupo Editor.

[9] Ting, L., Gangyi, J., Mei, Y. (2016). Inter-View Local Texture Analysis Based Stereo Image Reversible Data Hiding. Freng Shao University Ningbo, China Digital Signal Processing 48(2016)116-129 Elseiver.

[10] Valverde, J. (2010). Detección de bordes mediante el algoritmo de Canny. 29 de septiembre de 2016. https://goo.gl/nDqJ2v.