

# DETECTOR DE BORDES DE IMÁGENES USANDO UN MICROCONTROLADOR ARM

## EDGE DETECTION IMAGE USING ARM MICROCONTROLLER

Juan Francisco Ordóñez<sup>1</sup>, Luis Oñate<sup>2,\*</sup>

### Resumen

En este documento se describe cómo se desarrolló un detector de bordes de imágenes, esta técnica es útil en una variada gama de aplicaciones como metrología, reconocimiento de rostros, detección de patrones debido a que determina la frontera entre dos regiones de características distintas. Con este fin se utilizó la tarjeta de desarrollo STM32FI-DISCO, aunque este microcontrolador no está diseñado para procesamiento digital de imágenes (PDI) como es el caso de los microcontroladores de Texas Instruments, Analog Device, etc., sin embargo, tiene una velocidad de procesamiento elevada y utiliza 32 bits de programa. El detector de bordes utiliza el módulo del microcontrolador ARM de STmicroelectronics añadiéndole una cámara para tomar fotos y se presenta en un *display* de película delgada de transistor (TFT) la imagen donde se aplicó el detector de bordes usando el filtro de Sobel para determinar el movimiento de objetos mediante una diferencia de imágenes. Además, se implementó en MATLAB el filtro detector de bordes y se comparó las imágenes analizadas con el *software* MATLAB y las obtenidas en la tarjeta de desarrollo encontrando su semejanza, siendo las imágenes con mayor similitud: edificios, figuras geométricas y circuitos impresos. Con lo cual se logró aplicar un algoritmo de detección de bordes en este *hardware* a bajo costo.

**Palabras clave:** ARM, detector de bordes, filtro de Sobel, procesamiento digital de imágenes.

### Abstract

In this document is described as an image edge detector was developed, this technique is useful in a wide range of applications such as metrology, face recognition, pattern detection because it determines the border between two regions with different characteristics. the development board STM32FI-DISCO was used, although this microcontroller is not designed for digital image processing (PDI) as microcontrollers Texas Instruments, Analog Device, etc. has a speed high processing and uses 32-bit program. The edge detector module uses the ARM microcontroller STmicroelectronics adding a camera to take pictures and is presented on a *display* thin film transistor (TFT) image where the edge detector was applied using the Sobel filter to determine movement object by a difference images. In addition, the detector implemented in MATLAB filter edges and the images analyzed with the MATLAB *software* and obtained the development board finding its likeness, being the images with a high degree of similarity buildings, geometric figures and printed circuits compared. Thus it was possible to apply an edge detection algorithm on this *hardware* at low cost.

**Keywords:** ARM, edge detection, digital image processing, Sobel Filter.

---

<sup>1</sup>Graduado de la carrera Ingeniería Electrónica sede Quito, Universidad Politécnica Salesiana – Ecuador.

<sup>2,\*</sup>Docente de la carrera Ingeniería Electrónica sede Quito, Universidad Politécnica Salesiana – Ecuador. Autor para correspondencia ✉: lonate@ups.edu.ec

Recibido: 02-08-2016, aprobado tras revisión: 05-10-2016

Forma sugerida de citación: Oñate, L. y Ordóñez, J. (2016). «Detector de bordes de imágenes usando un microcontrolador arm». INGENIUS. N.º 16, (Julio-Diciembre). pp. 30-35. ISSN: 1390-650X.

## 1. Introducción

Este artículo es un primer intento de poner en funcionamiento detectores de bordes de imágenes en una tarjeta de desarrollo basada en el microcontrolador STM32, existen desarrollos basados en sistemas FPGA de Altera [1], FPGA de Xilinx [2] y utilizando el microcontrolador de Microchip el dspic33f [3].

En cualquier desarrollo de procesamiento digital de imágenes uno de los principales problemas es la cantidad de memoria, que debe tener el microcontrolador y la cantidad de cálculos que deben realizar [4] los microcontroladores ARM y en especial el STM32, tiene altas capacidades de memoria, y un módulo FPU (unidad de punto flotante) [5].

Entre las principales aplicaciones de los detectores de bordes se encuentran la identificación de tumores cerebrales, mediante el análisis de una fotografía MRI (Imagen de resonancia magnética) [6]. En detección de movimiento en videos de vigilancia, un objeto en movimiento siempre se mostrara cambios y si un objeto permanece estático sus bordes no variaran [7]. Reconocimiento de objetos en una imagen o video separando en partes la imagen y buscando ciertas características particulares como son los rostros de personas, o las formas de un objeto [8]. Aplicaciones en geología donde se puede diferenciar las líneas de las fallas geológicas de las líneas de los ríos subterráneos [9]. Sistemas biométricos para detección de los contornos del iris del ojo. [10]

El presente trabajo propone el diseño e implementación de un detector de bordes usando el filtro de Sobel, se utiliza el microcontrolador ARM STM32FI el mismo que tiene 32 bits, una cámara para tomar las imágenes, un *display* TFT. Se procede con esta tarjeta a adquirir la imagen original y la imagen procesada. Usando un par de interruptores, el primero permite detectar los bordes de las imágenes, y el segundo envía los datos a una computadora personal, la imagen tomada por la cámara y la imagen a la que se aplicó el filtro para detección de bordes. En el *software* MATLAB se filtra la imagen tomada por la tarjeta construida mediante el algoritmo de Sobel, y se compara las dos imágenes procesadas. Se aprovechó la versatilidad de la programación en lenguaje C de la tarjeta de desarrollo STM32FI-DISCO.

Este artículo está organizado de la siguiente manera en la sección II: se describen los materiales y métodos para el diseño del detector de bordes, en la sección III: los resultados y la comparación entre las imágenes filtradas el *software* MATLAB y en la tarjeta de procesamiento y, en la sección IV las conclusiones.

## 2. Materiales y métodos

### 2.1. Diseño del detector de bordes

El diseño del detector de bordes de imágenes mediante el filtro de Sobel se basó en el diagrama de bloques que se puede observar en la Figura 1, el mismo que está constituido por la cámara que toma la foto de la imagen, el procesador STM32FI DISCO donde se realiza el algoritmo de filtrado, la pantalla TFT para visualizar la imagen y la conexión serial para transmitir los datos al PC. La computadora personal mediante un cable USB sirve como la fuente de energía de este módulo de procesamiento digital de imágenes.

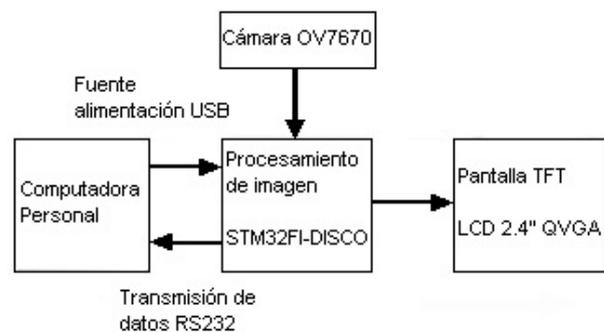


Figura 1. Diagrama de bloques

El procesamiento de las imágenes permite determinar si hay cambios en el movimiento con una simple diferencia. La detección de cambios en la imagen sirve determinar si hay movimiento con aplicaciones en seguridad o para conocer si una edificación o un circuito presenta deterioros. Para la determinación de los bordes, el microcontrolador toma la fotografía de la imagen y hace un barrido de una matriz de pixeles 3x3, aplicando la máscara de Sobel a cada matriz, con lo cual se suavizan las zonas de pixeles que son continuas y se refuerzan las zonas donde los pixeles cambian abruptamente de intensidad.

### 2.2. Sensor de imagen

La cámara OV7670 es un sensor de imagen CMOS de bajo voltaje funciona como una cámara VGA, necesita una fuente de alimentación entre 2,5 voltios y 3 voltios, opera a 30 FPS (tramas por segundo), tiene la capacidad de reducir el ruido de iluminación y eléctrico, así como los efectos de niebla, manchas. Además, usa una interfaz compatible con I2C y trabaja con formato RGB565 [11]. El formato RGB565 guarda las intensidades en 2 bytes o 16 bits, los colores rojo y azul tienen valores de intensidad que varían entre 0 y 31 y el verde varía entre 0 y 63, el 565 indica el número de bits que se utiliza para cada color. [12]

### 2.3. Tarjeta STM32FI DISCO

La tarjeta de desarrollo STM32FI DISCO es el cerebro del procesador de imágenes, está basada en un microcontrolador ARM de 32 bits de la casa STMicroelectronics, tiene 2 Mbytes de memoria flash, un reloj con frecuencia máxima de 180 MHz, 225 DMIPS, un controlador dedicado para LCD TFT, convertidores A/D y D/A de 12 bits, una DMA (acceso directo de memoria) de 16 bits de tipo FIFO [13], con instrucciones orientadas hacia el lenguaje de programación C.

La tarjeta de desarrollo toma la fotografía después de que un interruptor es presionado, controlando la cámara OV7670, la misma que envía los datos mediante el puerto de comunicaciones I2C, lo almacena en memoria flash del microcontrolador, y procede a enviar los datos para que sean visualizados en la pantalla TFT. Al presionar el segundo interruptor, el microcontrolador aplica el algoritmo de Sobel y presenta los resultados en la pantalla TFT. Posteriormente se envía la imagen original y la imagen con los bordes detectados al PC por medio de un módulo serial. La pantalla TFT es parte del módulo de desarrollo, sus características son las siguientes: soporta el formato RGB con 240 x 320 pixeles, necesita una fuente de 5 voltios y se comunica con bus paralelo de 8 o 16 bits [14]. Para la comunicación serial con el PC se utiliza el módulo CP2102 para transmitir datos mediante comunicación serial RS232 [15], las imágenes que son almacenadas en la memoria flash y que posteriormente serán reconstruidas en el *software* MATLAB como una matriz de intensidades RGB. En la Figura 2 se puede observar el módulo de desarrollo.

### 2.4. Filtro de Sobel

Aparte de los filtros que disminuyen el ruido en las imágenes. Existen los filtros que detectan los bordes de una imagen. Un borde es una discontinuidad, de tal forma que la detección de los mismos se basa en obtener la derivada de la función de la imagen [16]. Se aplican las siguientes ecuaciones para determinar el gradiente de la imagen [17]:

$$\frac{df}{dx} = \left(\frac{1}{6}\right) (M_x \circ N_8 [x, y]) \quad (1)$$

$$\frac{df}{dy} = \left(\frac{1}{6}\right) (M_y \circ N_8 [x, y]) \quad (2)$$

$$\nabla f = \sqrt{\frac{df}{dx}^2 + \frac{df}{dy}^2} \quad (3)$$

$$\theta = \tan^{-1} \left( \frac{\frac{df}{dx}}{\frac{df}{dy}} \right) \quad (4)$$



Figura 2. Tarjeta de desarrollo de procesamiento de imágenes

La máscara M sobre una imagen N, la máscara de Sobel es una matriz 3 x 3, la cual tiene un centro estimado que es el doble del valor del pixel vecino, existe una máscara  $M_x$  y una máscara  $M_y$  [17] como se puede ver en la figura 3.

$$M_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$M_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Figura 3. Máscaras  $M_x$  y  $M_y$  para el filtro de Sobel

Al aplicar sobre una matriz de pixeles de orden 3x3 de los vecinos al punto de análisis.

$$G_{pixeles} = \begin{bmatrix} l1 & l2 & l3 \\ l4 & l5 & l6 \\ l7 & l8 & l9 \end{bmatrix}$$

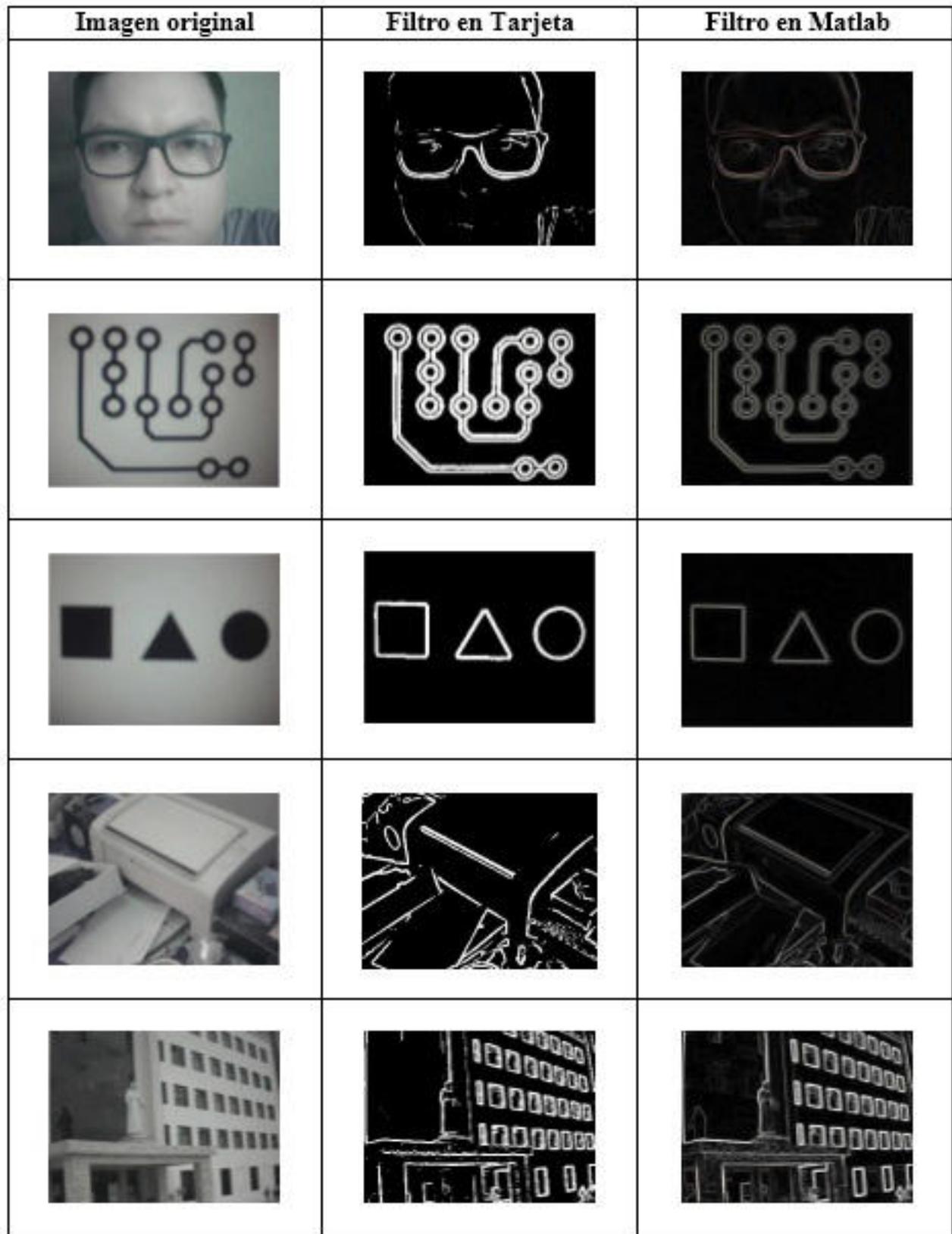


Figura 4. Comparación entre los bordes de las imágenes obtenidos con la tarjeta de desarrollo y el *software* MATLAB

Aplicando la máscara de Sobel se obtiene:

$$G_x = (l7 + 2 \cdot l8 + l9) - (l1 + 2 \cdot l2 + l3)$$

$$G_y = (l3 + 2 \cdot l8 + l9) - (l1 + 2 \cdot l2 + l7)$$

El cálculo del gradiente de la combinación de la matriz horizontal y vertical.

$$G = \sqrt{G_x^2 + G_y^2} \quad (5)$$

Este valor debe ser remplazado en cada una de las matrices RGB de cada pixel analizado [18].

### 3. Resultados de discusión

En la Figura 4 se puede observar en la columna izquierda la imagen tomada usando el equipo para adquirir imágenes con el microcontrolador STM32FI DISCO, en la columna del centro se observa los bordes obtenidos de la imagen tomada con el equipo de procesamiento de imágenes, en la columna de la derecha se observa la imagen analizada con el *software* MATLAB, la misma que ha sido enviada desde la tarjeta de procesamiento de imagen mediante el puerto serial, en el *software* MATLAB se ha aplicado el filtro de Sobel para determinar los bordes y conocer qué grado de precisión que tiene el detector de bordes instalado con el microcontrolador ARM. Para determinar la similitud de las imágenes se utilizó el método de correlación [18]. Se ha tomado distintos tipos de imágenes como son el rostro del estudiante Juan Francisco Ordóñez, un circuito impreso, figuras geométricas, una impresora, y parte del edificio del campus Sur Quito, como se puede ver en la Tabla 1. Las imágenes con mayor similitud tienen un valor de correlación cercano a 1; cuanto más se acercan las imágenes al valor de correlación igual a cero hay más diferencias.

**Tabla 1.** Valores de Correlación obtenidas al comparar las imágenes procesadas con el microcontrolador STM32 y el *software* MATLAB

% de correlación	Imagen
0,8744	circuito impreso
0,8700	figuras geométricas
0,8659	edificio
0,7533	impresora
0,6794	rostro

Las imágenes procesadas utilizando el método de Sobel permiten detectar los bordes con mayor precisión en circuitos impresos, figuras geométricas y edificios

en los cuales los bordes de la imagen a ser procesada están bien definidos, en el caso del rostro y la impresora, aunque se logra detectar los bordes no se llega a tener un grado de similitud para recomendar este método.

### 4. Conclusiones

Nosotros hemos presentado en esa investigación la detección de bordes de diversas imágenes utilizando un microcontrolador ARM de bajo costo y el método de Sobel. Se recomienda el uso de este método en detección de borde bien definidos en las imágenes, por lo cual tendría una buena aplicación en el uso para reconstrucción de edificios, planos de casas, y circuitos impresos. En el futuro se puede proponer utilizar métodos más robustos de tipo heurístico en combinación con los filtros tradicionales de Sobel, Canny entre otros.

### Referencias

- [1] G. Anusha, J. T, and N. Satya, "Implementation of sobel edge detection," *International Journal of Computer Trends and Technology*, vol. 3, no. 3, pp. 472–475, 2012.
- [2] S. Ravi, A. Rahim, and F. Shaik, "Fpga based design and implementation of image edge detection using xilinx system generator," *International Journal of Engineering Trends and Technology*, vol. 4, no. 10, pp. 4657–4660, 2013.
- [3] P. Premaratne, Ajaz, Monaragala, Bandara, and M. Premaratne, "Design and implementation of edge detection," in *2010 Fifth International Conference on Information and Automation for Sustainability*, Colombo, 2010.
- [4] L. Tan, *Digital Signal Processing, Fundamentals and Applications*, Academic Press, 2013.
- [5] STMicroelectronics. (2016). [Online]. Available: <http://www2.st.com/content/ccc/resource/technical/document/datasheet/03/b4/b2/36/4c/72/49/29/DM00071990.pdf/files/DM00071990.pdf/jcr:content/translations/en/DM00071990.pdf>.
- [6] S. Pratibha, M. Diwakar, and C. Sangam, "Application of edge detection for brain," *International Journal of Computer Applications*, vol. 58, no. 16, pp. 21–25, 2012.
- [7] M. Anupam and K. Debaditya, "Motion analysis in video surveillance using edge detection," *IOSR Journal of Computer Engineering*, vol. 12, no. 6, pp. 10–15, 2013.

- [8] Y. Ramadevi, T. Sridevi, B. Poornima, and B. Kalyani, "Segmentation and object recognition using edge detection techniques," *International Journal of Computer Science & Information Technology (IJCSIT)*, vol. 2, no. 6, pp. 153–161, 2010.
- [9] U. Sefercik and O. Gulegen, "Edge detection in geologic formation extraction: Close range and remote sensing case studie," 2012.
- [10] N. Singh, D. Gandhi, and K. P. Singh, "Recognition system using a canny edge detection and a circular hough transform," *International Journal of Advances in Engineering & Technology*, vol. 1, no. 2, pp. 221–228, 2011.
- [11] Omnivision. (2016) datasheet ov7670. [Online]. Available: <http://www.voti.nl/docs/OV7670.pdf>.
- [12] R. Green and M. Zechner, "Beginning android games, New York: Apress," pp. 89–91, 2012.
- [13] STMicroelectronics. (2016) Stm32f series. ISSN 1424-8220. [Online]. Available: [http://www2.st.com/content/st\\_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f3-series.html](http://www2.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32f3-series.html).
- [14] Wide. (2016) 2.4" QVGA (320x240) tft-lcd. [Online]. Available: <http://www.wide.hk/pdf/240320.pdf>.
- [15] S. Labs. (2016) Cp2102-9. [Online]. Available: <https://www.silabs.com/Support%20Documents/TechnicalDocs/CP2102-9.pdf>.
- [16] C. Solomon and T. Breckon, "Fundamentals of digital image processing," *Oxford: Wiley-Blackwell*, pp. 97–103, 2011.
- [17] L. Shapiro and G. Stockman, "Computer vision," *New Jersey: Prentice-Hall*, pp. 156–166, 2001.
- [18] S. Jayaraman, "Digital imagen processing," *New Delhi: Tata McGraw Hill*, pp. 127–132, 2009.