Detección de bordes en tiempo real empleando el filtro de Sobel y tecnología reconfigurable

Julio Cesar Sosa Savedra

Instituto Politécnico Nacional-ESCOM. Av. Juan de Dios Batz s/n, La Escalera, Gustavo A. Madero, 07738, México D.F., México. Tel. +52 55 57296000 Ext. 52064.

jcsosa@ipn.mx

Rubén Ortega González

Instituto Politécnico Nacional-ESCOM. Av. Juan de Dios Batz s/n, La Escalera, Gustavo A. Madero, 07738, México D.F., México. Tel. +52 55 57296000 Ext. 52064.

rortegag@ipn.mx

Víctor Hugo García Ortega

Instituto Politécnico Nacional-ESCOM. Av. Juan de Dios Batz s/n, La Escalera, Gustavo A. Madero, 07738, México D.F., México. Tel. +52 55 57296000 Ext. 52064.

vgarciao@ipn.mx

Rubén Hernández Tovar

Instituto Politécnico Nacional-UPIITA. Av. Instituto Politécnico Nacional 2580, La Laguna Ticomán, Gustavo A. Madero, 07340, México D.F., México. Tel. +52 55 57296000 Ext. 56807.

rhtovar@ipn.mx

Resumen

En la visión por computadora y en el procesamiento digital de imágenes, la detección de bordes es un proceso muy importante el cual intenta capturar el significado de los objetos dentro de la imagen, entre las cuales están: discontinuidades, la geometría y características físicas de los objeto. Sin embrago, se sabe que las aplicaciones de procesamiento de imágenes y video tienen un alto costo computacional, debido a que deben procesar una gran cantidad de datos. Por ello y gracias al avance tecnológico es posible desarrollar sistemas de visión en hardware. En este trabajo se presenta la

implementación de un procesador dedicado para la detección de bordes en tiempo real empleando un dispositivo lógico programable del tipo FPGA, la EP2C35F672C6 de Altera. El sistema detecta los bordes empleando el filtro de Sobel y emplea sólo el 6% de los elementos lógicos y el 43% de multiplicadores embebidos que posee el dispositivo. Los resultados son desplegados directamente en una pantalla compatible con VGA.

Palabras Claves: Bordes, FPGA, Sobel, Tiempo-real.

1. Introducción

El sistema visual humano utiliza una amplia gama de fuentes de información, como sombras, proporciones, longitudes, color, curvaturas e intensidades. La variación de la intensidad de los bordes se encuentra entre las más importantes de ellas, pues permite segmentar la imagen y reconocer objetos. En la visión por computadora y en el procesamiento digital de imágenes, la detección de bordes es un proceso muy importante que intenta capturar el significado de los objetos dentro de la imagen, entre las cuales están: discontinuidades, la geometría y características físicas de los objetos [1].

La detección de bordes debe ser eficiente y confiable porque la validez, la eficiencia y la posibilidad de la finalización de las etapas de procesamiento posteriores dependen de ella. Por lo antes expuesto, continúan desarrollándose investigaciones sobre nuevas formas para la detección de bordes [1]. Hay muchos métodos para detectar bordes, sin embargo es posible dividirlos en dos grandes categorías: aquellos basados en la primera derivada y los basados en la segunda derivada [2].

Las aplicaciones de procesamiento de imágenes y video tienen un alto costo computacional, debido a que deben procesar una gran cantidad de datos. Esto ha provocado que muchas aplicaciones de cómputo específico empleen dispositivos lógicos programables (PLDs), en vez de computadoras de propósito general. Los PLDs presentan una gran versatilidad, capacidad de paralelización y reconfigurabilidad, por ello son utilizados en una gran cantidad de aplicaciones [3,4]. Además, se sabe que las

aplicaciones basadas en PLDs o FPGAs, han mostrado que poseen un mayor desempeño por Watts que muchas arquitecturas de propósito específico [4].

Este trabajo se presenta la implementación de un algoritmo para la detección de bordes con el filtro de Sobel y un dispositivo lógico programable del tipo FPGA. El sistema se ejecuta en tiempo real y emplea el FPGA EP2C35F672C6, de Altera.

2. Desarrollo

2.1 Sistema de visión

Un sistema de visión general consta de una etapa de adquisición, una etapa de pre-procesado, una etapa de procesado y una etapa final o de salida, como se muestra en la Fig. 1. En este trabajo se pretende desarrollar sólo tres de estas etapas, la de adquisición, del pre-procesado y el módulo para poder representar la salida.

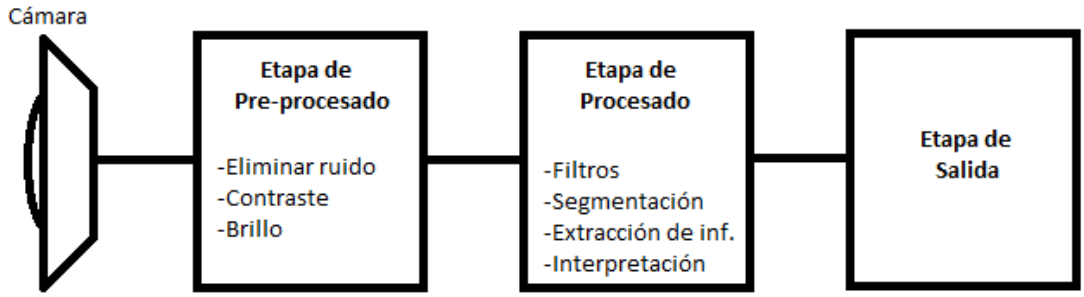


Fig. 1. Diagrama a bloque de un sistema de visión general.

La etapa de adquisición (cámara) consta de un sensor a color de tecnología CMOS [5], la cual ya contiene la lente y la etapa para la interconexión con la tarjeta que se empleará para desarrollar el sistema, que es la tarjeta de desarrollo DE2 de Altera [6].

La etapa de pre-procesado, consiste básicamente en realizar dos procesos. El primero de ellos es pasar de una imagen a color a una imagen a blanco y negro. El segundo proceso es aplicar el filtro de Sobel para obtener los bordes de los objetos.

Finalmente, la etapa de salida consiste en desplegar a través de un monitor VGA, el filtrado de la imagen.

2.2. Diseño

Para el desarrollo del sistema se pretende seguir un flujo de diseño empleado en el prototipado con dispositivos lógicos programables [7]. Dicha flujo es mostrado en la Fig. 2.

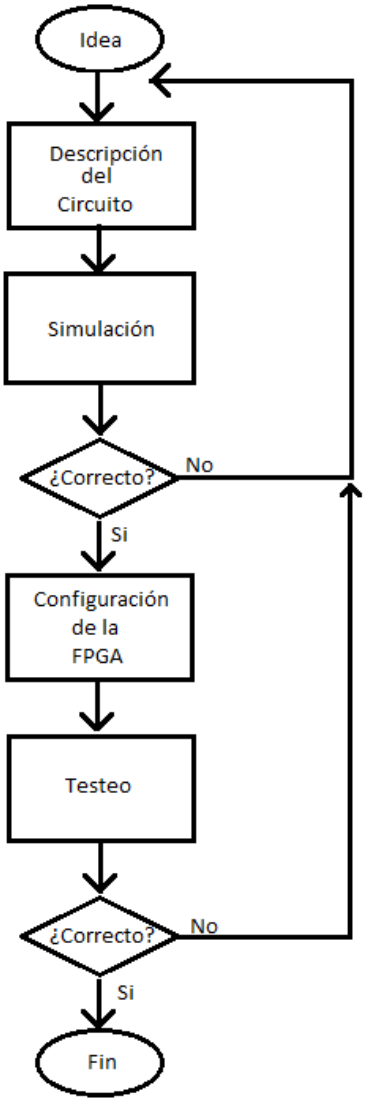


Fig. 2. Diagrama de flujo de diseño.

Existen distintas metodologías para diseñar un sistema de este tipo, sin embargo la flexibilidad y pertinencia que presentan las herramientas de diseño asistido por computadora para la automatización del diseño electrónico (CAD-EDA) permiten emplear la metodología de diseño *Top-Down*. Esta metodología de diseño consiste en capturar la idea en un alto nivel de abstracción e implementarla partiendo de esa descripción abstracta y después ir hacia abajo incrementando el nivel de detalle según sea necesario.

En otras palabras esta metodología sigue un poco el lema de divide y vencerás, de manera que un problema complejo es dividido en varios problemas de menor complejidad que llamaremos módulos los cuales a su vez podrán ser divididos en sub-módulos hasta llegar a componentes básicos o primitivas.

La arquitectura propuesta para el diseño de este sistema se muestra en la Fig. 3. Primero se realizará el controlador para capturar las imágenes, para ellos se accederá al sensor vía la interfaz I2C. Una vez que se inicia la captura de las imágenes a color se aplica un filtro para obtener las imágenes en 256 tonos de gris. Posteriormente se aplica el filtro de Sobel y se envía la imagen resultante al monitor VGA. Para poder manejar dicha interfaz es necesario desarrollar el controlador VGA del mismo.

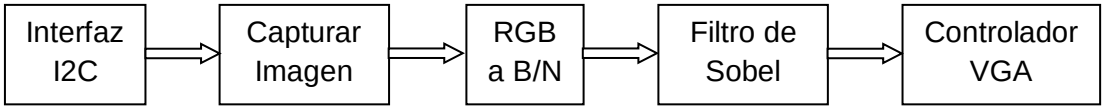


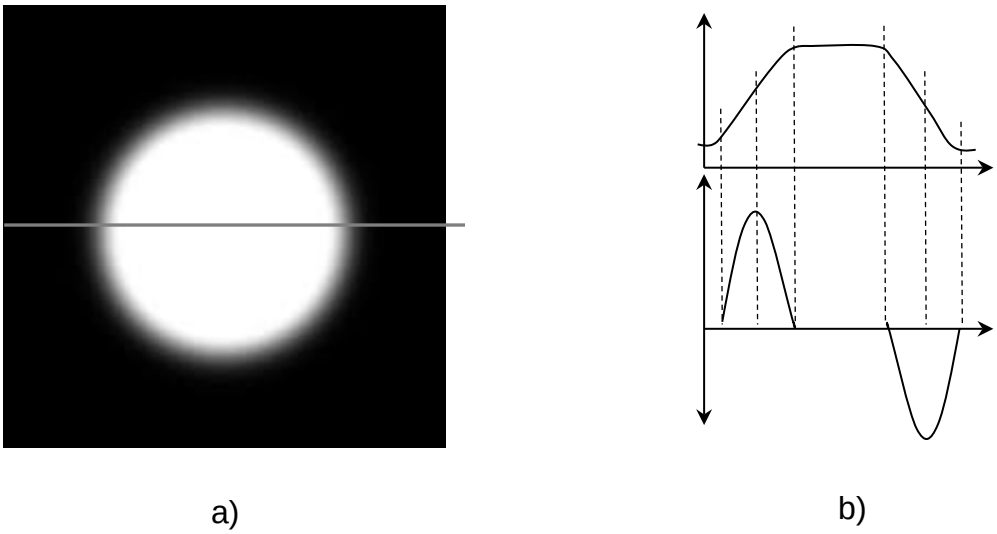
Fig. 3. Diagrama a bloques de la arquitectura propuesta.

Un punto importante a considerar es que las imágenes serán almacenadas en la memoria RAM, que posee la tarjeta de desarrollo DE2, la cual posee una SDRAM de 8 Mbyte. Así pues, es necesario diseñar el controlador de dicha memoria la gestión de lectura y escritura en la misma.

Si bien, la mayoría de los bloques consisten en estudiar cada uno de los manuales y protocolos necesarios para el desarrollo. Lo primero fue implementar un sistema que sólo realice la captura y despliegue de las imágenes a color que son capturadas por el sensor de visión. Una vez hecho esto se procedió a pasar las imágenes a blanco y negro para posteriormente aplicar el filtro de Sobel.

2.3 El filtro Sobel

Existen una gran cantidad métodos para la detección de bordes. Entre ellos están los basados en la primera derivada y los basados en la segunda derivada, también conocidos como máximo local del gradiente y cruce por cero del laplaciano, respectivamente. Para comprender esto, consideremos una sola dimensión y tomemos como ejemplo una imagen con fondo negro y en el centro una región blanca, como se muestra en la Fig. 4a. El perfil en escala de grises a lo largo de una línea de la imagen podría verse como lo muestra en la Fig. 4b. Se definirá esta señal unidimensional como $f(u)$ y calculando a partir de ella su primer derivada mediante $f'(u) = df/du\ (u)$. Así se produce una elevación positiva en todo lugar donde la intensidad aumenta y una negativa donde la intensidad disminuye. Sin embargo, la derivada no está definida para funciones discreta de $f(u)$ por lo que es necesario un método para calcularla.



a) b)

Fig. 4. Detección de bordes con primera y segunda derivada.

Si bien, la derivada de una función continua en un punto x puede ser interpretada por la pendiente de la tangente en ese punto en particular. Para una función discreta la derivada

en un punto ***u*** puede ser calculada a partir de la diferencia existente entre los puntos vecinos a ***u*** dividido por el valor de muestreo entre ambos puntos. Por lo que la derivada puede ser aproximada por la ecuación 1. Y el mismo proceso podría ser llevado para el sentido vertical ***v***, a lo largo de la columna de la imagen.

$$\frac{df}{du}(u) \approx \frac{f(u+1) - f(u-1)}{2} \tag{1}$$

Entonces, sí $\frac{\partial I}{\partial x}(x,y)$ y $\frac{\partial I}{\partial y}(x,y)$ expresan la derivada parcial de la función de la imagen ***I(u,v)*** con respecto a la variable ***u*** o ***v*** [8]. Entonces el vector del gradiente de la función ***I*** en un punto ***(x,y)***, está dado por la ecuación 2.

$$\nabla I(x,y) = \begin{bmatrix} \frac{\partial I}{\partial x}(x,y) \\ \frac{\partial I}{\partial y}(x,y) \end{bmatrix} \tag{2}$$

Y la magnitud del gradiente está definida como:

$$|\nabla I| = \sqrt{(\nabla_x)^2 + \left(\nabla_y\right)^2} \tag{3}$$

Y la dirección del gradiente en cada píxel es calculado a partir de:

$$\emptyset(\mathrm{x},\mathrm{y}) = tan^{-1}\left(\nabla_x, \nabla_y\right) \tag{4}$$

Las componentes del gradiente de la ecuación 2 no son otra cosa que la primera derivada tanto en el sentido de los renglones como en el de las columnas de la imagen. Una forma de calcular la primera derivada es empleando operadores y dos de los operadores más utilizados en la detección de bordes son: el de Prewitt y el de Sobel, mostrados en la ecuación 4 y 5, respectivamente [8].

$$H_x^P = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ y } H_y^P = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{5}$$

$$H_x^S = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ y } H_y^S = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (6)$$

Estos operadores son muy similares, como se puede observar, la única diferencia es que el filtro de Sobel le da un mayor peso al renglón o columna central del filtro. Sin embargo el operador Sobel es a razón de sus buenos resultados y facilidad de implementación muy utilizado e implementado en la mayoría de los paquetes de software comerciales utilizados para el procesamiento de imágenes digitales, según [8]. Por esta razón en este trabajo se decidió a implementar dicho operador.

2.4 Implementación

Para la implementación del sistema se emplearon interruptores con la finalidad de separar los procesos e ir evaluando cada una de las etapas. De esta manera se emplearon los interruptores SW16 y SW 17, de la tarjeta de desarrollo DE2.

Cuando ambos interruptores están apagados en la salida sólo se desplegará las imágenes adquiridas por el sensor, que son imágenes a color. Si el interruptor SW16 está apagado y SW17 encendido, entonces sólo convierte las imágenes en escala de grises y son mostradas en el monitor VGA.

Finalmente, si el interruptor SW16 esta encendido, entonces se convierten las imágenes a blanco y negro y además se aplica el filtro Sobel en el proceso, como se puede apreciar en la Fig. 5.

El diseño se implementó utilizando la herramienta de Quartus II software web edition, de Altera. Para su diseño, se empleó el lenguaje de descripción de hardware Verilog. Parte del código del módulo de Sobel, con las máscaras se presenta en la tabla 1.

module Sobel (	// mascara x

input iCLK,
input iRST_N,
input [9:0] iDATA,
input iDVAL,
input [7:0] iTHRESHOLD,
output [9:0] oDATA
);
	parameter X1 = 8'hff, X2 = 8'h00, X3 = 8'h01;
	parameter X4 = 8'hfe, X5 = 8'h00, X6 = 8'h02;
	parameter X7 = 8'hff, X8 = 8'h00, X9 = 8'h01;
	// mascara y
	parameter Y1 = 8'hff, Y2 = 8'hfe, Y3 = 8'hff;
	parameter Y4 = 8'h00, Y5 = 8'h00, Y6 = 8'h00;
	parameter Y7 = 8'h01, Y8 = 8'h02, Y9 = 8'h01;

Tabla 1. Etapa del módulo del filtro Sobel.

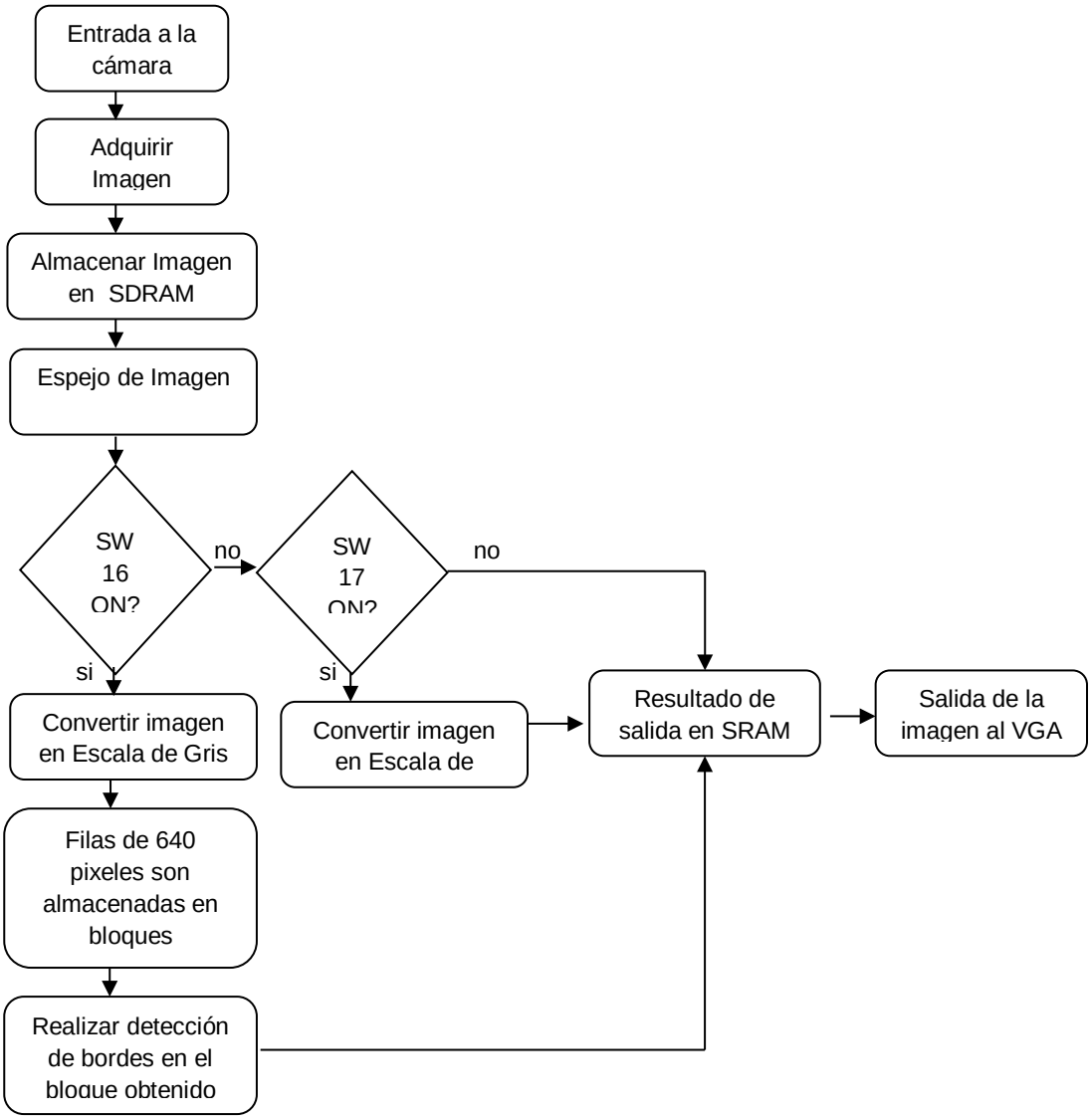


Fig. 5. Diagrama de flujo del sistema.

3. Resultados

El sistema de detección de bordes se ejecuta en tiempo real y es posible procesar más de 24 fps con una frecuencia máxima de reloj de 65 MHz. Se empleó la FPGA EP2C35F672C6, sólo uso el 6% de los elementos lógicos que posee y el 43% de los Multiplicadores embebidos, todo esto se puede observar en la Fig. 6.

Fig. 6. Reporte de la síntesis del circuito.

Adicionalmente, en este trabajo se redujo el número de elementos lógicos (LEs) utilizados respecto con los resultados publicados, por Z. E. M. Osman, en [9].

En [9] implementaron el sistema sobre una FPGA EP2S60 en la cual utilizaron el 8% del total de elementos lógicos. En este trabajo se empleó la EP2C35, la cual posee menos recursos, y utilizó menos del 6% de elementos lógicos que posee el dispositivo. En la tabla 2 se presenta la cantidad de recursos que posee cada dispositivo y el 8% de 60,440 da un total de 4,835 elementos lógicos utilizados. En este trabajo se utilizaron sólo 1,920 elementos lógicos como se puede apreciar en la Fig. 6.

Características	EP2S60 [9]	EP2C35
Elementos Lógicos (Les)	60,440	33,216
M4K Ram blocks	255	105
RAM bits	2,544,192	483,840
Multiplicadores embebidos	144 de 18 bits	35 de 9 bits
PLLs	12	4

Tabla 2. Características de las FPGAs.

Se realizaron varias pruebas, con el fin la velocidad y capacidad de segmentación del sistema. En la Fig. 7, se presentan las fotos del proceso de adquisición de las imágenes del sensor a color y cuando se selecciona la salida monocromática, en 256 tonos de gris. En la Fig. 8 se presenta una imagen de la salida una vez que se aplica el filtro Sobel y el sistema en donde se está realizando el procesado.

Fig. 7. Fotos de las imágenes del monitor, izquierda a color, derecha en tono de gris.

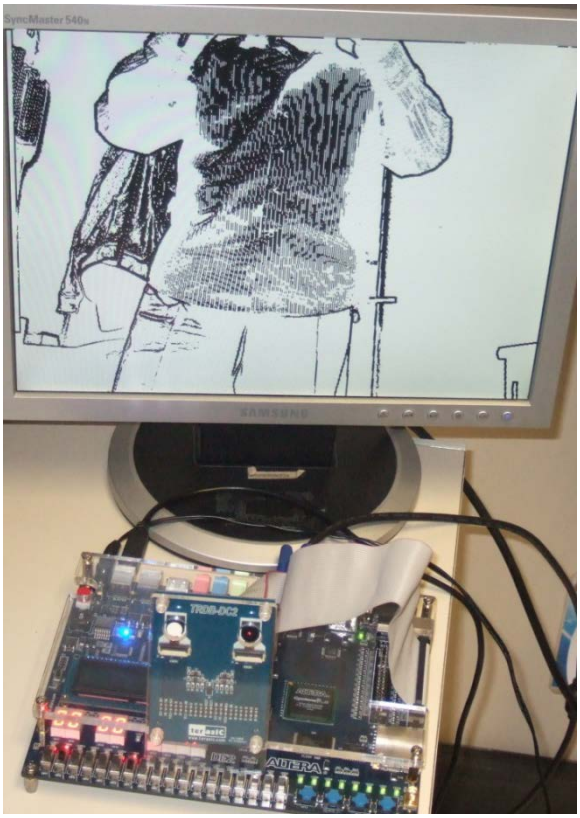


Fig. 8. Foto de la imagen del monitor y el sistema implementando el filtro Sobel.

Siguiendo con este proceso se expusieron diferentes tipos de objetos, frente a la cámara, como son: puerta y techo, una persona mostrando una caja, un borrador, un bolígrafo y una placa, como se aprecia en la Fig. 9, Fig. 10 y Fig. 11, respectivamente. En la Fig. 9 se ve la foto a color de una puerta, con fondo de madera y su techo de falso plafón. En dicha foto también aparecen hojas a la izquierda y derecha de la imagen. Se puede apreciar que al aplicar el filtro de Sobel los bordes son resaltados. Los puntos que existen en el plafón una vez aplicado el filtrado, son debido a la porosidad del plafón.

Fig. 9. Fotos de la puerta y techo, izquierda a color, derecha en tono de gris.

En la Fig. 10 se muestran las imágenes del resultado de aplicar el filtro Sobel, a una persona mostrando una caja y un borrador de pizarrón, en el mismo fondo de la Fig. 9.

Fig. 10. Fotos de imágenes después de aplicar el filtro Sobel, izquierda de una caja y derecha de un borrador.

En la Fig. 11 se muestran las imágenes del resultado de aplicar el filtro Sobel, a una persona mostrando un bolígrafo en el mismo fondo de la Fig. 9 y una placa impresa en una hoja tamaño carta, pegada en la puerta.

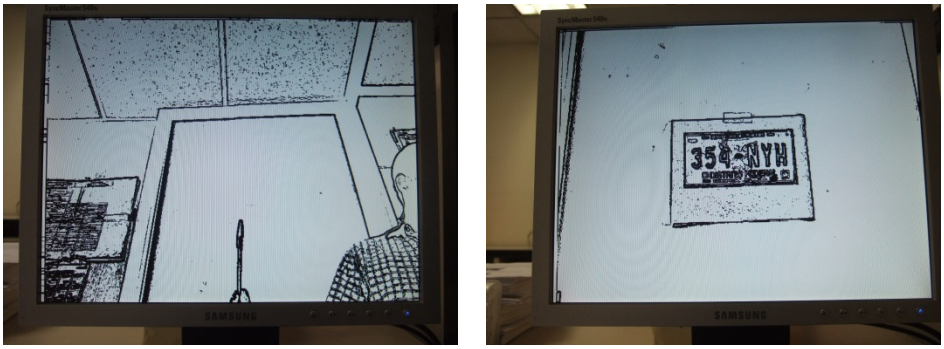


Fig. 11. Fotos de imágenes después de aplicar el filtro Sobel, izquierda una persona mostrando un bolígrafo y derecha, placa impresa en una hoja.

4. Discusión

Los resultados obtenidos en este trabajo permitieron implementar un filtro Sobel en tiempo real, esto a pesar de tener un sensor a color. El hecho de emplear un sensor que adquiere imágenes a color obliga a realizar un procesado adicional que pase de las imágenes adquiridas en color a imágenes en blanco y negro. Por lo tanto es deseable contar con un sensor que adquiera las imágenes en blanco y negro. Debido a que la aplicación del filtro de Sobel es realizada en tiempo real y es mostrada en un monitor no es posible hacer una comparación cuantitativa, sólo cualitativa. Bajo esto, es posible decir que el filtro se ejecuta con una excelente calidad, pues el marco de la puerta y cada uno de los objetos se aprecian los contornos existentes de los objetos, incluso hasta la oreja, nariz o boca de la persona, cuando aparece mostrando las piezas. Un inconveniente detectado es a distancias mayores a 2 metro, y sólo en los casos donde aparecen letras de cierta magnitud, las cuales no pueden ser reconocidas fácilmente. En dos imágenes, en donde aparece el borrador y la placa es posible alcanzar a ver la marca del borrador y las letras y números de la placa. Para eso fue necesario acercar a 1 metro de distancia de los objetos y de hecho la imagen en donde se muestra el borrador aparece la marca ***VINCI*** y la palabra ***Borrador***, los cuales tienen letras de 1 cm de altura en mayúsculas y minúsculas, respectivamente. La placa tiene letras de 4 cm de altura y en el centro aparece la figura del ángel de la independencia. En las imágenes en donde aparece una

persona, mostrando las piezas, esta lleva puesta una camisa de cuadros, los cuales son apreciados claramente en su segmentación. Finalmente, la arquitectura desarrollada redujo en casi un 40% de elementos lógicos utilizados, respecto al trabajo publicado en [9].

5. Conclusiones

La arquitectura propuesta para la implementación del filtro Sobel emplea recursos dedicados incorporados en la FPGA, así como recursos localizados en la misma tarjeta de desarrollo DE2, como la memoria SDRAM. Con esa estrategia se liberaron una gran cantidad de recursos de la FPGA. La finalidad es permitir implementar algún otro algoritmo en el mismo dispositivo. Así, sólo se emplearon 1,920 elementos lógicos, que es un 6 % del total de sus componentes y 30 de sus 70 multiplicadores embebidos.

Por otro lado, se logró implementar el sistema en tiempo real, logrando procesar más de 24 fps trabajando a una frecuencia máxima de reloj de 65 MHz. Es posible trabajar a mayores frecuencias si se utiliza una cámara a blanco y negro para evitar realizar la transformación de imagen a color a una escala de grises (blanco y negro).

Finalmente, es posible considerar que este trabajo podrá servir como una primera etapa para el reconocimiento de caracteres en placas de autos, o en otro tipo de imágenes, con la finalidad de realizar algún tipo de identificación.

Agradecimientos

Este trabajo ha sido financiado por el proyecto de investigación SIP: 20130647 y SIP: 20141144.

6. Referencias

[1] L. Zhai, S. Dong y H. Ma. “Recent Methods and Applications on Image Edge Detection”. International Workshop on Education Technology and Training and International Workshop on Geoscience and Remote Sensing. 2008, p.p: 332-335.

[2] W. K. Pratt. Digital Image Processing. Fourt Edition. 2007. Wiley. California, USA. 782 p.p.

[3] J. C. Sosa. “Sistema de visión basado en procesado guiado por cambios y lógica reconfigurable para el análisis de movimiento de alta velocidad”. 2007. Universidad de Valencia. Valencia, España. 301 p.p.

[4] J. Monson, M. Wirthlin, B. L. Hutchings. “Optimization techniques for a high level synthesis implementation of the Sobel filter”. International Conference on Reconfigurable Computing and FPGAs (ReConFig). 2013, p.p. 1-6.

[5] Micron Technology, Inc. 1/3-Inch Megapixel CMOS Active-Pixel Digital Image Sensor. Data Sheet. 2004, 41 p.p.

[6] Altera. DE2 Development and Education Board: User Manual. Altera Corporation. 2006, 69 p.p.

[7] F. Pardo, J. A. Boluda. VHDL: Lenguaje para síntesis y modelado de circuitos. Tercera Edición, 2011. Alfaomega. México, D.F. México. 308 p.p.

[8] E. Cuevas, D. Zaldívar y M. Pérez. Procesamiento digital de imágenes con Matlab y Simulink. 1ª Edición, 2010. Alfaomega. México D.F. México. 815 p.p.

[9] Z. E. M. Osman, F. A. Hussin y N. B. Z. Ali. “Hardware implementation of an optimized processor architecture for SOBEL image edge detection operator”. Int. Conference on Intelligent and Advanced Systems (ICIAS). 2010, p.p. 1-4.

7. Autores

Dr. Julio Cesar Sosa Savedra obtuvo su grado en Tecnología de la Información Comunicación y Computación por la Universidad de Valencia, España (2007). Su grado de M. en C. en Ingeniería Eléctrica por el CINVESTAV, México (2000) y su título de Ingeniero en Electrónica, por el Instituto Tecnológico de Lázaro Cárdenas Michoacán (1997). Sus áreas de interés son sistemas embebidos y procesamiento digital de señales e imágenes.

Dr. Rubén Ortega González obtuvo su grado de doctor en Ingeniería Eléctrica, Computación y Sistemas Electrónicos por la Universidad Politécnica de Valencia, España (2013). Su grado de M. en C. en Ingeniería en Sistemas por el IPN-SEPI-ESIME, Zacatenco (2001) y su título de Ingeniero en Comunicaciones y Electrónica por el IPN-ESIME, Zacatenco (1997). Sus áreas de interés son sistemas de control y procesamiento digital de señales e imágenes.

M. en C. Víctor Hugo García Ortega obtuvo su grado maestría en Ingeniería de Cómputo con especialidad en Sistemas Digitales, en el Centro de Investigación en Computación del IPN (2006). Su grado de Ing. en Sistemas Computacionales por la Escuela Superior de Cómputo, del Instituto Politécnico Nacional (1999). Actualmente es profesor titular en la Escuela Superior de Cómputo del IPN y trabaja en el área de Arquitectura de Computadoras, Microprocesadores y Procesamiento Digital de Imágenes y Señales.

Dr. Rubén Hernández obtuvo su grado de Doctor en Ciencias Técnicas por el Instituto de Cibernética, Matemática y Física de Cuba (2014). Su grado de M. en C. en Ingeniería Eléctrica por el CINVESTAV, México (1999) y su título de Ingeniero en Comunicaciones y Electrónica, por el IPN-ESIME, Zacatenco (1995). Sus áreas de interés son sistemas de comunicación y procesamiento digital de señales e imágenes.