

Sistema Digital para Procesamiento Básico de Imágenes con Cámara OV7670

Gaspar Badillo José Eduardo, Ramos Arreguín Juan Manuel✉, Pedraza Ortega Jesús Carlos y Efrén Gorrostieta Hurtado

Universidad Autónoma de Querétaro, Facultad de Ingeniería
✉jsistdig@yahoo.com.mx

Resumen

Las aplicaciones de procesamiento digital de imágenes han crecido de manera importante, hasta ser parte importante de la vida diaria. En el campo de sistemas embebidos, el desarrollo de aplicaciones que requieren del uso de procesamiento digital de imágenes también ha crecido significativamente. Este crecimiento se ha dado especialmente en aplicaciones de tiempo real, con la limitante del tiempo de procesamiento que cada sistema digital requiere. Una importante alternativa es el uso de Dispositivos Lógicos Programables, los cuales se caracterizan por su paralelismo natural en la implementación de hardware. Esto permite implementar algoritmos de procesamiento de imágenes, cuyo tiempo de cómputo depende directamente del diseño de la arquitectura de hardware. Debido a que la implementación de este tipo de algoritmos no es algo trivial, es importante establecer bases que permitan a un diseñador sin mucha experiencia, poder realizar este tipo de implementaciones en hardware. Este trabajo se enfoca a establecer las bases para implementar algoritmos de procesamiento de imágenes, facilitando el crecimiento del desarrollo de arquitecturas en hardware para el procesamiento digital de imágenes.

Palabras clave: Dispositivos Lógicos Programables, VHDL, Procesamiento digital de imágenes, binarización, escala de grises.

Abstract

The applications of digital image processing have grown significantly and are an important part of our life. In the field of embedded systems, the development of applications that require the use of digital image processing has also grown significantly. This growth has occurred especially in applications of real time, with the limitation of the processing time that each digital system requires. An important alternative is the use of Programmable Logic Devices, which are characterized by their natural parallelism in hardware implementation. This allows the implementation of image processing algorithms, whose computation time depends directly on the design of the hardware architecture. Due that the implementation of this type of algorithm is not trivial, it is important to establish bases that allows a designer the hardware implementation algorithm, when the designer has not experience to be able to carry out this type of implementation. This work focuses on establishing the bases to implement image processing algorithms, facilitating the growth of the development of hardware architectures for the digital processing of images.

Keywords: Programmable Logic Devices, VHDL, image digital processing, binarization, gray scale.

1. Introducción

El desarrollo de descripciones de hardware enfocados a procesamiento digital de imágenes tiene reportes en los inicios de la década del 2000, cuando se aprovecha a nivel de datos el paralelismo del FPGA (Compuerta Reconfigurable en Campo) en la implementación de algoritmos para procesamiento de imágenes [1]. En el 2011, se reporta un nuevo lenguaje específico de dominio (SDL), llamado CAPH, el cual es utilizado para la implementación de aplicaciones de procesamiento de flujo en campo de arreglos de compuertas programables (FPGA), con un intérprete y un compilador que generan el código VHDL [2]. Se han reportado trabajos de interfaces humano-máquina, para el reconocimiento de gestos [3].

En los años recientes, el uso de dispositivos lógicos programables ha ido en aumento especialmente para el desarrollo de sistemas de visión por computadora. Es por esto que Sérot reporta que las aplicaciones que requieren de un procesamiento de datos de manera continua, requieren un alto poder de cómputo, especialmente para el procesamiento de imágenes en tiempo real, por lo que son buenos candidatos para la aplicación de la lógica reconfigurable como matrices de puertas programables (FPGAs) [4]. En el siguiente año, se presenta las técnicas de diseño de sistemas embebidos utilizando arreglos de compuertas programables en campo (FPGA) para la aplicación de procesamiento de imágenes y vídeo [5]. En este mismo año, se propone un nuevo enfoque para la solución de problemas de automatización industrial de renombre tales como Control de Calidad y Paletización (QCP), y se usa un algoritmo de procesamiento de imágenes para modelar la inteligencia de un mecanismo de cuadrilátero singular [6]. Así mismo, se establece que un sistema de prevención de colisiones de alto rendimiento a bordo establece restricciones drásticas, que pueden cumplirse mediante el uso de arquitecturas computacionales de múltiples núcleos cuidadosamente optimizados [7]. También se han realizado descripciones de procesadores de imágenes, como el que reporta donde se presenta un procesador reconfigurable multi-propósito; sin embargo, requiere de imágenes ya binarizadas [8]. También se presenta un procesador de núcleo blando escalable orientado a aplicaciones de procesamiento de imágenes, basado en la arquitectura Xilinx DSP48E1 [9].

En el 2015, se presenta una descripción de hardware en FPGA para la aceleración del proceso de reconstrucción y segmentación simultánea de imágenes [10]. Posteriormente se propone una descripción de hardware donde la carga de procesamiento de imágenes se distribuye entre FPGAs y procesadores de señales digitales (DSP) para aprovechar los puntos fuertes de ambas tecnologías para su uso en inspección de superficies [11]. Se presenta un sistema para procesamiento de imágenes de alta resolución y alta fidelidad [12]. Mientras, se diseña un módulo para FPGA para el procesamiento de imágenes de resonancia magnética [13].

Las posibles aplicaciones de implementar procesamiento digital de imágenes tienen cada vez mayor relevancia, como es el caso de reconstrucción 3D de objetos [14] [15], entre otros.

En las líneas de trabajo actuales que se están llevando a cabo en la Facultad de Ingeniería, se tienen diversos trabajos que se han encaminado a la implementación de arquitecturas de hardware para el procesamiento de imágenes, de las cuales han surgido temas de tesis de maestría y de doctorado. Para poder contar con sistemas económicos y novedosos, es necesario diseñar e implementar diversos algoritmos de procesamiento de imágenes en sistemas embebidos basados en FPGA. Por lo tanto, se requiere modificar los algoritmos para que puedan ser implementados en hardware, y en ocasiones, se requiere diseñar métodos numéricos alternativos que ayuden a simplificar la implementación. Una alternativa es la implementación de procesos paralelos que permitan una mayor velocidad en la obtención del resultado buscado. Algunas posibles aplicaciones son: Dehazing, detección de movimiento, flujo óptico, entre otros.

2. Procesamiento básico.

Algunos de los tipos de procesamiento digital de imágenes más usados son los siguientes:

- Imágenes RGB.
- Escala de grises.
- Imágenes binarias.

Las imágenes RGB se caracterizan por tener 3 componentes de color: rojo (Red), verde (Green) y azul (Blue). La mezcla de estos valores son lo que da a cada pixel de la imagen el color deseado. Cada componente tiene un valor cuantizado de 0 a 255. La combinación de valores RGB determina el color que se puede visualizar en un pixel específico, de modo que el color negro se representa por la combinación RGB de (0,0,0). Mientras que el color blanco está representado por la combinación (255,255,255). La figura 1(a) muestra una imagen en formato RGB.

Las imágenes en escala de grises se caracterizan en que el valor de cada componente RGB tienen el mismo valor. Las imágenes en escala de grises pueden tener su origen en una imagen a color, o bien, desde su adquisición el sensor puede entregar el valor de la imagen en escala de grises. En el primer caso, donde la imagen puede tener su origen en una imagen de color, para obtener la escala de grises, se puede utilizar 3 métodos: promedio (average), ligero (lightness) y luminosidad (luminosity). El primer método, promedio, se obtiene sumando la intensidad de cada componente y dividiendo entre 3, como se ve en (1). El segundo método, ligero, es también un promedio, pero entre el valor más alto y el más bajo de las intensidades RGB, y dividiendo entre 2, lo que se aprecia en (2). Finalmente, el método de luminosidad se apoya en (3) para obtener el valor resultante de la escala de grises.

$$Gris = \frac{R + G + B}{3} \quad (1)$$

$$Gris = \frac{\max(R, G, B) + \min(R, G, B)}{2} \quad (2)$$

$$Gris = K_1R + K_2G + K_3B \quad (3)$$

El valor obtenido en *Gris* representa la intensidad del pixel en cada uno de los componentes RGB. En la ecuación 3, los valores K_i pueden tener valores entre 0 y 1, con la limitación de que se debe cumplir (4).

$$K_1 + K_2 + K_3 = 1 \quad (4)$$

La figura 1 muestra la conversión a escala de grises de una imagen a color, utilizando los métodos de conversión de (1), (2) y (3).

Las imágenes binarias se caracterizan porque cada pixel maneja solamente dos valores de intensidad, blanco y negro. Este tipo de imágenes tienen su origen en la escala de grises, donde se determina el valor de intensidad del pixel a partir de un valor de umbral, como se muestra en (5).

$$I(x, y) = \begin{cases} 255 & \text{si } Gris \geq U \\ 0 & \text{si } Gris < U \end{cases} = \begin{cases} \text{Blanco} & \text{si } Gris \geq U \\ \text{Negro} & \text{si } Gris < U \end{cases} \quad (4)$$

Donde $I(x, y)$ es la intensidad del pixel ubicado en la coordenada (x, y) del monitor. La variable U es el valor de umbral deseado. Este valor de umbral es definido por el usuario, pues depende de las características buscadas en la imagen.

La figura 2 muestra un ejemplo de imágenes binarias con un umbral de 80, comparando con los diferentes métodos de escala de grises. La figura 3 muestra las imágenes binarias con un umbral de 100.

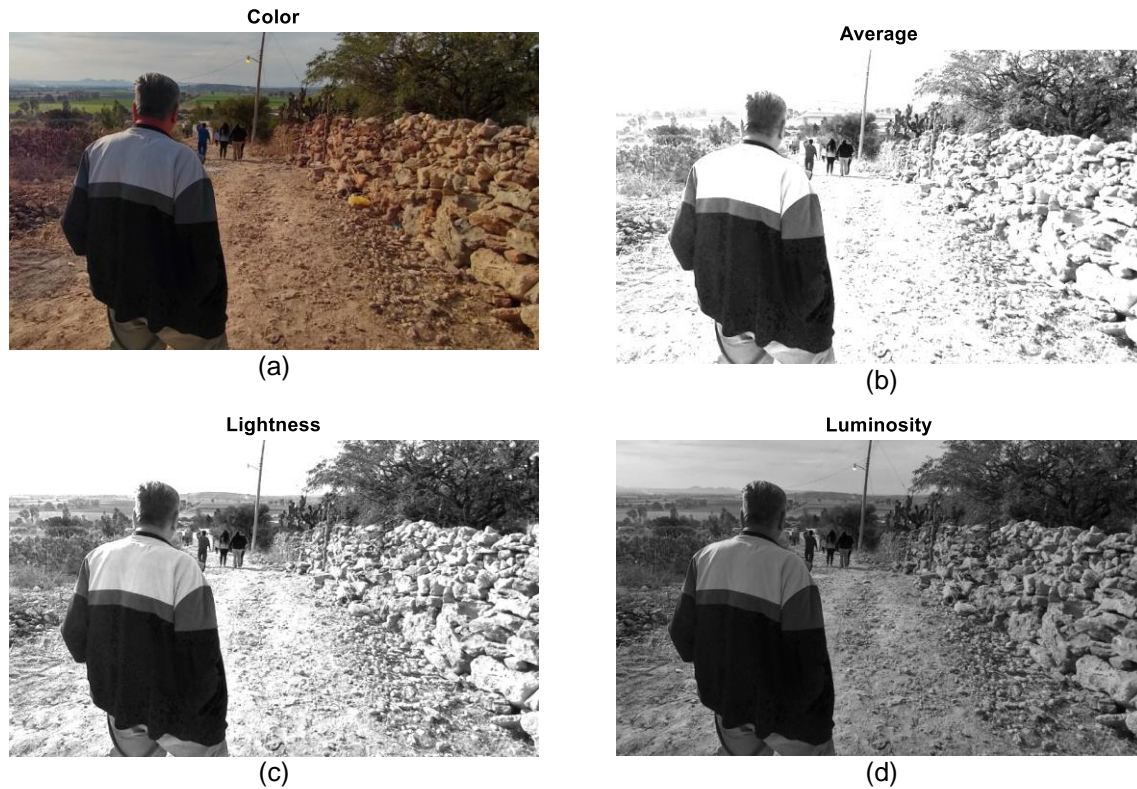


Figura 1. Ejemplo de imagen RGB. (a) Color. (b) Escala de grises método de promedio. (c) Escala de grises método ligero. (d) Escala de grises método de luminosidad con $K_1=0.3$, $K_2=0.6$ y $K_3=0.1$.



Figura 2. Imágenes binarias de acuerdo al método de escala de gris, con $U=80$. (a) Original. (b) Método de promedio. (c) Método ligero. (d) Método de luminosidad con $K_1=0.3$, $K_2=0.6$ y $K_3=0.1$.



Figura 2 (continuación). Imágenes binarias de acuerdo al método de escala de gris, con $U=80$. (a) Original. (b) Método de promedio. (c) Método ligero. (d) Método de luminosidad con $K_1=0.3$, $K_2=0.6$ y $K_3=0.1$.

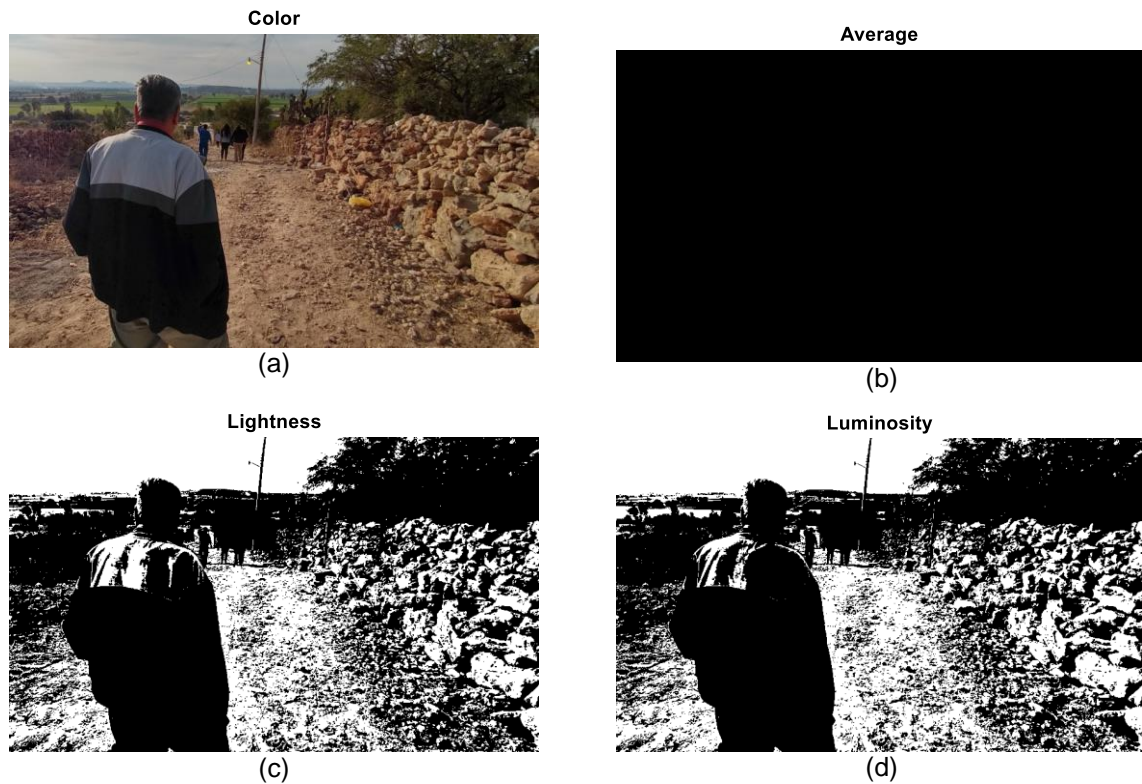
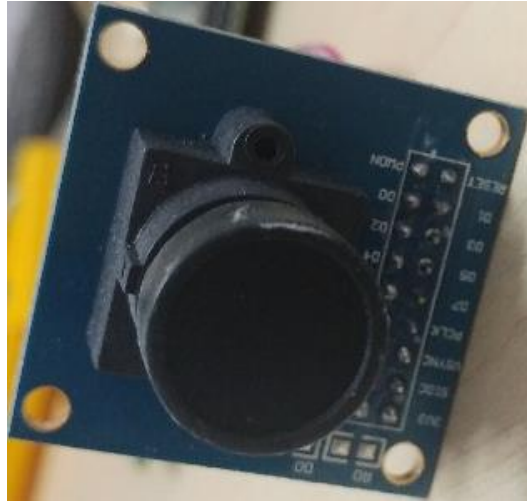


Figura 3. Imágenes binarias con $U=100$. (a) Original. (b) Método de promedio. (c) Método ligero. (d) Método de luminosidad con $K_1=0.3$, $K_2=0.6$ y $K_3=0.1$.

3. Adquisición de la imagen.

En este caso, se propone el uso de una cámara OV7670, la cual es una cámara que entrega imágenes VGA, es decir, en formato RGB. La interfaz de comunicación es compatible con I2C, y su consumo de energía es relativamente bajo.

La figura 4, muestra la cámara a usar, así como una relación de las terminales que utiliza.



Signal	Usage
3V3	3.3V de alimentación
Gnd	Tierra de alimentación
SIOC	Reloj del bus Serial de comandos (hasta 400KHz)
SIOD	Bus de datos de comandos serial
VSYNC	Sincronía vertical
HREF	CE de salida para muestreo de pixeles
PCLK	Reloj de pixel
XCLK	Reloj de sistema (10-48MHz, 24MHz típico)
D0-D7	Datos de pixel
RESET	Reinicio del dispositivo
PWDN	Apagado del sistema

Figura 4. Cámara OV7670 y su descripción de terminales.

La figura 5 muestra un bloque descriptivo de hardware (BDH) correspondiente a la entidad del proyecto de procesamiento de imágenes en hardware.

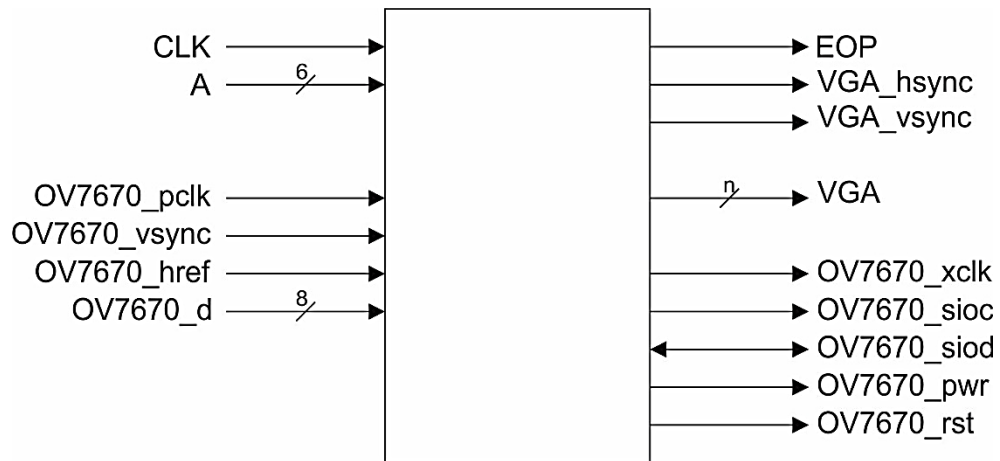


Figura 5. Entidad del proyecto de Procesamiento de Imágenes en Hardware.

La tabla 1 muestra una descripción de las terminales de la entidad.

Tabla 1. Descripción de las terminales de la entidad de procesamiento de imágenes.

Terminal	Descripción
CLK	Señal de reloj.
A	Señal de configuración de exhibición de resultados.
OV7670_pclk	Salida de reloj de pixel desde el sensor.
OV7670_vsync	Activo en alto. Marco válido (frame).
OV7670_href	Activo en alto. Línea de dato válido. Indica pixel activo.
OV7670_d	Salida de datos de pixel.
OV7670_xclk	Reloj maestro del sensor.
OV7670_sioc	Reloj de la interfaz serial de dos hilos.
OV7670_siod	Interfaz de datos I/O serial de dos hilos.
OV7670_pwr	Selección del modo ON/OFF.
OV7670_rst	Reinicialización de la cámara.
VGA_hsync	Sincronía horizontal para el monitor.
VGA_vsync	Sincronía vertical para el monitor.
VGA	Puerto de datos RGB para VGA.
EOP	Fin del procesamiento de imágenes.

La tabla 2 muestra la cantidad de recursos utilizados en un dispositivo Artix, correspondiente a las pruebas en una tarjeta de desarrollo Nexys4.

Tabla 2. Recursos utilizados por el sistema de procesamiento de imágenes.

Recurso	Utilizado	Disponible	%Usado
LUT	688	63400	1.09
LUTRAM	1	19000	0.01
FF	212	126800	0.17
BRAM	104	135	77.04
DSP	1	240	0.42
IO	38	210	18.10
BUFG	4	32	12.50
MMCM	1	6	16.67

4. Resultados.

Utilizando la cámara OV7670, se ha realizado la adquisición de imágenes, las cuales han sido procesadas y almacenadas en hardware. En la figura 4 se muestra el sistema de adquisición y procesamiento de imágenes, con la cámara instalada, y su conexión a un monitor utilizando la interfaz VGA. Se utiliza el sistema de desarrollo Nexys4 de Xilinx.

Los algoritmos básicos que han sido considerados para su implementación, es la conversión de imágenes a color a escala de grises, así como imágenes binarias.

En la figura 5 se muestran resultados de procesamiento básico de imágenes. La figura 7(a) muestra la adquisición de la imagen. La figura 5(b) muestra la imagen convertida a escala de grises. La figura 5(c) muestra la imagen binarizada. Finalmente, se muestra el sistema completo.

Se considera algoritmos más complejos para ser implementados, los cuales a su vez han sido o son temas de tesis de doctorado. El primer algoritmo considerado es flujo óptico, el cual puede ser utilizado en sistemas de vigilancia como detectores de movimiento y seguimiento de objetos. El segundo algoritmo, es para eliminar defectos en las imágenes, ocasionado por contaminación

atmosférica, o bien, el efecto de neblina. Los algoritmos de dehazing pueden ser aprovechados en sistemas de vigilancia, robótica móvil, entre otros.

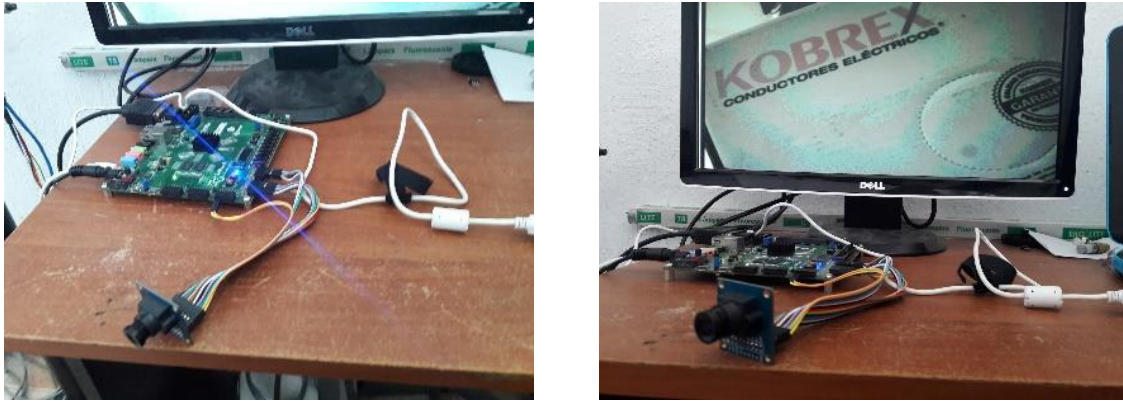


Figura 6. Sistema de adquisición y procesamiento de imágenes basado en Nexys4.

Los resultados obtenidos son satisfactorios, y pueden ser escalados. La tarjeta Nexys2 maneja una interfaz VGA de 10 bit, lo que limita la calidad de la imagen al momento de la exhibición. En comparación, la tarjeta Nexys4 maneja 12 bits en la interfaz VGA. Sin embargo, el procesamiento de los valores de cada pixel es realizado en todos los casos en 24 bits para las imágenes a color, y de 8 bits para imágenes en escala de grises y binaria. Por esto, el procesamiento es estándar y solamente se adapta la interfaz de salida a VGA de acuerdo a la interfaz de salida disponible.

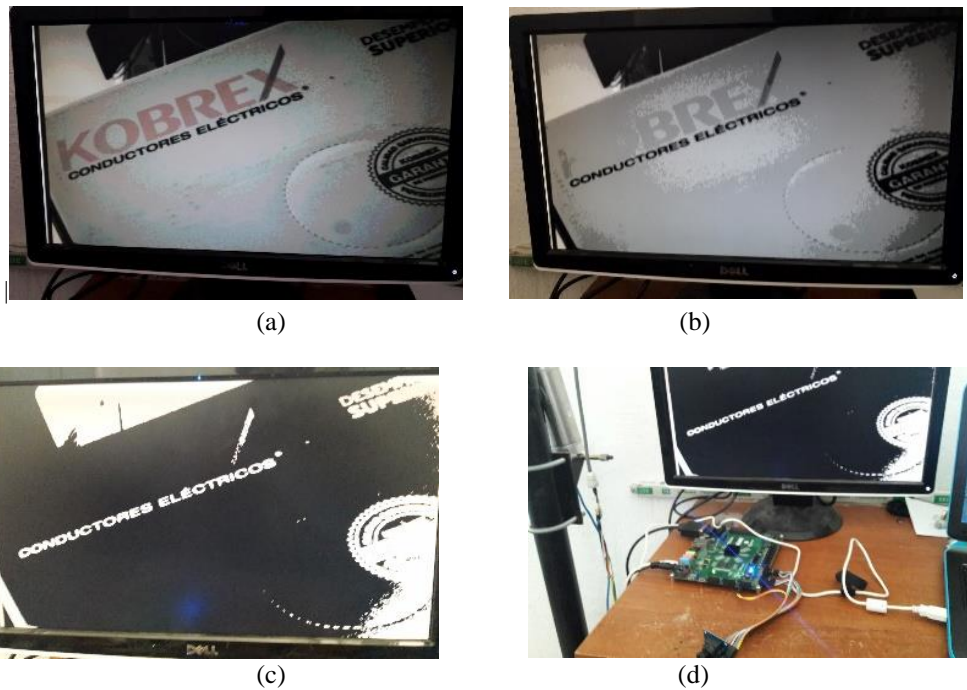


Figura 7. Resultados del procesamiento de imágenes en hardware. (a) Imagen a color. (b) Escala de grises. (c) Imagen binaria. (d) Vista completa del sistema.

5. Conclusiones.

Este trabajo muestra una manera de poder realizar procesamiento de imágenes, utilizando Dispositivos Lógicos Programables. Para esto, se ha utilizado la tarjeta de desarrollo Nexys 2, manufacturada por Xilinx, y utiliza el dispositivo FPGA de la familia Spartan-3E. Así mismo, se realizaron pruebas en un sistema de desarrollo Nexys4, que usa un dispositivo FPGA de la familia Artix, modelo XC7A100T-CSG324. El sistema resulta apropiado, pues aún cuenta con el espacio suficiente para la implementación de hardware adicional. Por esto, el sistema resulta apropiado para hacerlo crecer en capacidad de procesamiento.

Referencias

- [1] Baumstark Jr, L., & Wills, L. (2002). Exposing data-level parallelism in sequential image processing algorithms. In *Reverse Engineering, 2002. Proceedings. Ninth Working Conference on* (pp. 245-254). IEEE.
- [2] Sérot, J., Berry, F., & Ahmed, S. (2011, September). Implementing stream-processing applications on FPGAs: a DSL-based approach. In *Field Programmable Logic and Applications (FPL), 2011 International Conference on* (pp. 130-137). IEEE.
- [3] Li, C. T., & Chen, W. H. (2012). A Novel FPGA-based Hand Gesture Recognition System. *Journal of Convergence Information Technology*, 7(9).
- [4] Sérot, J., Berry, F., & Ahmed, S. (2013). CAPH: a language for implementing stream-processing applications on FPGAs. In *Embedded Systems Design with FPGAs* (pp. 201-224). Springer New York.
- [5] Pandey, J. G., Karmakar, A., Shekhar, C., & Gurunayanan, S. (2014, July). Architectures and algorithms for image and video processing using FPGA-based platform. In *VLSI Design and Test, 18th International Symposium on* (pp. 1-1). IEEE.
- [6] Ashir, A. M., Ata, A., & Salman, M. S. (2014, May). FPGA-based image processing system for Quality Control and Palletization applications. In *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on* (pp. 285-290). IEEE.
- [7] Nagy, Z., Zarándy, Á., Kiss, A., Németh, M., & Zsedrovics, T. (2014). FPGA implementation of a foveal image processing system for UAV applications.
- [8] Balaji, V., & Krishnaveni, R. (2014, February). FPGA based low complexity multipurpose reconfigurable image processor. In *Information Communication and Embedded Systems (ICICES), 2014 International Conference on* (pp. 1-6). IEEE.
- [9] Siddiqui, F. M., Russell, M., Bardak, B., Woods, R., & Rafferty, K. (2014, October). IPPro: FPGA based image processing processor. In *Signal Processing Systems (SiPS), 2014 IEEE Workshop on* (pp. 1-6). IEEE.
- [10] Zhang, W., Shen, L., Page, T., Luo, G., Li, P., Maaß, P., ... & Cong, J. (2015, February). FPGA Acceleration for Simultaneous Image Reconstruction and Segmentation based on the Mumford-Shah Regularization. In *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (pp. 261-261). ACM.
- [11] Hasani, Y., Bodenstorfer, E., Brodersen, J., & Mayer, K. J. (2015, February). FPGA based image processing for optical surface inspection with real time constraints. In *Seventh International Conference on Machine Vision (ICMV 2014)* (pp. 94450P-94450P). International Society for Optics and Photonics.
- [12] Huang, W. T., Tseng, H. L., Dai, J. C., Chen, C. H., & Tan, S. Y. (2015, June). Design of a High-Reliability FPGA Platform for High-Resolution Image Process System. In *Applied Mechanics and Materials* (Vol. 764, pp. 1283-1287).
- [13] Li, L., & Wyrwicz, A. M. (2015). Design of an MR image processing module on an FPGA chip. *Journal of Magnetic Resonance*, 255, 51-58.
- [14] López-Torres C. V., Pedraza-Ortega J. C., Talavera-Velázquez D., Gorrostieta-Hurtado E. & Vargas-Soto J. E. (2017) Análisis de la Perfilometría Wavelet 2D en su Desempeño para la Reconstrucción 3D de Objetos. *La Mecatrónica en México*, 6(2), pp. 36-45.

- [15] Moya-Morales J. C., Pedraza-Ortega J. C., Rodríguez-Reséndiz J. & Ramos-Arreguín J. M. (2017). Propuesta para la detección y corrección de discontinuidades generadas en la digitalización 3D. *La Mecatrónica en México*, 6(1), pp. 9-20.