

9^o Congreso Nacional MECATRONICA

Benemérita Universidad Autónoma de Puebla
Complejo Cultural Universitario
13 - 15 de Octubre, 2010



Puebla

XV Aniversario

de la Facultad de Ciencias de la Electrónica



ISBN: 978-607-95347-2-1



Metodología de una Etapa Básica de un Sistema de Procesamiento de Imágenes Basado en FPGA

Ramos-Arreguín Carlos-Alberto, Moya-Morales Juan-Carlos, Ramos-Arreguín Juan-Manuel, Pedraza-Ortega Jesús Carlos, Canchola-Magdaleno Sandra-Luz, Vargas-Soto José-Emilio.

Facultad de Informática, Universidad Autónoma de Querétaro.

Resumen

En este trabajo se propone una metodología para el manejo de imágenes directamente en hardware, utilizando dispositivos reprogramables. Esto es parte de un proyecto de procesamiento de imágenes en hardware, en la Facultad de Informática de la Universidad Autónoma de Querétaro. La principal contribución de esta metodología es la propuesta de un método para el manejo de imágenes utilizando herramientas de dispositivos lógicos programables y el lenguaje descriptivo de hardware VHDL, utilizando las librerías estándar del IEEE para conservar la portabilidad de la aplicación. Como casos de estudio se muestran imágenes a las que se les aplica un procesamiento específico, y son comparados con la figura original y con resultados obtenidos utilizando Matlab. Las imágenes son almacenadas en una memoria estática/dinámica de acceso aleatorio (SDRAM), la cual es transferida desde una PC vía serial. En este punto del desarrollo del proyecto, se probó la conversión de una imagen a color, a una imagen binario y a escala de grises. Este trabajo da pie para el desarrollo en hardware de un proyecto de reconstrucción tridimensional.

Palabras clave: FPGA, RGB, SDRAM, Metodología

1. Introducción

Desde 1964 hasta nuestros días, el campo de procesamiento de imágenes ha crecido enormemente. Las técnicas de procesamiento se usan ahora para resolver una gran variedad de problemas. Aunque a menudo no relacionados, esos problemas requieren comúnmente métodos capaces de realizar y extraer la información contenida en las imágenes para su interpretación y análisis por parte de los humanos. En cualquier caso se contemplan

tanto técnicas de mejora de la calidad de las imágenes como relativas a la percepción de la máquina. Por otra parte, el desarrollo de equipos cada vez más sofisticados en diversos campos del procesamiento digital de imágenes, han justificado el desarrollo tecnológico alrededor del tratamiento digital de imágenes. A modo de ejemplo, encontramos:

a) *Medicina*, donde los sofisticados equipos PET (**Positrón Emisión Tomography**), resonancia magnética, rayos X, etc., tienen en el tratamiento digital de imágenes un alto grado de importancia en los diagnósticos, los cuales no podrían ser posibles, sin este tipo de procesos.

b) *Observación de la Tierra*, donde los sensores acoplados en los satélites artificiales son capaces de proporcionar imágenes en las que se pueden observar detalles de hasta 0.6 metros de tamaño. Como son la detección de zonas deforestadas, evolución de fenómenos meteorológicos, etc.

c) *Reconstrucción de Objetos 3D*, es el proceso mediante el cual, es posible duplicar las características físicas de un objeto, obteniendo así una réplica del original, existen gran variedad de métodos y técnicas que se utilizan para este fin, una de ellas es a través del procesamiento de imagen donde se digitaliza el objeto y posteriormente se realiza su reconstrucción. El uso del procesamiento de imágenes en la reconstrucción 3D es relativamente nuevo pero su investigación ha avanzado muy rápido en los últimos años [1]. Esta investigación es llevada a cabo en la Facultad.

Hoy en día el uso de dispositivos lógicos programables, en particular los FPGA's (Field Programmable Gate Arrays) [2], ha crecido de una manera muy significativa en el área de software embebido, debido a que son tan eficientes como un ASIC (Application Specific Integrated Circuit) y compiten en costo con los microcontroladores [3]. Cabe mencionar que estos dispositivos tienen la capacidad de procesar una gran cantidad de

información en muy poco tiempo, a comparación de una PC convencional.

Se toma como base la metodología [4] para lograr manejar imágenes en un FPGA, y comenzar a realizar procesamiento de imágenes en hardware, cabe mencionar que los procesos implementados son básicos, y consisten en la conversión de imágenes en color a imágenes en gris, así como a blanco y negro.

En la actualidad, se almacenan imágenes en FPGA en escala de grises, esto si se desea aplicar inmediatamente un filtro a la imagen, o por medio de una herramienta no estándar, lo cual es posible mediante el software MemUtil de Digilent, con el cual se almacena la información de una imagen en una memoria de tipo SDRAM [5]. Así mismo hay aplicaciones ya desarrolladas, las cuáles no utilizan únicamente el estándar de la IEEE, sino que también se basan en librerías del fabricante. Así mismo, hay lenguajes avanzados para realizar operaciones complejas en el FPGA [6, 8], este trabajo se basa en el lenguaje descriptivo VHDL. Se utilizó una tarjeta de desarrollo NEXYS 2 de Digilent, con un circuito SPARTAN 3E, de Xilinx, con número de parte XC3S1200E y una memoria SDRAM Micron con capacidad de almacenamiento de 16 Mbit.

A continuación se menciona la manera cómo se desarrolla la implementación de una imagen en hardware utilizando una memoria SDRAM. Comenzando con una imagen a color a la cual se le aplica el proceso o filtro deseado.

2. Desarrollo

En la actualidad, para realizar el procesamiento de imágenes se utiliza una computadora convencional de propósito general, así como diversas herramientas de software, por ejemplo, MATLAB, C, C++, Java, etc. Por ejemplo, en un procesamiento convencional de imágenes, primero se adquiere la imagen (video, cámara digital, escáner, etc.), enseguida se aplica un determinado filtro a la imagen en MATLAB, y por último la imagen es mostrada en el monitor, como se muestra en la figura 1 [7].

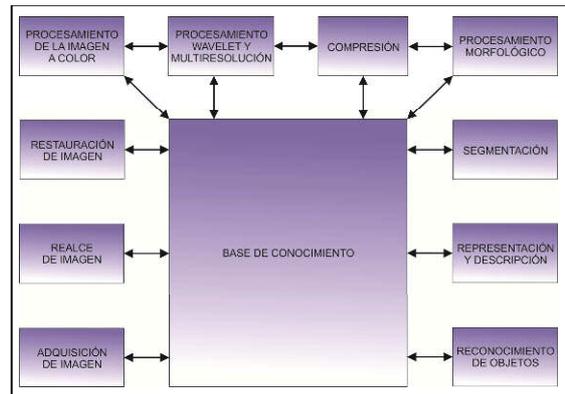


Figura 1. Procesamiento de Imágenes Convencional

El proceso de la figura 2 es aplicado en un sistema embebido basado en FPGA, aprovechando las ventajas de flexibilidad en la implementación de la arquitectura debido a las ventajas mencionadas previamente. En el proceso de la figura 2, se muestra la propuesta para realizar procesamiento de imágenes en hardware, donde la imagen que estamos usando tiene un formato jpg, que puede ser adquirida por cualquier método convencional. Debido a que la tarjeta Nexys 2 utiliza 8 bit para el manejo de video, se utiliza MATLAB para ajustar la resolución de 24 bit de la imagen original, a los 8 bit de la tarjeta. Una vez ajustada la imagen a la resolución de la tarjeta, se transfiere la imagen por medio del puerto serial DB9 de nuestra tarjeta de experimentación, a la memoria SDRAM de la tarjeta utilizando una interfaz digital en el FPGA, y una vez que se tiene la imagen completa en la memoria, se inicia el procesamiento de la imagen. Previo a la exhibición de la imagen en el monitor, es necesario el uso de un módulo de sincronía, el cual se muestra en la figura 4.

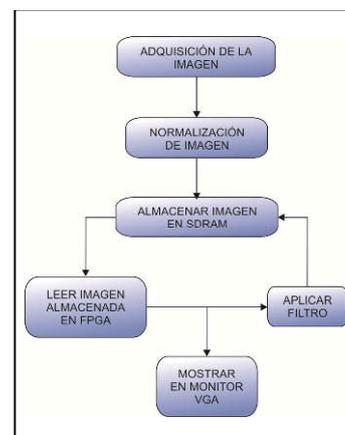


Figura 2. Metodología Propuesta de Procesamiento de Imágenes en Hardware

El proceso de la figura 2, se compone de diversas etapas, por lo que es necesario explicar un poco más cada uno de los bloques mostrados en los siguientes puntos.

2.1 Imagen

Las imágenes digitales obtenidas de una cámara o escáner y que son mostradas en una PC normal, manejan el formato de 3 canales de colores RGB (Rojo, Verde, Azul), lo que significa que para representar el color de un pixel, se utilizan 24 bit, es decir, 8 bit para cada banda de color. Sin embargo, la tarjeta, solamente maneja 8 bit de video, donde 3 son para el rojo, 3 para verde y 2 para azul. Por lo que es necesario ajustar de la imagen para ser procesada con solamente 8 bit. Debido a esto, el negro es (0,0,0); el blanco (7,7,3); rojo (7,0,0); verde (0,7,0) y azul (0,0,3). Como consecuencia, la tarjeta proporciona una resolución muy baja en colores, a comparación de una PC convencional.

2.2 Imagen en Escala de Grises

En una imagen en escala de grises el pixel posee un valor equivalente a una intensidad de gris, que va desde el negro más profundo variando gradualmente la intensidad de gris hasta llegar al blanco. Éstas imágenes se representan con 8 bit para representar cada pixel lo que nos permite únicamente obtener 256 tonalidades diferentes (0 a 255). Cuando se convierte una imagen a color a escala de grises se debe obtener el promedio de las 3 bandas de color de cada pixel. Esta operación se muestra en la ecuación 1.

$$Gray(x, y) = (f(x, y, R) + f(x, y, G) + f(x, y, B))/3 \quad (1)$$

Cabe mencionar que la ecuación (1), sirve para una resolución de color de 8 bit por banda de color. Por lo tanto, la ecuación (1) es adaptada a únicamente obtener la suma total, y para asignar el valor a cada pixel se realiza una LUT (Look Up Table), la cual dependiendo la suma en cada pixel se le asigna su valor ajustado para la tarjeta. Los valores de la LUT se muestran en la tabla 1.

SUMA TOTAL	Valor R, G	Valor B
0	000	00
1	000	00
2	001	00
3	001	00
4	001	00
5	010	00

6	010	01
7	010	01
8	011	01
9	011	01
10	011	01
11	100	01
12	100	10
13	100	10
14	101	10
15	101	10
16	101	10
17	110	10

2.2 Imagen en Blanco y Negro

Si la imagen es en *Blanco y Negro (B/N)*, se almacena un valor por cada píxel, este valor es el nivel de intensidad o nivel de gris comentado anteriormente. Se suele utilizar un rango de valores para su representación, que generalmente es de 0 a 2^{n-1} .

Para obtener una imagen en blanco y negro, debemos obtener primero la imagen en escala de grises, y establecemos un valor umbral, con el cual vamos a definir en qué momento tendremos un valor negro o un valor blanco, este umbral se representa en la ecuación (2).

$$BW(x, y, z) = \begin{cases} 0 & \text{si } Gray(x, y) \leq \lambda \\ 1 & \text{si } Gray(x, y) \geq \lambda \end{cases} \quad (2)$$

Donde $\lambda=127$ para el caso de imágenes con banda de color de 8 bit y $\lambda=4$ para las bandas R y G, y $\lambda=2$ para B, en el caso de la tarjeta utilizada.

2.3 Puerto VGA

El puerto VGA de la tarjeta, tiene cinco señales activas, incluyendo la sincronía horizontal y vertical (HSYNC, VSYNC), y tres señales de video: rojo, verde y azul (RGB), y se encuentran físicamente dirigidas al conector de video tipo D de 15 pines. La señal de video es analógica y el controlador de video utiliza un convertidor digital a analógico, para convertir la salida digital al nivel analógico deseado. En esta tarjeta, las señales de video RGB pueden generar solamente 2^8 colores diferentes. Lo cual, nos da una restricción en la resolución, ya que, los resultados serán de menor calidad, comparado con los resultados obtenidos en una PC.

2.4 Representación en VHDL

Para la implementación del proceso de la figura 3, se utiliza MATLAB, por las facilidades

que presenta en el manejo de las imágenes. El primer paso, es asignar la imagen a una matriz M , la matriz M generada, es de 3 dimensiones, por lo que, en el segundo bloque, el valor obtenido para j es el número de renglones, para k el número de columnas, y para l es de 3, pues la matriz es tridimensional, $M(j,k,l)$.

Una imagen binaria puede ser el resultado de un proceso de umbralización aplicado a una imagen en niveles de gris, usando las siguientes fórmulas:

$$N(j, k, l) = \frac{M(j,k,l)}{255} * 7 \quad (3)$$

Donde $l = 1 \text{ ó } 2$

$$N(j, k, l) = \frac{M(j,k,l)}{255} * 3 \quad (4)$$

Luego obtenemos la matriz N de 8 bit aplicando las fórmulas (3) y (4), tales ecuaciones se muestran en el proceso de la figura 4.

Finalmente, el último bloque representa el envío de los datos a través del puerto serial de una PC, hacia el FPGA, para posteriormente ser almacenados los datos recibidos en la memoria SDRAM.

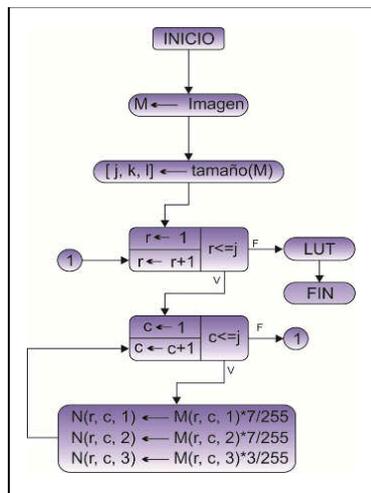


Figura 3. Diagrama de flujo del proceso de normalización de una imagen de 24 bit a 8 bit de color.

En la figura 4 se muestra el diagrama de bloques general del sistema para procesamiento de imágenes.

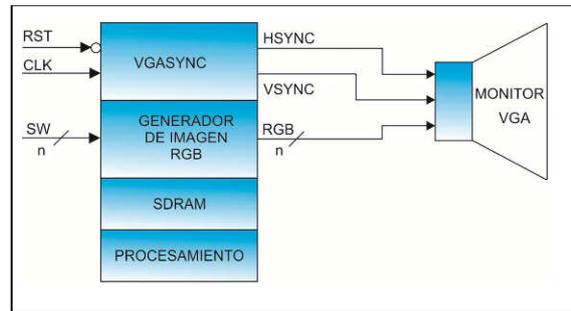


Figura 4. Diagrama de Bloques del Sistema

El circuito VGA-SYNC, genera el tiempo y las señales de sincronización, las señales HSYNC y VSYNC están conectadas al puerto VGA para controlar el barrido horizontal y vertical del monitor. Las dos señales son decodificadas desde los contadores internos las cuales son PIXEL_X y PIXEL_Y, que indican la posición del barrido y esencialmente la localización del pixel actual. El circuito también genera la señal VIDEO_ON, que indica cuando habilitar o deshabilitar el monitor.

El generador imagen RGB, se encarga de generar el color de cada pixel, de acuerdo al valor de la posición actual del pixel a mostrar, lo cual se forma con las coordenadas de las señales (PIXEL_X, PIXEL_Y), así como con el control externo. El control SDRAM nos ayuda a escribir y leer los datos de la imagen de la memoria SDRAM. Por último la parte de procesamiento almacena los procesos a realizar a una imagen.

La tabla 2 nos muestra la combinación de 8 colores básicos que pueden ser generados por el puerto VGA, donde tenemos las 3 bandas de color básicas RGB, y de acuerdo a los bit en 1 se nos genera una cierta intensidad o color en el monitor. Aunque el puerto únicamente permite la generación de 8 colores distintos, se generan hasta 256 intensidades distintas.

Tabla 2. Tabla de colores a 8 bit

ROJO (R)	VERDE (G)	AZUL (B)	COLOR RESULTANTE
000	000	00	NEGRO
000	000	11	AZUL
000	111	00	VERDE
000	111	11	CYAN
111	000	00	ROJO
111	000	11	MAGENTA
111	111	00	AMARILLO
111	111	11	BLANCO

3. Análisis de resultados

La metodología propuesta fue probada utilizando dos imágenes diferentes, tal y como se visualiza en una PC o cualquier otro equipo que maneje los 24 bit. La figura 5 muestra el sistema completo, donde se observa la PC utilizada, la tarjeta de experimentación y el monitor CRT.



Figura 5. Sistema completo utilizado.

Las imágenes a 24 bit utilizadas como casos de estudio son de tamaño 256 x 256 píxeles, y se muestran en la figura 6.



Figura 6. Imágenes utilizadas para las pruebas

La figura 7 muestra el resultado de ajustar la resolución de 24 a 8 bit, y se puede observar que se puede distinguir la semejanza con las imágenes originales.



Figura 7. Imagen normalizada y almacenada en SDRAM

Enseguida, se realizaron diversas pruebas con las imágenes de la figura 7, las cuales fueron conversión a escala de grises y binarización.

Conversión a Escala de Grises. En las figuras 8 a 10, se muestra la conversión a escala de grises realizadas en MATLAB, y se comparan con las imágenes obtenidas con el FPGA.

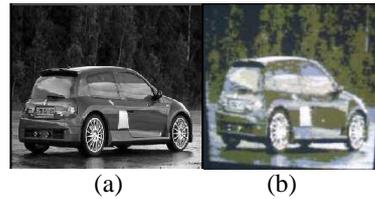


Figura 8: Comparación en escala de grises prueba 1. (a) con 24 bit en Matlab. (b) con 8 bit en el FPGA.

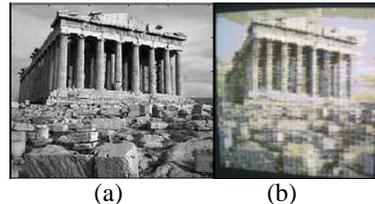


Figura 9. Comparación en escala de grises MATLAB 2. (a) con 24 bit en Matlab. (b) con 8 bit en el FPGA.

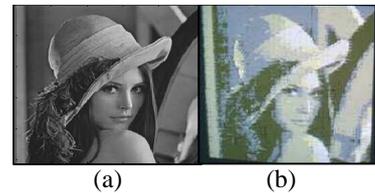


Figura 10. Comparación en escala de grises prueba 3. (a) con 24 bit en Matlab. (b) con 8 bit en el FPGA.

Binarización: Los resultados de esta prueba realizada en MATLAB se muestran en las figuras 11 a 13 para las tres imágenes.

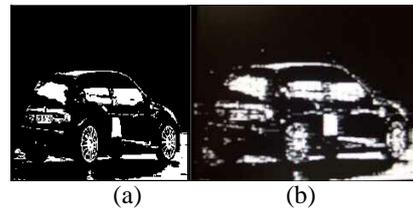


Figura 11: Binarización prueba 1. (a) con 24 bit en Matlab. (b) con 8 bit en el FPGA.

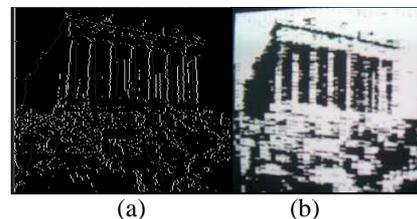


Figura 12. Binarización prueba 2. (a) con 24 bit en Matlab. (b) con 8 bit en el FPGA.

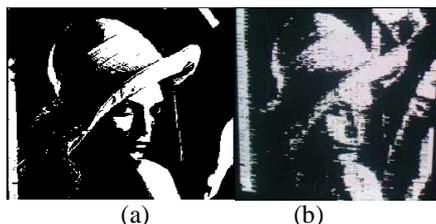


Figura 13. Binarización prueba 3. (a) con 24 bit en Matlab. (b) con 8 bit en el FPGA.

Con las pruebas realizadas, se tiene una metodología funcional, y que permite realizar trabajos más avanzadas, con la ventaja de que las operaciones realizadas tardan nanosegundos en entregar el resultado. Considerando que el FPGA utiliza una frecuencia de reloj de 50 MHz, y que la arquitectura fue diseñada para aprovechar al máximo la velocidad del sistema, de tal manera que el proceso es aplicado a la imagen en un ciclo de reloj para cada pixel. También se desarrolla un sistema base para el procesamiento de imágenes con un mayor poder de cómputo que una PC convencional.

4. Conclusiones

Esta metodología propone una manera de realizar el procesamiento de imágenes en hardware, teniendo altas velocidades de cómputo, y a un bajo costo.

Las diferencias que se encuentran en las figuras 6 a 13, se deben a los cambios de resolución, por lo que es necesario realizar las modificaciones necesarias para aumentar la resolución a 24 bit y obtener los mismos resultados en ambas plataformas.

Los FPGAs, nos permiten crear una arquitectura propia, que nos permita mejorar el desempeño del sistema, con los resultados requeridos por el proyecto principal. Así mismo, debido a la diversidad de trabajo en el procesamiento de imágenes al tener un hardware de arquitectura abierta, nos ayuda en ahorro de costos y podremos obtener una eficiencia mayor por parte del sistema, además de poder darle el debido mantenimiento para hacerlo crecer conforme crezcan las necesidades del proyecto, o bien, que se pueda ajustar a las necesidades de cualquier otro proyecto. Se está utilizando el lenguaje descriptivo de hardware VHDL, respetando el estándar del IEEE, para conservar la portabilidad a cualquier tecnología de FPGAs disponible.

Trabajo a futuro

Se requiere modificar las tarjetas de tal manera que se aumente la resolución de video para mejores

resultados. De la misma manera, se tiene contemplado trabajar en la adquisición de imágenes en tiempo real, así como el procesamiento de las mismas, utilizando las ventajas que nos proporciona. Para esto, es necesario el uso de memorias dinámicas, para almacenar la imagen, y poder aplicar procesamiento de imágenes, como filtros, y obtener resultados de una manera más rápida y dinámica.

Referencias

- [1]. Moya J., González Z, Pedraza J., Soto A., Delgado M., Canchola S., Caracterización de un Sistema de Reconstrucción de Objetos 3D, 6º Congreso Internacional de Ingeniería, Querétaro, Qro., México; 2010.
- [2]. Toledo A., Vicente C., Suardíaz J., Cuenca S.; Xilinx System Generator Based HW Components for Rapid Prototyping of Computer Vision SW/HW Systems; Springer Verlag; pp. 667-674; 2005.
- [3]. Quintero M. Alexander, Vallejo R. Eric; Image Processing Algorithms using FPGA; Revista Colombiana de Tecnologías de Avanzada; Vol. 1; No. 7; 2006; págs. 11 a 16.
- [4]. Ramos C., Cora O., Ramos J., Pedraza J., Canchola S., Vargas J., Metodología para Manejo de Imágenes en FPGA, 6º Congreso Internacional de Ingeniería, Facultad de Ingeniería, UAQ Querétaro, Qro, México, 2010.
- [5]. <http://www.digilentinc.com>
- [6]. <http://procesadodeimagenenfpga.wordpress.com>
- [7]. González R., Woods R; Digital Image Processing; Prentice Hall, USA, 3ª Edición, 2008.
- [8]. Bravo I., Arquitectura Basada en FPGAs para la detección de Objetos en Movimiento Utilizando Visión Artificial y técnicas PCA, Tesis Doctoral: Madrid, España, 2007.