ISSN: 1692-7257 Volumen 1 Número 7 año - 2006

Revista Colombiana de
Tecnologías de Avanzada

IMAGE PROCESSING ALGORITHMS USING FPGA

ALGORITMOS DE PROCESAMIENTO DE IMÁGENES EN FPGA

Ing. Alexander Quintero M, Ph.D Eric Vallejo R.

Universidad del Norte
Km. 5 Antigua vía a Puerto, Barranquilla, Colombia
{aiquintero, evallejo}@uninorte.edu.co

Abstract: This paper deals with the FPGA methodologies used to perform image processing and recognition of objects inside images. In the different implementations it was a high performance, measured in frames per second. The implementations avoided complex mathematical proccesing, but without losing generality. The final result is an image processor whose main obstacle is the iteratives algorithms of labeling and objects identification.

Resumen: Este artículo tiene por objeto, mostrar las metodologías empleadas en una FPGA para realizar el procesamiento de imágenes y el reconocimiento de objetos dentro de dichas imágenes. En las distintas implementaciones se encontró un alto desempeño, medido en cuadros por segundo. Las implementaciones evitaron procesamientos matemáticos complejos, pero sin perder generalidad. El resultado final es un procesador de imágenes cuya principal limitante son los algoritmos iterativos de etiquetado e identificación de objetos.

Keywords: FPGA, VHDL, Artificial vision, Images proccesing, Programable logic

1. INTRODUCCIÓN

Tradicionalmente los sistemas electrónicos se han basado en la utilización de microprocesadores, su facilidad de uso y bajo costo contribuyeron a su popularización a través de los años. A medida que las aplicaciones se hicieron más variadas, surgieron nuevos dispositivos más eficientes para la realización de algunas tareas específicas, como los Procesadores Digitales de Señales (DSP), concebidos para el manejo de tramas largas de datos y operaciones matemáticas complejas o los Microcontroladores diseñados especialmente para aplicaciones embebidas. Sin embargo ninguno de estos dispositivos es capaz de superar la eficiencia en cuanto a procesamiento y consumo de potencia de los Circuitos Integrados de Aplicación Específica (ASIC), que como su nombre lo indica sólo pueden realizar la función para la que fueron construidos.

Adicionalmente, se cuenta con otro tipo de herramientas, que en términos de procesamiento son tan eficientes como un ASIC y que pueden ser reprogramadas como un microprocesador, los Dispositivos Lógicos Programables (PLD).

Actualmente, gracias a los avances en la fabricación de circuitos integrados, estos dispositivos han madurado y compiten en eficiencia con los ASIC y en costo con los microprocesadores, por lo que se han convertido en una opción interesante a la hora de diseñar sistemas que requieran una alta capacidad de procesamiento y un consumo bajo de potencia. (Shimamoto, 2003).

En este documento se presenta la implementación de varios algoritmos de procesamiento de imágenes en una FPGA, orientados hacia visión artificial, los algoritmos comprenden procesamiento de alto y bajo nivel para lograr el reconocimiento de patrones en 2D y fueron desarrollados en VHDL.

2. DESCRIPCIÓN DEL DISEÑO

Los algoritmos de visión artificial, se encuentran por lo general divididos en dos categorías: Algoritmos de bajo nivel, que son aquellos utilizados para adecuar las imágenes adquiridas y los algoritmos de alto nivel que sirven para manipular la información que proveen dichas imágenes, para su posterior interpretación. (Davies, 1997). Recientemente se habla incluso de un nivel intermedio, sin embargo, esto es irrelevante para lo que se desea mostrar en este artículo.

Los algoritmos de visión se caracterizan por ser repetitivos y por manejar grandes cantidades de información, la complejidad matemática de dichos algoritmos no es un problema si la implementación se realiza en un DSP, pero en una FPGA este aspecto se convierte en una dificultad; si se quiere, como en este caso, que todos los algoritmos sean desarrollados a nivel lógico. Esto se debe principalmente a las limitaciones de los sintetizadores lógicos de las herramientas CAD disponibles, para procesar operaciones en punto flotante, por esta razón en la elaboración de los algoritmos se buscaron metodologías que no necesitaran operaciones matemáticas complejas y de esta forma lograr algoritmos eficientes tanto en la velocidad de procesamiento como en la utilización de los recursos de la FPGA. La simplificación de los algoritmos desde el punto de vista matemático, no implica como se podrá ver más adelante, una pérdida de generalización.

Los algoritmos fueron probados con imágenes de 256x256 píxeles e implementados en una FPGA ACEX1K50TC144-2, fabricada por *Altera Corporation* operando a una frecuencia de reloj de 80 MHz. (Quintero & Esmeral, 2003).

3. ALGORITMOS DE BAJO NIVEL

3.1. Filtrado.

Los errores en la adquisición y digitalización de las imágenes, el ruido y las imperfecciones en el lente de la cámara; son factores que introducen discontinuidades en las imágenes y que pueden afectar el desempeño del sistema de visión artificial. Por esta razón se hace necesaria la utilización de una etapa de filtrado.

Hay muchas metodologías para realizar el filtrado de imágenes, la mayoría se basa en el procesamiento local de un píxel y sus vecinos. A través de la convolución espacial del píxel con los factores adecuados se eliminan los errores en la imagen, por lo general este procesamiento requiere una alta carga computacional. (Costa, 2001) Otras metodologías locales se basan en el análisis estadístico de los píxeles en una región, los valores de intensidades que se encuentren muy alejados de la distribución de los demás píxeles son reemplazados con un parámetro estadístico que puede ser entre otros, el promedio, la moda o la mediana.

El cálculo de la mediana en una ventana de 3x3 píxeles, se puede realizar en una FPGA de manera rápida y económica en recursos, aprovechando las líneas de interconexión rápidas entre los bloques lógicos. (Smith, 2003).

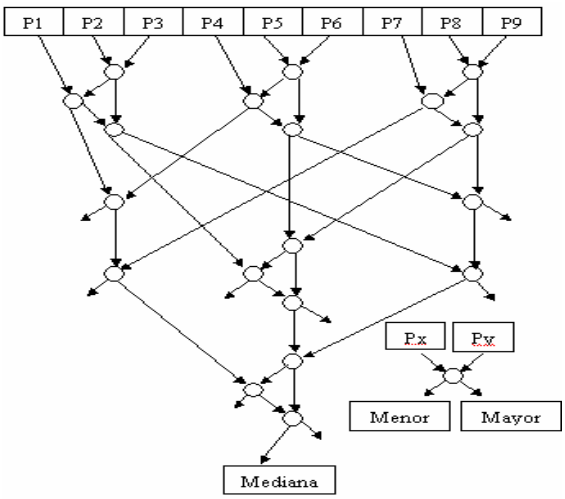

Fig. 1 Estructura del filtro de medianas

Un árbol de comparadores como el que se muestra en la figura 1, es el camino más corto para ordenar 9 píxeles y encontrar el quinto valor, correspondiente a la mediana en la ventana de 3x3. Cada nodo representa un bloque que ordena 2 píxeles, está constituido por un comparador de 8 bits, la señal resultante de la comparación selecciona el puerto de salida para cada dato de entrada.

En la conexión de los bloques comparadores se debe tener en cuenta el retardo de propagación típico de la FPGA, este retardo se acumula y el resultado final debe pasar por un registro. Los datos válidos se obtienen cuando ha pasado un periodo de tiempo equivalente a la suma de los retardos de los comparadores. Esto puede llegar a ser una desventaja ya que el retardo varía dependiendo de la FPGA empleada, por ejemplo para esta implementación el filtrado tuvo un desempeño de 76 fps (*frames per second*).

3.2. Segmentación

Cómo determinar cuál grupo de píxeles hacen parte de un objeto de interés; y cuáles hacen parte del fondo con información secundaria, es lo que se busca a través de los algoritmos de segmentación, este proceso, es el más importante y complejo que se realiza en bajo nivel. La complejidad se debe principalmente a la naturaleza abstracta de la segmentación, el resultado final depende en gran medida de la aplicación final que se busque para el sistema.

La metodología empleada en la FPGA se basó en la distribución de las intensidades y la búsqueda de umbrales locales. Esta aproximación permite que el sistema se ajuste a los cambios en la iluminación dentro de la imagen y lo hace apto para ambientes en los que la luz es poco controlada y su limitación principal es que sólo puede detectar objetos oscuros en un fondo claro. El cálculo de los umbrales locales se logra dividiendo la imagen en sectores más pequeños, en este caso la imagen fue dividida en 64 sectores de 32x32 píxeles cada uno, Algunos autores (Davies, 1997) utilizan solapamiento entre los sectores para evitar las discontinuidades, sin embargo en las pruebas se encontró que este problema se elimina con la corrección de la desviación de los umbrales.

La implementación en la FPGA fue de la siguiente manera: aprovechando uno de los bloques de memoria embebida del chip, se elabora un histograma por cada sector de la imagen, a medida que se construye el histograma la memoria es leída por un comparador y un registro que almacena los máximos encontrados leyendo el histograma de izquierda a derecha (de menor a mayor intensidad) y en sentido contrario. Ver figura 2.

En el mejor de los casos el sector presenta una distribución bimodal como la figura 2, con máximos en las regiones de baja y alta intensidad.

También es posible que la distribución sólo contenga una moda, lo cual significa que el sector hace parte únicamente del fondo o del interior del objeto; y en el peor de los casos el sector contiene más de dos modas, producto de una sombra o un reflejo en el fondo.

Estos casos se traducen en una desviación del umbral del sector, que hasta este punto corresponde al promedio entre las intensidades correspondientes a los máximos encontrados.

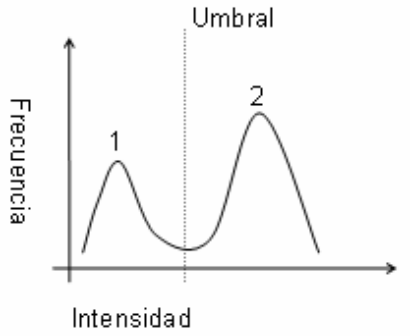

Fig. 2. Histograma idealizado.

Estas situaciones se solucionan aplicando un factor de corrección al umbral, el cual es determinado por la separación entre los máximos 1 y 2, entre mayor sea la diferencia de las intensidades correspondientes a esos máximos, mayor es la probabilidad de desviación del umbral y entre más alto sea el valor del máximo con mayor frecuencia, mayor es la probabilidad de que el umbral se desvíe en ese sentido.

El factor de corrección está almacenado en una tabla de *look-up* (LUT) configurada en otro bloque de memoria embebida dentro del chip. El desempeño de este algoritmo fue de 120 *fps*.

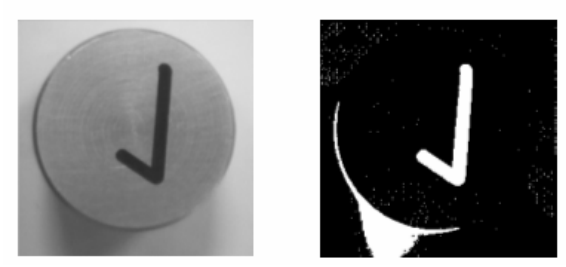

Fig. 3. Una imagen y su respectiva segmentación.

3.3. Filtrado Binario

El proceso de segmentación puede introducir ruido en la imagen, como se aprecia en la figura 3. Este tipo de ruido afecta principalmente el algoritmo de transformada de media abcisa que se emplea posteriormente. La implementación más eficiente en la FPGA es una máquina de estados, que analice la vecindad de un píxel en una ventana de 3x3. La máquina de estados tiene como entrada un *bit*, correspondiente al píxel que se ve de la ventana, y de salida 8 bits correspondientes al valor del píxel después del filtrado. Si el numero de píxeles blancos es mayor o igual a 6 el píxel hace parte del objeto y debe ser blanco (255d) sino hace parte del fondo y debe ser negro (0d), el diagrama de estados de la figura 4 ilustra el algoritmo empleado que tuvo un desempeño de 110 *fps*.

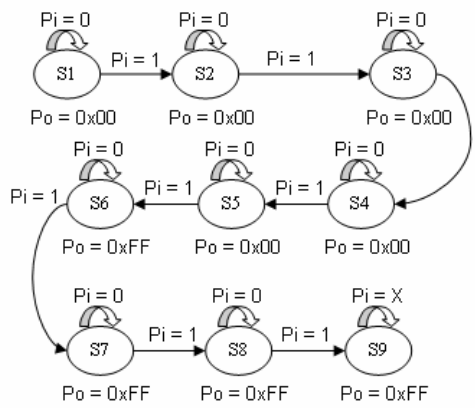

Fig. 3. Diagrama de estados para el filtro binario.

4. ALGORITMOS DE ALTO NIVEL

4.1. Etiquetado

Una vez terminada la segmentación es necesario conocer cuántos objetos fueron detectados, información que resulta útil para un posterior reconocimiento de los objetos encontrados. Gran parte de los algoritmos implementados en hardware para etiquetar objetos se basan en procesamiento paralelo, los que realizan el procesamiento local de los píxeles generalmente asignan unidades de procesamiento a cada píxel; esta arquitectura garantiza un desempeño veloz del algoritmo pero demanda una alta complejidad computacional y un consumo costoso de recursos dentro del chip. (Crookes et al, 1999)

Otras metodologías más propias de los microprocesadores se basan en la conexión entre los píxeles del objeto, se conocen como algoritmos de *Connected Component Labeling (CCL)*. Dichos algoritmos son secuenciales y realizan un procesamiento iterativo sobre la imagen, lo cual se convierte en su mayor desventaja ya que el tiempo de procesamiento depende de la complejidad y el número de objetos que se detecten como resultado de la segmentación.

El algoritmo de CCL implementado en la FPGA consta de dos bloques de procesamiento, el primero está compuesto por un comparador de 1 *bit* que escribe las etiquetas iniciales del objeto, el valor de la etiqueta se decrementa cuando hay un paso de píxeles de objetos al fondo. Posteriormente dos máquinas de estados aplican las máscaras que se muestran en la figura 5 sobre la imagen, en una ventana de 3x3 píxeles.

La máscara 1 se aplica barriendo la imagen de izquierda a derecha y de arriba abajo, la segunda máscara se aplica en el sentido inverso, es decir de derecha a izquierda y de abajo hacia arriba. El objetivo es encontrar el valor del píxel máximo dentro de la máscara, esto se logra a través de un árbol de comparadores parecido al empleado en el filtro por medianas.

El valor del píxel analizado se reemplaza con el máximo encontrado, este procedimiento se repite hasta que no ocurran cambios en la imagen.

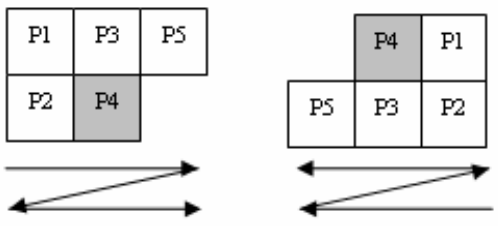

Fig. 4. Máscaras empleadas en el algoritmo de CCL.

4.2. Transformada de media abcisa. Identificación de objetos

El resultado del algoritmo de segmentación entrega una imagen binaria, en la que ha sido eliminada gran parte de la información secundaria de la imagen. Los algoritmos de procesamiento no lineal se ajustan muy bien a este tipo de imágenes y se basan en las matemáticas morfológicas, que se ocupa de las operaciones entre conjuntos. Matemáticamente la imagen se define como un conjunto de píxeles f(x,y) y cada máscara o ventana se conoce como un elemento estructurador g(x,y); de las operaciones entre estos dos elementos depende el resultado del procesamiento.(Costa, 2001)

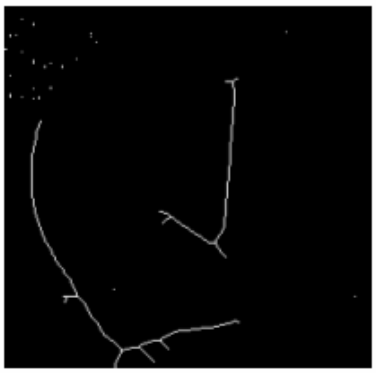

Fig. 5. Imagen adelgazada hasta la meia abcisa.

La transformada de media abcisa es un algoritmo que se basa en matemáticas morfológicas, por medio de él se busca adelgazar el objeto hasta el punto medio entre sus bordes, y se obtiene como resultado una imagen como la que se muestra en la figura 6. Con esta representación se puede describir la imagen a través de las ramas y los nodos o intersecciones de ramas que la conforman. Estas características son invariantes a la orientación o escala de los objetos; sin embargo son altamente sensibles al ruido dentro de la imagen segmentada ya que discontinuidades en el borde de los objetos introducen ramas inexistentes, que cuando se cruzan con las ramas reales del objeto generan nuevos nodos que desvían las medidas de los descriptores del objeto.

La transformada de media abcisa se basa en las operaciones de diltatación y erosión, dos operaciones básicas de las matemáticas morfológicas, utiliza los 2 elementos estructuradotes mostrados en la figura 7, cada uno es rotados 90º en cada barrido de la imagen, la secuencia de rotación depende de la dirección en la que son leídas las imágenes de la memoria y el número de iteraciones depende de la complejidad del objeto y del nivel de ruido presente en la imagen.

En la FPGA no se rotan los elementos estructuradores, en cambio, se emplean 8 elementos cuya secuencia es determinada por una máquina de estados externa. Cada imagen debe ser iterada por lo menos una vez antes de obtener algún resultado. Los elementos estructuradotes, son a su vez 8 máquinas de estados, parecidas al filtro binario, si la secuencia del bit de entrada permite llegar al estado s9, el píxel se encuentra en alguno de los bordes del objeto y debe ser borrado, sino no se cambia su valor.

Fig. 6. Elementos estructuradotes de la transformada de media abcisa.

5. MANEJO DE MEMORIA

Las imágenes que procesa un sistema de visión artificial, en la mayoría de los casos se encuentran almacenadas en una memoria, el desempeño de los algoritmos empleados depende en gran medida de la eficiencia del procesador para hacer el direccionamiento a la memoria. En una FPGA, el direccionamiento se puede llevar a cabo sobre medida, esto quiere decir que se puede configurar el sistema de direccionamiento que más convenga para realizar la tarea deseada.

En la implementación que se trata en este artículo se diseñó un bloque manejador de los 512 KB de memoria externa con la que cuenta el sistema, este bloque es independiente de los algoritmos de procesamiento, para hacerlo reutilizable por varios algoritmos. Los 3 *bits* más significativos del bus de direcciones son empleados como selectores de bancos de memoria, así se obtienen 8 bancos de 64 KB cada uno; esto permite que en cada banco de memoria se almacene una imagen completa de 256 x 256 píxeles.

El direccionamiento de cada banco de memoria lo realizan dos contadores de 16 bits, uno para incrementar las direcciones y otro para decrementarlas, cada uno cuenta con carga en paralelo, una señal de habilitación y una entrada de reloj. La salida de los contadores se divide en dos buses de 8 bits, la parte baja corresponde a las columnas y la parte alta a las filas dentro de la imagen.

P1	P4	P7
P2	P5	P8
P3	P6	P9

Fig. 7. Un píxel y su vecindad.

La figura 7 muestra la ventana de 3 x 3 píxeles, este tipo de direccionamiento entrega las direcciones de un píxel central sobre el cual se realiza el procesamiento y su vecindad, es decir todos los píxeles que tienen contacto con él.

La principal dificultad a superar en este tipo de direccionamiento sucede en los bordes de la imagen, ya que en esos lugares no se puede apuntar a la ventana completa, la solución que ofreció los mejores resultados para superar este problema fue la de repetir el píxel central en los lugares en los que no se puede reproducir la ventana, por ejemplo en el borde superior izquierdo de la imagen los píxeles correspondientes a las posiciones P1, P2, P3, P4 y P7 no pueden ser leídos porque la dirección del píxel central es 0000h, en ese caso las posiciones son reemplazadas con el valor almacenado en la dirección 0000h.

En la FPGA el apuntador de la ventana de 3 x 3, es una máquina de 9 estados en los que se suma o se resta 1 a la posición de memoria de entrada, si el resultado de una suma es 00h, el píxel se encuentra en un borde derecho o inferior y el valor se reemplaza con el píxel central, si el resultado de una resta es FFh el píxel se encuentra en un borde superior o izquierdo y se reemplaza con el mismo valor.

6. CONCLUSIONES

En las páginas anteriores se mostró el desempeño y las implementaciones de varios algoritmos de procesamiento de imágenes en una FPGA. El resultado final de dichas implementaciones es un procesador de imágenes, este procesador junto con un sistema monitor o maestro puede ser empleado en aplicaciones embebidas, liberando a un procesador central de las extenuantes tareas que requiere un sistema de visión artificial.

Con excepción de las metodologías iterativas, los algoritmos implementados son aptos para ser empleados en sistemas que requieran información en tiempo real. La principal ventaja que ofrecen los dispositivos de lógica programable sobre los microprocesadores es la gran capacidad con la que cuenta el diseñador para crear los circuitos específicos que pueden realizar las tareas de la manera más eficiente, esto se pudo apreciar principalmente en el manejo de la memoria, gran parte del desempeño de los algoritmos depende de la velocidad con la que se puedan generar las direcciones de memoria.

REFERENCIAS

Shimamoto, Brion; Tredennick, Nick. Go Reconfigure. IEEE Spectrum. Volumen 40, No. 12, Diciembre 2003. P. 36.

Davies, E. R. Machine Vision: Theory, Algorithms, Practicalities. 2ed. Londres: Academic Press, 1997. 750 p.

Esmeral, Mario; Quintero, Alexander. Sistema de Visión Artificial Para Plataforma Móvil, Barranquilla 2003. Proyecto de Grado (Ingeniero Electrónico). Universidad del Norte. Departamento de Ingeniería Electrónica.

Costa, Luciano da Fontoura; Marcondes, Roberto. Shape Analysis and Classification: Theory and Practice. Florida: CRC Press. 2001.

Smith, Jhon. Implementing Median Filters in XC4000E FPGAs. Univision Technologies Inc. Billerica Massachusetts.

Crookes, D; Benkrid, K. An FPGA Implementation of Image Component Labeling. The Queen's University of Belfast.

Suzuki, Kenji; Horiba, Isao; Sugie, Noboru. Linear-Time Connected-Component Labeling Based on Sequential Local Operations. Computer Vision and Image Understanding No. 89. Octubre 2003. P. 1.

Torsello, Andrea; Hancock, Edwin R. A Skeletal Measure of 2D Shape Similarity. Computer Vision and Image Understanding No. 95. Mayo 2004. P. 1. Disponible e