

CORRELATION COPROCESSOR BASED ON FPGA

COPROCESADOR DE CORRELACIÓN BASADO EN FPGA

Miguel O. Arias Estrada¹, Francisco J. Enríquez Aguilera^{1,2},
Ricardo E. Pérez Blanco².

¹INAOE, Coordinación de Ciencias Computacionales.

²UACJ, Departamento de Eléctrica y Computación . ariasm@inaoep.mx ,
fenrique@uacj.mx, rblanco@uacj.mx

Abstract: In this work, a hardware-architecture for correlation algorithm implementation is presented. The proposed architecture allows real time tracking. For the design the hardware description language VHDL (Very High Speed Integrated Circuit Hardware Description Language) was used, it was implemented on a FPGA (Field Programmable Gate Arrays) Spartan 2.

Resumen: En este trabajo se plantea una arquitectura hardware para la implementación del algoritmo de correlación. La principal característica de la arquitectura propuesta, es que permite efectuar el procesamiento de seguimiento en tiempo real. Para el diseño se utiliza el lenguaje de descripción de hardware VHDL, así mismo la implementación se lleva a cabo en un FPGA Spartan 2.

Keywords: VHDL, FPGA, Vision, hardware y pixel.

1. INTRODUCCIÓN

La visión por Computadora ha basado sus desarrollos en el estudio del sistema visual humano, el cual sugiere la existencia de diferentes tipos de tratamiento de la información visual, dependiendo de metas u objetivos específicos; es decir, la información visual percibida es procesada en distintas formas con base en las características particulares de la tarea a realizar, por lo que la visión por computadora propone varias técnicas que permiten obtener una representación del mundo a partir del análisis de imágenes obtenidas desde cámaras de video.

Debido a que la información visual es una de las principales fuentes de datos del mundo real, resulta útil el proveer a una computadora digital del "sentido de la vista" (a partir de imágenes tomadas con cámaras digitales o analógicas), que junto con otros mecanismos como el aprendizaje, hagan de esta, una herramienta capaz de detectar y ubicar objetos en el mundo real, objetivo principal de la Visión por Computadora. Los dispositivos reconfigurables permiten a los diseñadores construir todo o parte de su diseño en hardware en vez de software (Swan, 1999).

Exportando funcionalidad al hardware se pueden conseguir significativas mejoras de velocidad, porque esa funcionalidad no tiene que dividirse en instrucciones individuales que el CPU transporte, decodifique, etc.

Esa funcionalidad se convierte en hardware. La capacidad de implementar una aplicación en hardware proporciona la oportunidad de explotar la concurrencia inherente de los circuitos digitales. Es decir, el dispositivo puede configurarse o dividirse en múltiples subsistemas, todos los cuales podrían funcionar concurrentemente con los demás.

2. SEGUIMIENTO POR CORRELACIÓN

El Algoritmo de Seguimiento por correlación de área, es una técnica efectiva para el seguimiento de objetos que no tienen características sobresalientes, ni un alto contraste con el fondo de la imagen, además el objetivo puede ser un área de interés sobre la imagen a seguir. La correlación por área consiste en superponer una imagen de referencia o patrón sobre una imagen de búsqueda y calcular una función de similaridad (Sankar K. and Veerabhadra, 2001). La imagen dentro de la ventana de búsqueda tiene máxima similaridad con la imagen de referencia, cuando es superpuesta sobre la localización del objetivo. Para evitar algunos problemas como son: cambios en el tamaño del objetivo, diferente ángulo de vista, variaciones de contraste, cambios en la forma y textura del objetivo, etc., la imagen de referencia se actualiza periódicamente.

La función de correlación normalizada [2] es:

$$C_{xy} = \frac{\sum_{m=1}^O \sum_{n=1}^P \text{ven_pat}(m,n) * \text{ven_bus}(i,j)}{\left[\sum_{m=1}^O \sum_{n=1}^P \text{ven_pat}^2(m,n) * \sum_{m=1}^O \sum_{n=1}^P \text{ven_bus}^2(i,j) \right]^{1/2}} \quad (1)$$

Donde ven_bus es la imagen de búsqueda con tamaño de $M * N$, ven_pat la imagen de referencia con tamaño de $O * P$, donde $O < M$ y $P < N$. Los datos a comparar de ven_bus inician en el píxel (i,j) donde $i = x + m - 1$ y $j = y + n - 1$. Además $1 \leq x \leq M - O + 1$ y $1 \leq y \leq N - P + 1$.

Las imágenes son en escala de gris con 256 niveles, por lo que cada píxel toma un valor comprendido en el rango de:

$$0 \leq \text{ven_bus}(M,N) \leq 255 \text{ y} \\ 0 \leq \text{ven_pat}(O,P) \leq 255.$$

El máximo valor de C_{MN} proporciona la localización del punto de la imagen más parecida a la de referencia, el cual tiende entre 0 y 1. La normalización es requerida para evitar falsos picos, aún cuando la actual referencia esté presente en la imagen de búsqueda sin alguna distorsión o ruido (Swan, 1999).

La Ec. 1 se utiliza para comparar valores de correlación entre sub-imagenes de la ventana de búsqueda y la ventana patrón. Ahora, si p es el número de veces que se va a realizar la operación de la ventana patrón sobre la de búsqueda (donde $p = x * y$) y Correl_may es el dato almacenado con la correlación más alta, encontramos que la condición a utilizar para encontrar la mayor similitud es:

Si
 $\text{Correl}(x,y)_p > \text{Correl_may}$

Entonces

$$\text{Correl_may}(x,y) = \text{Correl}(x,y)_p$$

Fin.

Dado que en la comparación, el operando

$$\left[\sum_{m=1}^O \sum_{n=1}^P \text{ven_pat}^2(m,n) \right]^{1/2} \quad (2)$$

existe en ambos lados de la ecuación y corresponde a un mismo valor, podemos excluirlo de los cálculos.

Ya que algunas operaciones matemáticas resultan más costosas en recursos hardware, como son la división y la raíz cuadrada, se opta por sustituirlas a multiplicaciones, las cuales son más económicas en cuestión de clbs. Si separamos el numerador (Ec. 3) y el divisor (Ec. 4)

$$\text{Num_corr}(x,y) = \sum_{m=1}^O \sum_{n=1}^P \text{ven_pat}(m,n) * \text{ven_bus}(i,j) \quad (3)$$

$$\text{Den_corr}(x,y) = \left[\sum_{m=0}^O \sum_{n=0}^P \text{ven_bus}^2(i,j) \right]^{1/2} \quad (4)$$

La comparación sería:

$$\frac{\text{Num_corr}(x,y)_p}{\text{Den_corr}(x,y)_p} > \frac{\text{Num_corr_may}(x,y)}{\text{Den_corr_may}(x,y)} \quad (5)$$

Que es similar a:

$$\text{Num_corr}(x,y)_p * \text{Den_corr_may}(x,y) > \\ \text{Num_corr_may}(x,y) * \text{Den_corr}(x,y)_p \quad (6)$$

En la Ec. 6 se puede observar que ya no es necesario realizar operaciones de división; para eliminar la raíz cuadrática solo se elevan al cuadrado todos los términos.

3. IMPLEMENTACIÓN DEL ALGORITMO DE CORRELACION

Mediante Matlab[®] se implementó el algoritmo de correlación para seguimiento de características de una imagen. El tamaño de la ventana patrón es de 32 * 32 píxeles y el de la ventana de búsqueda de 64 * 64 píxeles.

El hecho de utilizar multiplicaciones da como resultado un número con una gran cantidad de bits, el cual es un problema para la implementación en hardware (debido a que ocupan un mayor espacio de silicio); por lo que para evitar dicho problema se truncan los bits menos significativos (LSB) y a continuación se realiza un análisis de los resultados a diferentes niveles de truncamiento de datos.

Las figuras 1 y 2 muestran los diagramas del comportamiento de seguimiento de una característica, en las cuales se grafican las coordenadas X e Y, de la variación de movimiento de la característica seleccionada; para las secuencias Barco (90 imágenes) y Marina (57 imágenes), respectivamente; utilizando el algoritmo: a) con todos los bits, b) con truncamiento de los 12 LSB's, c) con truncamiento de los 16 LSB's. Se observa que la diferencia en resultados es mínima entre las gráficas con el algoritmo completo y el algoritmo con truncamiento de los 12 LSB's, mientras que con el truncamiento de los 16 LSB's es mayor la diferencia.

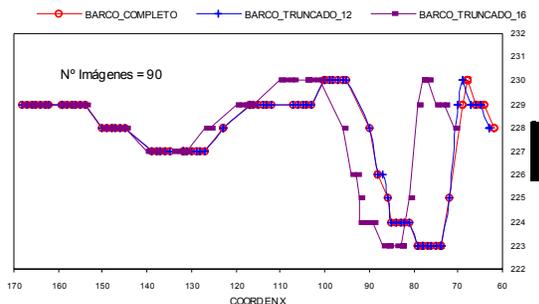


Fig. 1. Comportamiento en el seguimiento de imágenes del algoritmo a) completo, b) con truncamiento de datos de 12 bits y c) truncamiento de datos de 16 bits, para la secuencia de imágenes Barco.

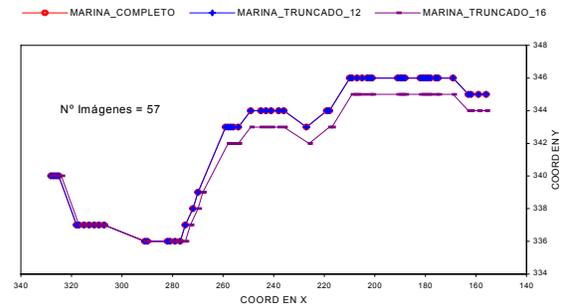


Fig. 2. Comportamiento en el seguimiento de imágenes del algoritmo a) completo, b) con truncamiento de datos de 12 bits y c) truncamiento de datos de 16 bits, para la secuencia de imágenes Marina.

En la Figura 3 se muestran la primer y última imagen de las secuencias Barco y Marina respectivamente, realizando el seguimiento de las características seleccionadas mediante el algoritmo de correlación. La característica está centrada en los cuadros, siendo el cuadro más pequeño la imagen patrón y el más grande la imagen de búsqueda.



a)



b)

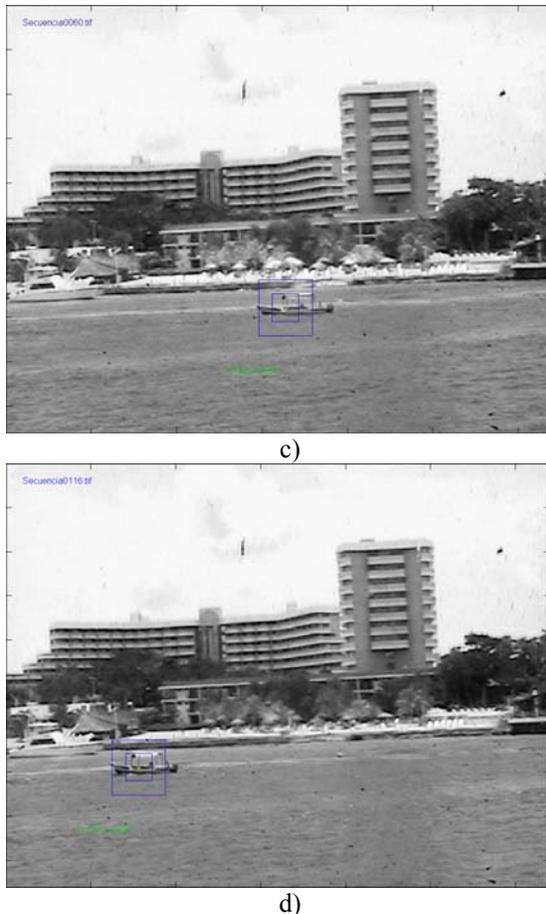


Fig. 3. Ejemplo del seguimiento de características para las secuencias Barco y Marina: a) imagen #1 de la secuencia Barco, b) imagen #90 de la secuencia Barco, c) imagen #1 de la secuencia Marina y d) imagen #57 de la secuencia Marina.

4. ARQUITECTURA PLANTEADA PARA EL ALGORITMO DE CORRELACIÓN

Debido a los resultados obtenidos en el truncamiento de datos, se optó por truncar los 12 LSB's, por lo que se presenta un diagrama a bloques de la arquitectura a implementar. En la Figura 4 se observa que los datos entran por el bus de entrada/salida, utilizando buffers de tres estados, también se cuenta con memorias para el almacenamiento de la ventana patrón y la de búsqueda, un bloque de control, el cual como su nombre lo indica se encarga de controlar los accesos de entrada / salida a memoria, un bloque de registro para el control del proceso y por último el procesador de correlación el cual se encarga de realizar el procesamiento de correlación.

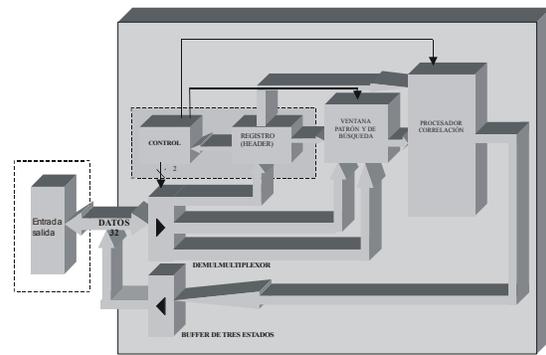


Fig. 4. Diagrama a bloques de la Arquitectura propuesta para la implementación del Algoritmo de correlación.

En la figura 5 se muestra como esta constituido internamente el bloque del procesador de correlación. Se constituye internamente por multiplicadores-acumuladores básicamente, que se encargan de las operaciones sobre los píxeles, que es cuando se tiene que acceder los módulos de memoria. Una vez realizados los cálculos sobre todos los datos, se requieren efectuar operaciones de multiplicación, las cuales no requieren de accesos a memoria ya que dependen de los resultados de los bloques anteriores. Por lo anterior se utiliza un multiplicador serie-paralelo; el cual tiene la característica de que cuenta con una latencia igual al número de bits de los datos a procesar (cabe mencionar que los datos están comprendidos en un mismo rango de bits, debido al truncamiento). Por ejemplo si los datos de entrada están constituidos por 14 bits, se requerirán 14 ciclos de reloj para obtener el resultado.

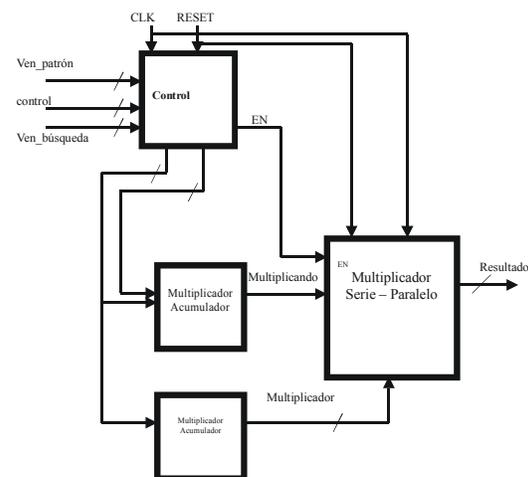


Fig. 5. Diagrama a bloques del procesador de correlación a utilizar.

En la figura 6 se muestra el ruteo interno del FPGA para 2 procesadores de correlación. La arquitectura de estos dos procesadores se codificó en VHDL y sintetizó para un FPGA Spartan II XC2S150FG456, mediante la herramienta de síntesis Xilinx ISE 4.1i. Este diseño tiene una frecuencia de operación de 55.2 MHz, utiliza 52% del espacio total del FPGA, puede procesar 8 píxeles por ciclo de reloj, tiene una latencia de 139392 ciclos y puede procesar una imagen en 2.52ms.

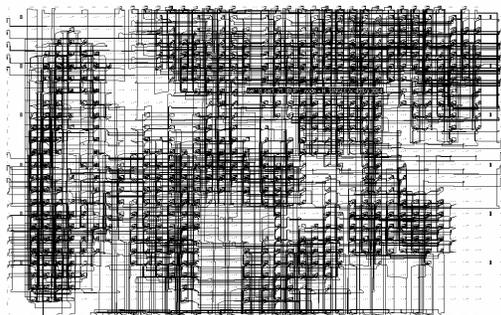


Fig. 6. Ruteo interno del FPGA de 2 procesadores de correlación.

5. CONCLUSIONES

Sobre los resultados presentados de los procesadores implementados, se observa que supera el requisito de procesar 30 imágenes por segundo con lo cual se permite el seguimiento de múltiples blancos en tiempo real.

REFERENCIAS

- Hernández B. José F. (2000), “Sistema para la navegación visual de un helicóptero”, http://www.elai.upm.es/spain/Investiga/GCII/publicaciones/pub00/Doc001_00.pdf,
- Swan R., Wyatt A. (1999), “Computación Reconfigurable”, <http://www.acm.org/crossroads/espanol/xrds5-3/ntu.html>,
- Sankar M K. and Veerabhadra Rao K, “Robust Correlation Tracker”, Sadhana, Vol. 26, Part 3, June 2001, pp. 227-236.
- Lewis J. P. (1995), “Fast Template Matching”, Vision Interface, p. 120-123,.
- Chatfield Christopher. (1979), “Statistics for technology”, Chapman and Hall, Great Britain,.