

Implementación en hardware reconfigurable de un modelo de retina

Antonio Martínez, Samuel Romero, Eduardo Ros, Alberto Prieto y Francisco J. Pelayo

Resumen— Se presentan los resultados obtenidos hasta el momento en el desarrollo de una plataforma software-hardware apropiada para el estudio y experimentación con distintos esquemas de procesamiento bioinspirado de información visual (retinas artificiales). La implementación de los primeros prototipos se ha llevado a cabo en hardware reconfigurable, haciendo uso de la placa RC1000 equipada con la FPGA Xilinx Virtex-E 2000eBG560.

La descripción funcional de diversos módulos del sistema de procesamiento de información visual ha sido elaborada en Handel-C, comprobándose la facilidad de este lenguaje para trasladar algoritmos típicos de procesamiento de imágenes a una descripción sintetizable a nivel de puertas lógicas. Por otra parte, también se está evaluando el System Generator de Xilinx para la síntesis de filtros espacio-temporales de un modelo neuromórfico de retina.

La plataforma Hw-Sw planteada debe finalmente permitir una experimentación ágil con diversos modelos particularizados por el usuario, y que han de ejecutarse en tiempo real en la RC1000 empleada como co-procesadora reconfigurable.

Palabras clave— Retina artificial, procesamiento bioinspirado, visión, hardware reconfigurable, prótesis visuales.

I. INTRODUCCIÓN. EL PROYECTO CORTIVIS

Este trabajo forma parte del proyecto europeo CORTIVIS (Cortical Visual Neuroprosthesis for the Blind, QLK6-CT-2001-00279), dentro del programa europeo “*Quality of Life and Management of Living Resources*”, iniciado en enero de 2002. En este proyecto colaboran neurofisiólogos, médicos, físicos e ingenieros electrónicos e informáticos de 8 instituciones europeas, incluida una SME.

El consorcio CORTIVIS tiene como principal objetivo desarrollar una neuro-prótesis visual que actúe a nivel cortical, destinada principalmente a personas que hayan sufrido una pérdida total de la visión. El problema planteado es complejo y en la actualidad existe un gran número de laboratorios en todo el mundo trabajando en direcciones similares, evaluando prótesis que actúen a nivel de distintos centros del camino visual (retina, nervio óptico, y corteza visual primaria). A diferencia de

las prótesis cocleares, que han alcanzado un elevado grado de éxito, siendo actualmente una opción quirúrgica muy válida para la rehabilitación de muchos casos de sordera, las prótesis visuales tienen ciertas dificultades añadidas que han provocado un retraso en su desarrollo. Estas dificultades son tanto de tipo técnico, motivadas fundamentalmente por el elevado número de electrodos de estimulación necesarios para producir una percepción visual con una resolución útil, como de carácter quirúrgico, ya que deben implantarse en centros más próximos al cerebro y por tanto más inaccesibles y con mayores riesgos en casos de bio-incompatibilidad.

En definitiva, nos enfrentamos con un reto difícil de superar pero en el que pequeños avances suponen una gran esperanza para las personas (cada vez más numerosas, debido a accidentes y al envejecimiento progresivo de la población) que pierden la visión, que es el órgano sensorial que más información nos aporta en la vida cotidiana. Deberá transcurrir algún tiempo, puede que entre 5 y 10 años, para alcanzar en este campo una madurez similar a la que se tiene actualmente en implantes cocleares.

Entre las tareas planteadas en CORTIVIS se encuentra la implementación de un sistema de procesamiento bioinspirado de información visual, que funcione en tiempo real (esto es, a una velocidad tal que pueda finalmente proporcionar una percepción visual casi continua), y que sea capaz de producir estímulos nerviosos similares a los recibidos en la corteza visual primaria de una persona vidente normal. Este sistema será inicialmente empleado para experimentación y desarrollo del resto de la electrónica asociada con la prótesis, pero finalmente debe poderse integrar en un chip específico, particularizable para cada paciente. Por este motivo es esencial generar descripciones hardware reutilizables, que inicialmente se proyecten en lógica reconfigurable, pero que en su momento puedan fácilmente sintetizarse para una tecnología de ASIC.

La figura 1 muestra esquemáticamente la organización del sistema completo, que incluye un bloque de procesamiento bioinspirado de la información visual (modelo de retina), un bloque de codificación de la salida retiniana en forma de impulsos que puedan producirse con ciertas cadencias y con desfases temporales concretos, y cuya proyección sobre la matriz de electrodos de estimulación sea totalmente reconfigurable. Aunque se dice que la proyección de imagen desde la retina hasta la corteza visual primaria es “retino-tópica”, esto significa que dos puntos próximos

A.Martínez y S.Romero son investigadores del proyecto CORTIVIS, que se desarrolla en la Facultad de Ciencias de Granada. E.Ros, A.Prieto y F. Pelayo son Profesores del Departamento de Arquitectura y Tecnología de Computadores, con sede en la ETS de Ingeniería Informática, 18071-Granada. E-mail de contacto: fpelayo@ugr.es.

en la escena visual proyectan en posiciones vecinas de la corteza visual, pero ésta tiene pliegues e irregularidades que pueden hacer que la proyección de una imagen plana sufra múltiples deformaciones [1]. Estas deformaciones, dependientes del paciente y de la zona concreta del posible implante, pueden ser corregidas artificialmente mediante una asignación oportuna de las direcciones de destino de los pulsos procedentes de la retina artificial.

Finalmente debe desarrollarse un sistema integrado de telemetría que permita transmitir los impulsos eléctricos al estimulador de los electrodos. En principio, las experiencias previstas harán uso de la matriz de microelectrodos de Utah [2] (figura 1), si bien son posibles otras matrices que puedan proporcionar una especificidad similar en la estimulación (la matriz de microelectrodos de Utah emplea microelectrodos de silicio con una separación de unos 400 μm , similar a la existente entre columnas corticales en humanos).

Retina artificial

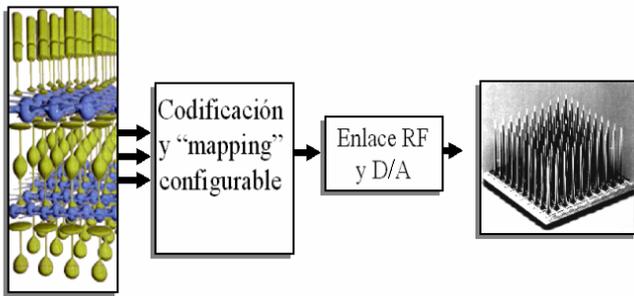


Fig. 1. Diagrama de bloques del esquema completo de procesamiento asociado a la prótesis visual en CORTIVIS.

El trabajo aquí descrito se refiere a la implementación en hardware reconfigurable de modelos de retina (primer bloque de la figura 1). Los bloques funcionales que estamos incorporando en estos modelos tienen por objeto intentar hacer llegar a la matriz de electrodos (con un número bastante limitado de electrodos), cuanta más información útil sea posible.

Una retina biológica realiza básicamente, con el concurso de varias capas de neuronas, un realce de contrastes espacio-temporal, cuya salida es bastante independiente de las condiciones de iluminación [4],[5]. En algo más de 1 millón de fibras de nervio óptico es transmitida la información visual desde la retina hacia el siguiente centro nervioso del camino visual. Y este número reducido de fibras, mediante frecuencias de impulsos entre decenas y centenares de Hz, es capaz de transmitir en tiempo continuo al cerebro toda la información necesaria para que podamos reconstruir forma, color, y movimiento con gran precisión. Por tanto, creemos que es esencial realizar un pre-procesamiento bio-inspirado de la información visual que transmitimos al neuro-estimulador, intentando que la señal que finalmente llegue a cada microelectrodo: **(a)** porte la mayor cantidad posible de “información útil” para la percepción visual y **(b)** mantenga la máxima independencia posible de las condiciones de iluminación.

Dado el grado de paralelismo inherente al problema, y la necesidad de obtener una velocidad de procesamiento de secuencias de imágenes elevada, estamos utilizando soluciones basadas en cámaras digitales y FPGAs como medio de implementación y ensayo de modelos de retinas. Con los resultados obtenidos se podrá concretar las características finales y necesidades de procesamiento que deban incorporarse en soluciones ASIC más compactas.

II. RECURSOS UTILIZADOS

Handel-C — Se trata de un lenguaje de alto nivel orientado al hardware [3]. Está basado en ISO-C con algunas extensiones lógicas, como la de ejecución en paralelo e interfaz con otros módulos (incluso para simulación) especificados en lenguajes estándar de descripción de hardware. Gracias a la herramienta DK1 de Celoxica es posible sintetizar descripciones en Handel-C que empleen un estilo algorítmico, y más abstractas que las habituales en HDL para otros sintetizadores.

Para nuestros propósitos, como hemos comentado con anterioridad, es especialmente útil la facilidad que encontramos a la hora de trasladar algoritmos a este lenguaje.

Placa RC1000 — Adquirida de Celoxica, se trata de una tarjeta PCI diseñada para el desarrollo de aplicaciones en hardware reconfigurable. Viene provista de una FPGA de Xilinx y 4 módulos de RAM de 2MBytes cada uno, accesibles desde dicha FPGA o desde un programa *host* ejecutado en el PC. Por sus características, está especialmente indicada para tareas de co-procesamiento gráfico.

Software de comprobación — Para comprobar el correcto funcionamiento de cada uno de los filtros o módulos de procesamiento de imagen, hemos desarrollado varios programas *host* (en la terminología de Celoxica), bajo entorno Windows. El principal programa, que hemos llamado “*Tester-Live 1*”, captura la secuencia de imágenes que nos brinda una sencilla *webcam*, la envía a la placa RC1000 y va refrescando la pantalla del PC con el resultado obtenido (véase figura 2). El mismo filtro estará, por lo general, implementado en software, para comparar velocidades entre las versiones hardware y software. Es posible, por tanto, ver simultáneamente la captura original y el resultado del procesamiento. También se puede seleccionar el filtro en cuestión con el que se quiere configurar la FPGA.

Las mediciones de velocidad se han llevado a cabo por otro programa *host* (*Test-Speed*) diseñado para terminal, con objeto de minimizar al máximo la latencia provocada por el gestor de ventanas de Windows. Este programa es capaz de calcular con precisión de 1 microsegundo, el tiempo de procesamiento de la FPGA más las dos transferencias por DMA que se efectúan a y desde la placa RC1000.

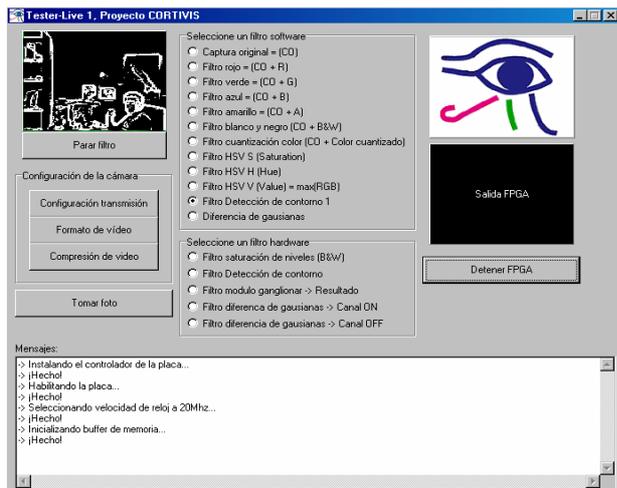


Fig. 2. Interfaz gráfica de la aplicación Tester Live-1.

Cámara logarítmica — La cantidad de información que el sistema sea capaz de extraer de las imágenes, y por tanto la capacidad para incluir más información en la codificación que se enviará a los electrodos, viene en gran parte determinada por la calidad del sensor.

Las cámaras CCD y CMOS convencionales, pierden información y saturan en escenas con un alto contraste luminoso. Para evitar este inconveniente en el punto de entrada al sistema, se está evaluando también el uso de una cámara CMOS en color, con característica de respuesta logarítmica ante los niveles de intensidad luminosa. El uso de este sensor nos permitiría adquirir imágenes en las que pueda haber un fuerte contraste luminoso, sin que las zonas claras se saturan o se pierdan las zonas oscuras. Los fotorreceptores de la retina humana se comportan de manera similar ante escenas de alto contraste luminoso. De esta forma, contaríamos con información más completa en origen.

El sensor CMOS logarítmico empleado permite adquirir imágenes de hasta 640x480 píxeles. Esta cámara es capaz de capturar 120 imágenes por segundo, frente a las 30 imágenes por segundo de una cámara CMOS estándar (ambas a 320x240). Su rango dinámico es de 120 dB. Otras características interesantes son el acceso aleatorio a píxeles individuales, como si se tratase de una RAM; la velocidad de lectura de un píxel, inferior a los 110 nanosegundos; y la capacidad para adquirir una subimagen (incrementando la tasa de transferencia hasta cerca de 4000 imágenes/s).

En la figura 3 se puede apreciar la diferencia entre una imagen adquirida con una cámara CMOS normal (a la izquierda) y la cámara logarítmica que estamos evaluando.



Fig. 3. Diferencia entre la captura de una escena de alto contraste luminoso, por un sensor CMOS normal y otro logarítmico.

módulo que realice la codificación final de las señales para los microelectrodos.

El diseño general se lleva a cabo a través de una serie de componentes, que modelan el comportamiento de los distintos tipos de células ganglionares de la retina. Cada uno de los componentes o “módulos ganglionares” implementa una combinación de filtros de tipo gaussiano, cuyas desviaciones típicas σ son programables.

Como se puede observar en la figura 4, el flujo de imágenes de la cámara (señal RGB), es primero procesado por el *host*. Este procesamiento se encarga de decodificar los tres *bytes* de cada color, de la terna RGB, prescindiendo de un cuarto *byte* enviado, que no nos interesa (usado por el controlador de la cámara). Los tres canales se reordenan convenientemente, y se construye un flujo de *bytes*, que mandamos de forma controlada a uno de los 4 módulos de SRAM, de que dispone la placa RC1000. Cada píxel que enviamos a la placa tiene por tanto 24 bits de profundidad de color (*True Color*). Cuando se ha terminado de transmitir un fotograma completamente, se comunica el evento a la placa, por medio de una línea de control. Es posible desde el *host* transmitir, como se indica en la figura, datos de configuración del filtro en cuestión, como máscaras y otros parámetros que manejará convenientemente el algoritmo implementado en la FPGA.

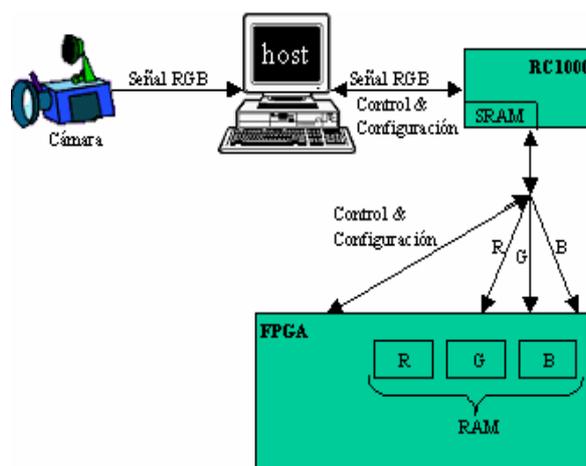


Fig. 4. Flujo de información en el sistema

III. IMPLEMENTACIÓN DEL MODELO DE RETINA

El modelo de retina en el que estamos investigando, se basa en el cálculo de una serie de filtros espaciales y temporales, cuyos resultados se integrarán dentro de un módulo “ponderador”. Las características de cada filtro deben ser parametrizables y bioinspiradas, así como el

Una vez que programa huésped comunica el fin de envío de un fotograma, la FPGA empieza a trabajar. Primero, se guardan en tres módulos de RAM interna de la FPGA (R, G y B en la figura 4), cada uno de canales de color. Para implementar esta memoria, hemos usado los bloques RAM de que dispone el Virtex-E 2000 (*BlockRAM*, según la denominación de Xilinx). Una vez que está lista toda esta información, se ejecuta en paralelo cada uno de los módulos ganglionares implementados en el prototipo. Al terminar el más lento de los módulos ganglionares, el resultado del proceso de cada módulo debe ser procesado por el módulo “ponderador”, que extrae la característica final a transmitir de cada fotograma. Según el tipo de módulo ganglionar que se esté implementando, y la salida que se precise, el resultado podrá ser almacenado en uno de los módulos de SRAM externa sin usar.

De esta manera, el esquema de trabajo para el diseño de la retina parte de una especificación de un módulo ganglionar, su descripción en Handel-C o VHDL, y su comprobación por separado. El trabajo aquí consistiría en fijar el mínimo número de unidades ganglionares que debemos implementar en la retina, para obtener información suficiente al objeto de generar los pulsos biocompatibles que exciten la corteza visual. Como es lógico, este esquema de funcionamiento depende fuertemente de la realimentación que nos aporte los experimentos biológicos previstos en el proyecto.

IV. RESULTADOS

Presentamos a continuación, las salidas que nos proporciona un módulo ganglionar concreto. Hemos escogido la implementación de la respuesta por oposición de color de una célula ganglionar del tipo ON-centro (rojo) OFF-periferia (verde). El esquema simplificado puede observarse en la figura 5. Los fotodetectores, sensibles a una longitud de onda baja (L) o media (M) se modelan mediante sendas vecindades de *píxeles*. Así, en la figura 5 la salida del sistema viene dada por la diferencia, ponderada por las distintas conexiones sinápticas, de las contribuciones de los *píxeles* del centro menos los de la periferia. En la FPGA, esto ha sido modelado usando filtros DOG (Diferencia de Gausianas) con máscaras de convolución definidas por el diseñador, en tiempo de implementación (mediante el uso de memorias ROM) o “en tiempo de ejecución” mediante el envío por parte del programa huésped, de las nuevas máscaras.

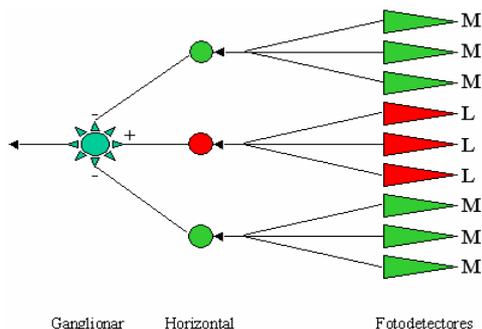


Fig. 5. Representación esquemática de un módulo ganglionar que implementa una respuesta de oposición de color rojo-verde, del tipo ON-centro OFF-periferia.

Como ejemplo, al aplicar las máscaras de la figura 6 sobre la imagen superior de la figura 7 (fotografía de un naranjo), se obtiene el resultado de la parte inferior.

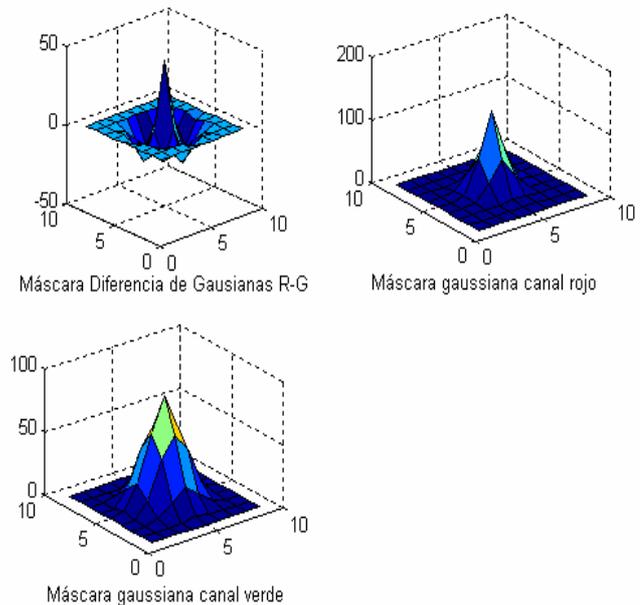


Fig. 6. Máscara utilizadas para el módulo ganglionar del ejemplo.

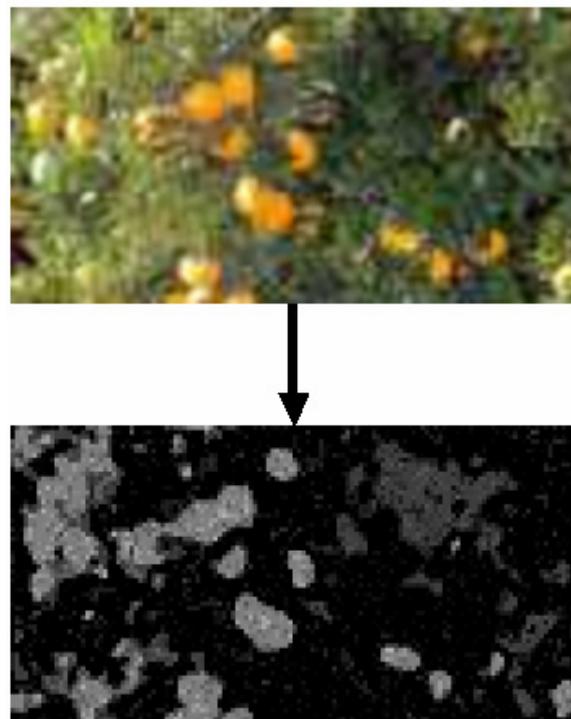


Fig. 7. Resultado obtenido por el módulo ganglionar rojo-verde descrito anteriormente en el texto.

Como puede observarse, se acentúan las zonas de la imagen con mayor contraste de rojo sobre verde, destacándose claramente la localización de las naranjas.

Este diseño, sintetizado con la herramienta *Xilinx Foundation 3.1i*, ha presentado el siguiente grado de ocupación de la Virtex-E 2000eBG560:

NIVEL DE OCUPACIÓN PRODUCIDO POR EL MÓDULO GANGLIONAR

Nº de Slices	599 de 19,200 = 3%
Nº de Slices Flip Flop	399 de 38,400 1%
Nº de LUTs	851 de 38,400 2%
Nº de IOBs	93 de 404 23%
Nº de Block RAMs	48 de 160 30%

El programa Test-Speed, comentado anteriormente, calcula aproximadamente 17'5 frames/segundo. Esta velocidad, aunque suficiente para nuestros propósitos podrá ser mejorada en el futuro mediante la conexión directa de la cámara logarítmica a la FPGA. Los posibles cuellos de botella son, por una parte, el inherente retardo de la operación de convolución y por otra, las transferencias de memoria entre los bloques blockRAM y la SRAM externa de la RC1000. Una transferencia por DMA de una secuencia entre *host* y RC1000, junto con un filtro sencillo (por ejemplo una umbralización), sin transferencias internas de memoria en la FPGA, es capaz de soportar más de 50 frames/segundo. La idea de usar los bloques internos blockRAM, está pensada para independizar la implementación del modelo de retina de la placa RC1000.

V. DESARROLLOS ALTERNATIVOS

A la par que se desarrolla el sistema descrito anteriormente, se está explorando el uso de *System Generator* como herramienta alternativa. Se trata de una "toolbox" desarrollada para *Simulink* por *Xilinx*. Dicha *toolbox* contiene una serie de elementos que son sintetizables automáticamente. Entre estos bloques se incluyen multiplicadores, registros, memorias, máquinas de estados, sumadores, etc.

Partiendo del entorno de *Simulink*, podemos incluir elementos de esta *toolbox* y combinarlos con otros bloques propios de *Simulink* [6]. De esta manera, es posible simular el sistema sin necesidad de sintetizarlo, obteniendo unos resultados que se corresponden con los del sistema final a nivel de bit (no así en cuanto a retardos).

Una vez que se comprueba que el sistema es "bit accurate", se procede a la generación automática de código VHDL. Obtenido este código, podremos simular su funcionamiento temporal, midiendo retardos (por ejemplo con *ModelSim*).

Por último, se procedería a la síntesis del sistema usando herramientas propias de *Xilinx*, como el *ISE Foundation*.

El sistema se desarrolla en dos fases. Durante la primera fase, en la que nos encontramos actualmente, se tratan imágenes individualmente. Durante la segunda etapa se tendrá en cuenta la componente temporal, procesando una secuencia.

Nos hemos basado en un sistema previo desarrollado en *Matlab* a partir de [7], en el que se modelan las tres capas de células fundamentales de la retina: fotorreceptores, células horizontales y células bipolares.

El desarrollo en paralelo de ambos sistemas nos permitirá comparar distintos parámetros de desarrollo de

estas dos herramientas de síntesis automática: tiempo de desarrollo, tiempo de síntesis, tamaño y calidad del código sintetizado, velocidad de procesamiento de imágenes, etc. El hardware de destino es el mismo tanto en el caso del modelo implementado en Handel-C como en el construido con *System Generator* (placa RC-1000 con la FPGA Virtex-E 2000).

VI. CONCLUSIONES

Se ha presentado el diseño e implementación de un sistema que modela el comportamiento de una retina humana, usando herramientas de síntesis automática de última generación. La alternativa de usar Handel-C como lenguaje principal para este cometido ha reducido en gran medida el tiempo necesario de desarrollo de los distintos módulos, a la vez que nos ha permitido depurar los mismos, de manera similar a como se realiza en cualquiera lenguaje de alto nivel moderno.

Los resultados obtenidos hasta el momento nos animan a continuar trabajando siguiendo la misma estrategia, si bien, estamos valorando otras posibles herramientas de síntesis automática, que puedan suponer un complemento en función de las necesidades de desarrollo.

En el futuro, esperamos obtener mejor rendimiento al minimizar los cuellos de botella de los prototipos actuales: implementando una captura directa de imagen desde la FPGA, manteniendo el paralelismo en los distintos filtros aplicados, e intentando incrementar el grado de paralelismo en la aplicación de cada uno de ellos.

AGRADECIMIENTOS

El trabajo aquí descrito se está desarrollando con financiación de la Comunidad Europea, con el proyecto CORTIVIS (QLK6-CT-2001-00279 ; <http://cortivis.umh.es>) dentro del programa europeo "Quality of Life and Management of Living Resources".

VII. REFERENCIAS

- [1] Normann RA, Maynard EM, Guillory KS, Warren DJ. *Cortical implants for the blind*. IEEE Spectrum 54-59, May (1996)
- [2] Maynard EM, Nordhausen CT, Normann RA. *The Utah intracortical electrode array: a recording structure for potential brain-computer interfaces*. Electroenceph. Clin. Neurophysiol. 102: 228-239 (1997).
- [3] Celoxica, <http://www.celoxica.com/>
- [4] Dowling J. E. "The retina. An approachable part of the brain", Harvard University Press, Cambridge 1987.
- [5] Spillmann L., Werner, J.S. "Visual Perception. The Neurophysiological Foundations", Academic Press. 1990
- [6] Xilinx. "Xilinx System Generator v2.1 for Simulink. User Guide. Xilinx Blockset Reference Guide".
- [7] Palomar Sáez, D. "Implementación VLSI mixta de una retina artificial adaptativa y salida codificada en frecuencia de impulsos." Memoria de Proyecto Fin de Carrera. Prof. Tutor. F. J. Pelayo. Dep. Electrónica y Tecnología de Computadores. Univ. de Granada, 1995.