

# Utilización de algoritmos morfológicos sobre FPGAs: un caso práctico

Ramón Sánchez Iborra, Álvaro Legaz Aparicio, Rafael Verdú Monedero, María-Dolores Cano

Departamento de Tecnologías de la Información y las Comunicaciones

Antiguo Cuartel de Antigones. Plaza del Hospital, Nº 1, 30202 Cartagena (Murcia)

Teléfono: 968338872

E-mail: ramon.sanchez@upct.es, alvar.legaz@gmail.com, {rafael.verdu, mdolores.cano}@upct.es

**Resumen.** Actualmente los algoritmos de visión artificial continúan estando en pleno auge. Métodos para el reconocimiento facial, conteo de productos, retoque de imagen, etc. son ampliamente utilizados tanto a nivel investigador como a nivel industrial. En aplicaciones reales, la base de este tipo de técnicas consiste en un conjunto de operaciones de filtrado básicas, muchas veces basadas en operaciones morfológicas. Este trabajo presenta el uso de estas operaciones de morfología matemática para el procesamiento de imagen. En concreto se emplea un método para la eliminación de ruido en imágenes binarias de huella digital por medio de aperturas y cierres. Su implementación se realiza en una FPGA con el programa Matlab y la herramienta Simulink con System Generator. De los resultados obtenidos, se observa la potencia de las operaciones morfológicas y cómo la plataforma empleada es una herramienta de diseño fácil e intuitiva que permite diseños HDL complejos en poco tiempo de forma eficiente.

## 1. Introducción

Los algoritmos de visión y procesamiento de imágenes han sido investigados y desarrollados desde hace décadas [1, 2]. Hoy en día, siguen en pleno auge debido a su aplicación en campos de actuación muy diversos. Por ejemplo, en entornos industriales se aplican para conteo y separación de piezas, reconocimiento de colores y texturas, etc.; en entornos agrícolas también son utilizados para controlar de forma autónoma las cosechas, obteniendo niveles óptimos en la calidad de los productos. Más recientemente, una conocida red social también ha hecho uso de este tipo de técnicas aplicado al reconocimiento facial en fotografías.

Para poder llevar a cabo este tipo de procesamiento una serie de transformaciones morfológicas deben ser empleadas. Las transformaciones morfológicas son herramientas matemáticas no lineales que permiten analizar la forma y estructura de los objetos que forman una imagen. Además, permiten modificar estas formas para separar elementos, obtener contornos primarios, descomponer los elementos más complejos en elementos más simples, eliminar ruido, etc. Este tipo de transformaciones pueden ser simples o, mediante el uso conjunto de éstas, formar operaciones complejas. Las operaciones básicas que se suelen utilizar en la amplia literatura relacionada son la erosión y la dilatación. Por su parte, algunas de las transformaciones complejas más empleadas son la apertura y el cierre. En siguientes secciones se realiza un exhaustivo análisis de cada una de las mencionadas técnicas.

Como se ha comentado anteriormente, el uso que se hace de las operaciones morfológicas es muy variado, como el Reconocimiento Óptico de Caracteres (Optical Character Recognition, O.C.R.), aplicado al reconocimiento de textos o matrículas [3], o el

reconocimiento de objetos por parte de robots [4]. En este trabajo se ha utilizado una serie de operaciones morfológicas con el objetivo de conseguir la completa eliminación de ruido en imágenes de huellas digitales. Más concretamente se hace uso de las operaciones de apertura y cierre obteniendo unos resultados altamente satisfactorios. Este algoritmo es diseñado en Matlab y es traducido a lenguaje de descripción hardware (Hardware Design Language, HDL) mediante la herramienta Simulink con System Generator para ser volcado a una FPGA; de esta forma, el procesamiento se realiza de forma hardware, aumentando el rendimiento del mismo. El resto del documento se estructura como sigue. En la Sección 2 se realiza un análisis de las transformaciones morfológicas empleadas. El entorno de simulación empleado es descrito en la Sección 3. La Sección 4 analiza de forma teórica el algoritmo empleado y su implementación sobre la FPGA. La Sección 5 muestra los resultados obtenidos. Finalmente, se presentan las conclusiones extraídas de este trabajo.

## 2. Transformaciones morfológicas

Como se ha mencionado anteriormente, las transformaciones morfológicas permiten modificar la estructura o forma de los objetos en una imagen, para simplificar ésta, preservando las formas principales de los objetos que la componen. Las transformaciones morfológicas básicas son la erosión y la dilatación. Por su parte, las transformaciones fruto de la utilización conjunta de las anteriores son, entre otras, la apertura y el cierre.

### 2.1 Erosión

La erosión es una técnica que permite la reducción de los contornos de los objetos que forman la imagen, permitiendo la separación de los mismos, o incluso su eliminación si su tamaño es pequeño. Para ello se

realiza un solapamiento secuencial de una máscara sobre todos los píxeles que forman la imagen. Cuando el píxel analizado no tenga un entorno exactamente igual que el elemento estructurante (máscara) empleado, dicho píxel se fija a 0. A la erosión de la imagen binaria  $A$  por el elemento estructurante  $B$  se le denota como  $A \ominus B$  y está definida por la expresión (1).

$$A \ominus B = \{x \in E^N \mid x = x + b \in A\} \forall b \in B \quad (1)$$

## 2.2 Dilatación

Es la operación complementaria a la anterior (pero no la inversa, ya que si realiza una erosión y, posteriormente, una dilatación, no se obtiene la imagen original). La dilatación produce un engrosamiento de los bordes de los objetos que forman la imagen, permitiendo un aumento en el contorno de los mismos para unir líneas discontinuas entre estos, fruto de, por ejemplo, algún filtrado. Para llevar a cabo esta transformación, se realiza un solapamiento de una máscara sobre todos los píxeles de la imagen original; si algunos de los elementos del entorno del píxel analizado son iguales que los de la máscara, dicho píxel se fija a 1. A la dilatación de la imagen binaria  $A$  por el elemento estructurante  $B$  se le denota como  $A \oplus B$  y está definida por la expresión (2).

$$A \oplus B = \{c \in E^N \mid c = a + b\} \forall a \in A, b \in B \quad (2)$$

## 2.3. Apertura

La apertura consiste en realizar sobre una imagen, primero una erosión y, posteriormente, una dilatación con el mismo elemento estructurante. Con ello se consigue un efecto de alisado de contornos, descomposición de objetos en elementos más simples y extracción de formas en entornos con ruido. A nivel matemático, la apertura de  $A$  por un elemento estructurante  $B$  se define tal y como se muestra en la expresión (3).

$$A \circ B = (A \ominus B) \oplus B \quad (3)$$

## 2.4. Cierre

Esta es la operación resultante de aplicar sobre una imagen, primero una dilatación y, posteriormente, una erosión con el mismo elemento estructurante. Los efectos que se consiguen sobre la imagen original son, entre otros, alisar porciones del contorno, fusionar estrechas grietas o rellenar pequeños agujeros. El cierre de la imagen  $A$ , mediante una máscara  $B$ , se define como se muestra en la expresión (4).

$$A \cdot B = (A \oplus B) \ominus B \quad (4)$$

## 3. Entorno de diseño

El entorno de desarrollo empleado para la realización de este trabajo ha sido *Matlab*, con la herramienta *System Generator* para procesamiento digital de señales (Digital Signal Processing, DSP) que permite, desde *Simulink*, modelar algoritmos DSP, generar código de

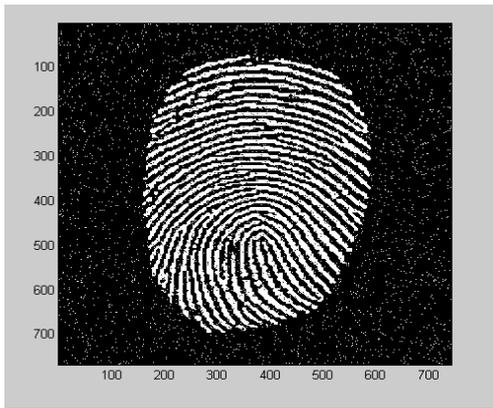
descripción hardware (HDL) con una librería de bloques optimizada para dispositivos DSP y verificar los diseños FPGA para dispositivos de *Xilinx*. La herramienta *System Generator* ofrece una gran cantidad de componentes básicos predefinidos. Además dispone de la posibilidad de implementar bloques propios desde una base de bloques más simples. Otra característica interesante que presenta *System Generator* es la posibilidad de generar código VHDL que indique el uso de los recursos de la FPGA empleados en la sintetización del algoritmo diseñado.

## 4. Algoritmo e implementación

Como ya se explicó en la introducción, el objetivo de este trabajo es realizar un procesamiento sobre la imagen binaria (blanco, negro) de una huella dactilar. Esta imagen proviene de un sistema de adquisición de huellas dactilares que pretende un posterior almacenamiento en una base de datos. El problema que presentan este tipo de imágenes es la aparición de cierto ruido en torno a la huella dactilar tomada (ver Fig. 1). El objetivo del algoritmo que se propone es la eliminación de este ruido en la señal binaria. Para esto se realizará un procesamiento de imágenes que constará de la concatenación de dos operaciones morfológicas; en primer lugar se hará uso de una operación de apertura con el fin de suavizar la imagen y eliminar elementos menores que el elemento estructurante. Finalmente, mediante una operación de cierre se alisa la imagen resultante de la apertura y obtiene una huella dactilar libre de ruido.

El algoritmo implementado está formado por cuatro bloques. Cada uno de ellos implementa una operación morfológica básica (Erosión + Dilatación = Apertura) + (Dilatación + Erosión = Cierre). Para la implementación del algoritmo se usará una FPGA modelo Spartan3E-XC3S100E-TQ144-4. El diseño circuital implementa cada uno de los cuatro bloques que modelan las operaciones morfológicas. Estos bloques presentan una estructura interna característica, consistente en procesar un píxel determinado como combinación del píxel y de sus vecinos multiplicados por un elemento estructurante, en este caso concreto, un cuadrado de  $3 \times 3$  y, dependiendo del tipo de operación morfológica que implemente, quedarnos con el mínimo o el máximo del conjunto de píxeles bajo estudio. La ventaja que aporta este diseño es que se trabaja sobre una imagen binaria, por tanto la operación de máximo o mínimo se pueden implementar como las operaciones lógicas OR y AND.

El tiempo de procesamiento es muy importante en este diseño ya que no dispondremos de una imagen filtrada hasta que no transcurra el tiempo suficiente para que se procese toda la imagen. El tiempo de procesamiento de este algoritmo es la suma del tiempo de latencia del sistema más el tiempo que se tarda en procesar el array de píxeles que forma la imagen. Es importante señalar que este tiempo de procesamiento estará expresado en ciclos de reloj. Por tanto, el tiempo total de procesamiento dependerá de la frecuencia

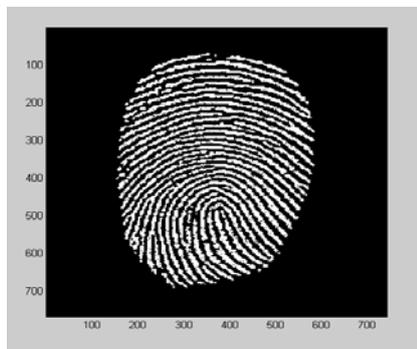


**Figura 1.** Imagen de la huella dactilar ruidosa. Imagen 750x750 píxeles.

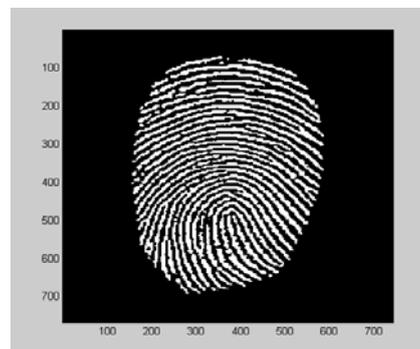
de la oscilación del reloj de la FPGA. Para el diseño global habrá que tener en cuenta el tiempo de procesado de cada uno de los bloques, que implementan cada operación morfológica. Afortunadamente los elementos estructurantes empleados en cada uno de los bloques son del mismo tamaño por lo que la latencia de cada bloque será idéntica al resto de bloques. Es decir, la latencia consistirá en cada una de las latencias de cada uno de los sub-bloques, es decir cuatro veces la latencia de un sub-bloque.

## 5. Resultados

El resultado final de aplicar el algoritmo que aquí se presenta sobre una imagen de huella digital es presentado en la Fig. 2-b. Se observa cómo se consigue eliminar las “impurezas” producto del ruido y, así, obtener una imagen lista para la extracción de características y su posterior almacenado. Los resultados parciales muestran el efecto de las operaciones internas de apertura y cierre (ver Fig. 2-a). La imagen procesada con la apertura ha eliminado las componentes de la imagen menores al elemento estructurante (ruido) mientras ha conservado las huellas dactilares pero con cierta distorsión en la huella. Para corregir este efecto, se aplica la operación de cierre que elimina la distorsión y nos proporciona la huella filtrada y lista para la siguiente etapa del diseño.



(a)



(b)

**Figura 2.** Resultado parcial. (Operación apertura sobre imagen ruidosa) (a). Resultado del procesamiento de la imagen con el algoritmo de eliminación de ruido completo (Apertura + Cierre) (b). Imágenes 750x750 píxeles

## 6. Conclusiones

El uso de FPGAs se encuentra cada vez más extendido, esto se debe principalmente tanto a las formidables capacidades que presentan como dispositivos reconfigurables, como al nivel de integración o la mejora de las frecuencias de trabajo. Si a esto añadimos el uso de *Simulink* con *System Generator*, disponemos de una herramienta visual muy intuitiva que permite el diseño rápido de sistemas de procesado de imagen, con la capacidad de generar código HDL. En este trabajo se han utilizado las capacidades descritas con anterioridad y la librería de DSP de *System Generator* para implementar elementos más complejos que han sido los encargados de realizar las operaciones morfológicas necesarias para el procesado de imagen. Concretamente, se ha llevado a cabo un filtrado de ruido sobre imágenes de huella dactilar adquiridas de forma binaria. En los resultados se muestra la efectividad del algoritmo diseñado, al quedar la imagen procesada completamente libre de ruido y sin sufrir distorsión adicional. Finalmente, es importante destacar la simplicidad y rapidez de diseño frente a lo que habría sido la implementación de un diseño directamente en VHDL.

## Referencias

- [1] R. A. Kirsch, L. Cahn, C. Ray, and G. H. Urban, “Experiments in processing pictorial information with a digital computer,” in *Papers and discussions presented at the December 9-13, 1957, eastern joint computer conference: Computers with deadlines to meet on XX - IRE-ACM-AIEE’57 (Eastern)*, 1958, pp. 221–229.
- [2] S. Peleg, J. Naor, R. Hartley, and D. Avnir, “Multiple Resolution Texture Analysis and Classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 4, pp. 518–523, Jul. 1984.
- [3] T. Sirithinaphong and K. Chamnongthai, “The recognition of car license plate for automatic parking system,” in *ISSPA’99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications*, vol. 1, 1999, pp. 455–457.
- [4] A. Gopalakrishnan, S. Greene, and A. Sekmen, “Vision-based mobile robot learning and navigation,” *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005*, pp. 48–53, 2005.