

Sistemas de Tempo Real: FPGAs

2013/14

Práctica 7: Segmentación de Imágenes

Objetivo

Repasar los conceptos vistos en las prácticas anteriores e introducir la lectura y escritura de ficheros.

Introducción

Una de las aplicaciones típicas de las FPGAs es el tratamiento de imágenes a altas velocidades. La FPGA suele obtener datos a través de los conectores de I/O a los que se acopla un dispositivo que permita realizar la captura de las imágenes (una cámara o directamente un sensor), las cuales son suministradas a intervalos regulares para su procesamiento. Este procesamiento puede requerir de múltiples operaciones como conversiones de formato, segmentación, detección de bordes...



Ejemplo: Identificador de matrículas.

Enunciado

En esta práctica vamos a implementar en VHDL un sencillo segmentador de imágenes que nos permita identificar determinados patrones en base a valores RGB. Para ello, supongamos que nuestro objetivo es realizar un sistema de detección de incendios forestales basado en FPGAs que procesan la información procedente de cámaras que suministran imágenes periódicamente. Estos dispositivos de monitorización de incendios estarán ubicados en dos tipos de localizaciones: en postes/árboles diseminados por el bosque y en UAVs (*Unmanned Aerial Vehicles*) que realizan recorridos sobre las zonas de monte más inaccesibles.

Los dispositivos situados en los postes/árboles deben detectar la presencia de fuego a distancias relativamente cercanas, mientras que los UAVs han de ser capaces de localizar humaredas mientras planean sobre las superficies arboladas.

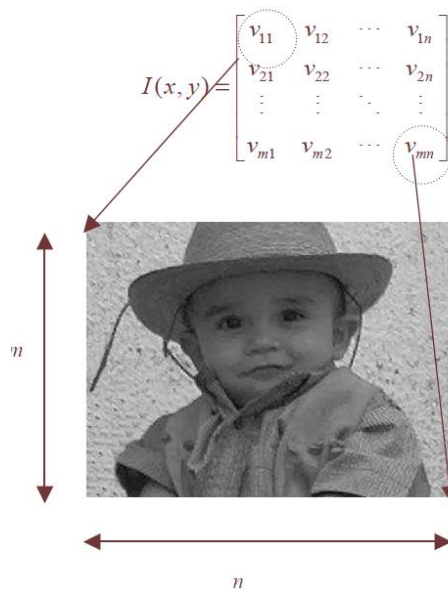


Cámaras en UAV y poste para monitorización de incendios en montes.

Desarrollo

Para simular el suministro de imágenes por parte de una cámara, se va a realizar la lectura de un fichero que en cada línea posee un entero entre 0 y 255. Este fichero se generará mediante la función de Matlab *generaEntradaFPGA*, utilizando alguna de las imágenes suministradas. Debe tenerse en cuenta (de cara a optimizar el código VHDL) que las imágenes que obtiene la cámara van a tener un tamaño fijo, que en este caso se ha establecido en 480x360 píxeles.

También es necesario saber que cada imagen leída en Matlab se compone de al menos una matriz en la que cada valor se corresponde con un pixel de la imagen.



En esta práctica, las imágenes a tratar serán en color por lo que habrá tres matrices distintas, una por cada componente RGB.



$$I_R(m, n, 1) = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix} \quad I_G(m, n, 2) = \begin{bmatrix} g_{11} & g_{12} & \dots & g_{1n} \\ g_{21} & g_{22} & \dots & g_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \dots & g_{mn} \end{bmatrix} \quad I_B(m, n, 3) = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix}$$

El fichero generado por el código en Matlab posee una única “columna” de valores que se crea al reordenar cada componente cogiendo los valores de columna en columna, y colocando primero la componente R, luego la G y luego la B.

Este fichero de entrada será leído y procesado en el Xilinx ISE donde se llevará a cabo la segmentación. Este proceso, aunque en determinados casos puede ser muy complejo, se va a reducir a tratar cada uno de los valores de las componentes para detectar los patrones de fuego y humo.

En el caso del fuego, es fácil detectarlo por poseer una componente R alta, mientras que los valores de G y B son relativamente bajos. El humo puede ser también localizado fácilmente debido al contraste del blanco con el verde del bosque.

Una vez realizado el procesado de cada valor, se escribirá en un fichero en el que habrá un único valor por línea (es decir, poseerá el mismo formato que el fichero de entrada), pudiendo ser leído en Matlab con la función *leeSalidaFPGA* con el fin de observar la imagen resultante.

Se pide:

- 1) Diseñar e implementar en VHDL el segmentador de imágenes haciendo uso de un único proceso. Este proceso será el encargado de tratar los valores del fichero de entrada e irá escribiendo el resultado en un fichero de salida con el mismo formato que el de entrada. Verificar que el resultado es el deseado visualizando la imagen resultante en Matlab.
- 2) Proponer, justificar, diseñar e implementar estrategias que permitan incrementar el nivel de paralelismo obtenido en el apartado (1). Por ejemplo, se podría mejorar el rendimiento tratando las componentes R, G y B en procesos paralelos que mantengan la comunicación mediante señales.
- 3) Escoger entre una de las siguientes opciones:
 - a. Proponer y utilizar de manera justificada alguna de las IP Core básicas disponibles.

- b. Aplicar el mecanismo de procesamiento de imágenes desarrollado en (1) y (2) en otra situación realista mostrando ejemplos de su uso. Justificar la aplicación de una FPGA en dicho problema.
- 4) Escribir una memoria justificativa con la descripción de los diseños realizados. Enviar este documento junto con una copia comprimida del directorio del proyecto a tmfernandez@udc.es.