



SIE 2019. AUTOMATIZACIÓN, ROBÓTICA Y SISTEMAS COMPUTACIONALES

Detección de matrículas mediante el algoritmo Viola-Jones en un sistema Zynq.

License plate detection using the Viola-Jones algorithm in a Zynq system.

Ing. Eduardo Javier Vázquez Pérez

Centro de Informática Industrial. Universidad de las Ciencias Informáticas, Cuba.
evazquez@uci.cu

Resumen:

En el presente trabajo se expone el desarrollo de una aplicación software para la detección de matrículas cubanas empleando el algoritmo de Viola-Jones utilizando el lenguaje de programación C. El algoritmo de Viola Jones ha sido implementado mediante una cascada de 13 etapas con un total de 99 rasgos, obtenidos previamente mediante una fase de entrenamiento. Estos pueden ser modificados, logrando que el sistema propuesto sea adaptable a diferentes tamaños de imágenes. La aplicación es implementada sobre una placa de desarrollo Zedboard que integra el dispositivo Zynq-7020.

Se expone el soporte hardware utilizado sobre el cual se ejecuta la aplicación, así como el entorno de desarrollo empleado para la programación del dispositivo. Se realiza además una comparación de efectividad y tiempo de la aplicación con el implementado software en un FPGA-Spartan6. El tiempo de ejecución es 3,4 veces menor que el obtenido anteriormente, mientras que la efectividad de la detección se mantiene en un 96%, debido a que el objetivo es mejorar la respuesta temporal y no la efectividad del algoritmo Viola Jones implementado previamente.

Palabras Clave: Sistemas Embebidos; Procesado de Imágenes; FPGA; Zynq



Abstract:

In the present work the development of a software application for the detection of Cuban license plates is exposed using the algorithm of Viola-Jones by the programming language C. The algorithm of Viola Jones has been implemented through a cascade of 13 stages with a total of 99 features, previously obtained through a training phase. These can be modified, making the proposed system adaptable to different image sizes. The application is implemented on a Zedboard development board that integrates the Zynq-7020.

It exposes the hardware support used on which the application is executed, as well as the development environment used for the programming of this device. A comparison of effectiveness and time of the application is also made with the software implemented in a FPGA-Spartan6. The execution time is 3,4 times less than that obtained previously, although the effectiveness of the detection is not affected, it is maintained in 96%, due to that the objective of the modifications are the improving of the temporal response and not the effectiveness of the Viola Jones algorithm implemented previously.

Keywords: *Embedded Systems; Processing of Images; FPGA; Zynq*

1. Introducción

En el mundo actual el procesamiento digital de imágenes se lleva a cabo en diversas esferas. Desde la tecnología detrás de una cámara digital o de un teléfono celular hasta aplicaciones industriales. Dentro de este campo la identificación de imágenes consiste en encontrar e identificar objetos dentro de una imagen o secuencia de video. Esta brinda múltiples servicios, entre los que se encuentra el control de calidad gracias a la detección de desperfectos en ciertos materiales y el reconocimiento de rostros o matrículas de vehículos que contribuye a la seguridad y al control.

Los algoritmos de procesamiento digital de imágenes (DIP, por sus siglas en inglés) presentan, en general, una gran carga computacional e imponen serias restricciones temporales para asegurar la operación en tiempo real. Por este motivo la inclusión de algoritmos de procesamiento de visión en sistemas empujados con recursos de cálculo limitados, requiere de la aplicación de métodos que permitan acelerar su ejecución (Marek, 2008) (Benkrid, 2002).



1.1 Problemática:

La identificación de matrículas de vehículos en la actualidad exige que una imagen completa sea transmitida hacia el centro de control del Sistema Inteligente de Transporte (SIT), para que en el mismo se realicen los procesos de detección de la matrícula, segmentación e identificación de los caracteres. Esto implica la necesidad de utilizar un elevado ancho de banda en la transmisión de una imagen completa. Este ancho de banda podría ser reducido si algunos de estos procesos se realizan mediante un sistema empotrado desde el mismo sitio donde se capta la imagen.

1.2 Objetivo: Desarrollar una aplicación software para la detección de matrículas cubanas utilizando el algoritmo de Viola-Jones implementado en una placa de desarrollo Zedboard basada en dispositivo Zynq.

1.3 Proceso de detección de matrículas.

El estudio sobre algoritmos de reconocimiento de matrículas ha atraído la atención de forma considerable en los últimos años debido a la amplia gama de aplicaciones que incluye, como por ejemplo: la recolección automática de peajes, el control del tráfico y la supervisión, así como el control y el acceso a estacionamientos (Kolour, 2011). El sistema de reconocimiento automático de matrículas tiene que ser apto para trabajar en un ambiente variable ya que puede existir en determinado momento problemas de iluminación o mal estado del clima.

Un sistema de reconocimiento de matrículas como se muestra en la Figura 1 consta de cuatro etapas: adquisición de la imagen, detección de la matrícula (LPD, por sus siglas en inglés), segmentación o extracción de caracteres y reconocimiento de caracteres.



Figura 1. Pasos para el reconocimiento de matrículas.



El primer paso tiene lugar al recibir una señal de un sensor de movimiento indicando que está pasando un vehículo y se captura la imagen. En el segundo, un sistema de procesado analiza la imagen capturada para determinar la localización de la matrícula. En el tercer paso se segmenta la zona de la imagen previamente localizada para que en un cuarto y último paso se identifiquen los caracteres de la matrícula utilizando un módulo de reconocimiento de caracteres (OCR, por sus siglas inglés). En estas aplicaciones, las imágenes imperfectas dificultan el proceso de reconocimiento de caracteres (Wang SZ, 2003). Este proceso se puede ver afectado ya sea por la suciedad presente en la placa o la presencia de algún material obstructivo por lo que el sistema puede no detectar los caracteres (Jing Y, 2017).

1.4 Algoritmos de detección de matrículas.

En la literatura consultada existen numerosos trabajos dedicados a la revisión bibliográfica del tema de detección y reconocimiento de matrículas. Sin embargo este trabajo se enfoca en la segunda etapa del sistema de reconocimiento por lo que a continuación se referencian trabajos relacionados a esta etapa.

El trabajo desarrollado por C. N. E. Anagnostopoulos et al. (Anagnostopoulos CNE, 2006) se destaca por crear una nueva técnica de segmentación denominada Desplazamiento de Ventana Concéntrica, la cual funciona a partir de las características estadísticas de una imagen binarizada, cuenta con una efectividad de detección del 86%.

Los métodos basados en el procesamiento de imágenes a color para la localización de matrículas son también propuestos en la literatura (Comelli P, 1995) (S, 1997) (Wang T, 2004) (Shi X, 2005) (Yan D, 2001). La etapa de segmentación de colores en los métodos basados en colores de una imagen es crucial para obtener resultados acertados. En el caso de (Del Toro Hernández E, 2012) se empleó un ordenador Intel Pentium III con un procesador de 1 GHz para detectar matrículas en imágenes de 1024x768 píxeles obteniendo un tiempo de detección de aproximadamente 300 milisegundos y una efectividad del 90%.

K. Thooyamani et al. (Thooyamani K, 2013) desarrollaron un método de escaneo el cual cuenta los bordes en las ventanas consideradas como posibles matrículas que se analizan de la imagen y, si el número de bordes en esta ventana es mayor que un valor umbral definido, esto indica la presencia de una matrícula. Los resultados para una implementación sobre un ordenador Core 2 Duo con un procesador a 2.4 GHz con imágenes de 640x480 píxeles tuvieron una efectividad del 95% pero tardó 2 segundos en realizar la detección.



Los filtros de Gabor han sido una de las mejores herramientas para el análisis de texturas en el campo de la detección de objetos. Esta técnica tiene la ventaja de analizar la textura en un ilimitado número de direcciones y escalas. Un método basado en la Transformada de Gabor es presentado et al. (Kahraman F, 2003) .Los resultados fueron muy alentadores (98% para la detección de matrículas) cuando se aplican en imágenes digitales tomadas en ángulos muy específicos para una implementación en un ordenador Intel Pentium IV con un procesador a 1.4 GHz. Pero el método es computacionalmente costoso y lento cuando las imágenes implican mucho análisis pues esta misma implementación para imágenes que variaron de 512×384 a 768×576 píxeles, tomó 3.12 segundos.

1.5 Soluciones sobre plataformas hardware de algoritmos de detección de matrículas.

S. Rizvi et. al (Rizvi S, 2017) presentan un sistema integrado de detección y reconocimiento de matrículas italianas basado en plataformas que utilizan clasificadores neuronales profundos en plataformas móviles con unidades de procesamiento gráfico (GPU, por sus siglas en inglés) .Además, presentan un tiempo de detección de 300 ms que se consideran demasiado elevados para este tipo de aplicación.

B. Rouhani et al. (Rouhani B, 2017)desarrollan una aplicación (framework) automatizada para la video vigilancia inteligente en tiempo real sobre un FPGA para el reconocimiento de matrículas, al que llamaron RISE. En este trabajo proponen una aplicación informática genérica para la substracción de fondo adaptable en entornos no controlados. Los autores introducen un enfoque adaptativo de aprendizaje de reglas con un consumo de memoria bajo, que se adapta a la naturaleza dinámica de la tarea de vigilancia. Según los autores, el enfoque propuesto es menos costoso desde el punto de vista computacional debido a su poco consumo de memoria y es más adecuado para aceleradores de hardware como los FPGAs. Aunque la solución cumple con los requisitos de tiempo real al criterio de los autores, no es viable para los FPGA de gama media por el alto de uso de bloques de 18 kB de memoria, 190 en total.

2. Metodología.

En este trabajó se implementó el algoritmo de detección de matrículas, en una placa de desarrollo Zedboard que cuenta con un dispositivo Zynq Programmable SoC.-7000 All.



2.1 Implementación del algoritmo utilizando una placa Zedboard como soporte hardware.

Un System-on-Chip (SoC) es un único circuito integrado de silicio que implementa la funcionalidad de un sistema completo, en lugar de necesitarse varios chips físicos diferentes interconectados entre sí. Hablando de sistemas digitales, se puede combinar procesado, lógica de propósito específico, interfaces y memorias, entre otros.

Los dispositivos Zynq-7000 All Programmable Soc de Xilinx comprenden dos partes principales integradas en un único chip: un Sistema de Procesado (PS, por sus siglas en inglés) basado en un microprocesador Dual-Core ARM Cortex A9 y Lógica Programable (PL, por sus siglas en inglés) basada en la arquitectura de las FPGA 7-series de Xilinx. Con esta arquitectura se combina la capacidad de programación de software de un procesador embebido con la flexibilidad del hardware de un FPGA. Además, permite a los desarrolladores de software embebido, abstraerse de la carga que representa para estos el desarrollo de hardware, dándoles un nivel sin precedentes de control en el proceso de desarrollo (Crockett LH, 2014).

La comunicación entre el PS y la PL se realiza mediante Advanced Extensible Interface (AXI, por sus siglas en inglés), estándar, que propician conexiones de alta velocidad de transmisión y baja latencia (Crockett LH, 2014). La parte de PL es ideal para implementar lógica de alta velocidad, acelerando tareas con una alta carga computacional, como sistemas aritméticos, procesado de imagen y video en tiempo real, entre otros. El PS soporta sistemas operativos completos y rutinas de software, de manera que la funcionalidad total de cualquier sistema puede ser adecuadamente particionada entre hardware y software.

Entre las potenciales aplicaciones de los Zynq se incluyen las comunicaciones cableadas y Wireless, automoción, procesado de imagen y video, computación de altas prestaciones, medicina, control industrial y muchas más. La arquitectura puede ser empleada para satisfacer las demandas de aplicaciones con requerimientos tanto de computación paralela de alta velocidad como de funcionalidad secuencial con uso intensivo del procesador.

2.2 Análisis del algoritmo de la aplicación para la detección de matrículas.

La aplicación desarrollada parte de considerar que existe una imagen en escala de grises almacenada en la memoria del sistema de procesamiento, la cual contiene un automóvil, pues se supone que el sensor de movimiento le indique a la cámara el momento adecuado para captar la imagen. La resolución de las imágenes de entrada utilizadas es de píxeles y,



partiendo de las características de las matrículas cubanas de automóviles, se utiliza una ventana de escaneo de tamaño fijo de píxeles. Antes de pasar a la descripción del algoritmo es conveniente realizar un grupo de consideraciones previas que ayudarán a la comprensión del mismo:

- Para ejecutar la detección de matrículas se cuenta con una imagen o un conjunto de las mismas con tamaño conocido.
- Dado este tamaño de imágenes y según la aplicación a que esté destinada la realización del algoritmo se supondrá un intervalo de tamaños de las matrículas que pudieran aparecer en las imágenes. Este intervalo de tamaños de matrículas a detectar servirá para definir el factor de escalado de las ventanas que se generen a partir de la imagen original (candidatos). Estas ventanas generadas son las que se pasan a las etapas de detección, puesto que son estas las que pudieran contener una matrícula.

La estrategia seguida para la generación de candidatos es una decisión del diseñador que, por tanto, puede variar. Las variantes que se pueden ejecutar son:

- Comenzar desde los más pequeños e ir incrementando el tamaño progresivamente, según el factor de escalado, hasta los más grandes.
 - Partir desde los más grandes hasta los más pequeños.
 - Tener un conjunto de tamaños de candidatos predeterminados de antemano.
 - Utilizar una ventana de escaneo fija. Esta variante fue la utilizada en la aplicación pues los resultados temporales se veían muy afectados cuando se trabajaba con más de una ventana y la eficiencia en cuanto a la detección no mejoraba en gran medida.
- Se decide la forma en que la ventana de detección recorrerá la imagen para generar los candidatos. Aunque existen diversas formas de realizar este recorrido, una opción básica sería recorrer la primera columna de la imagen desde arriba hacia abajo y luego desplazarse hacia la columna de la derecha y repetir el proceso hasta llegar a recorrer toda la imagen. Es posible definir también que el factor de escaneo sea distinto para cada tamaño de ventana, aunque para esta investigación el factor de escaneo fue fijo.

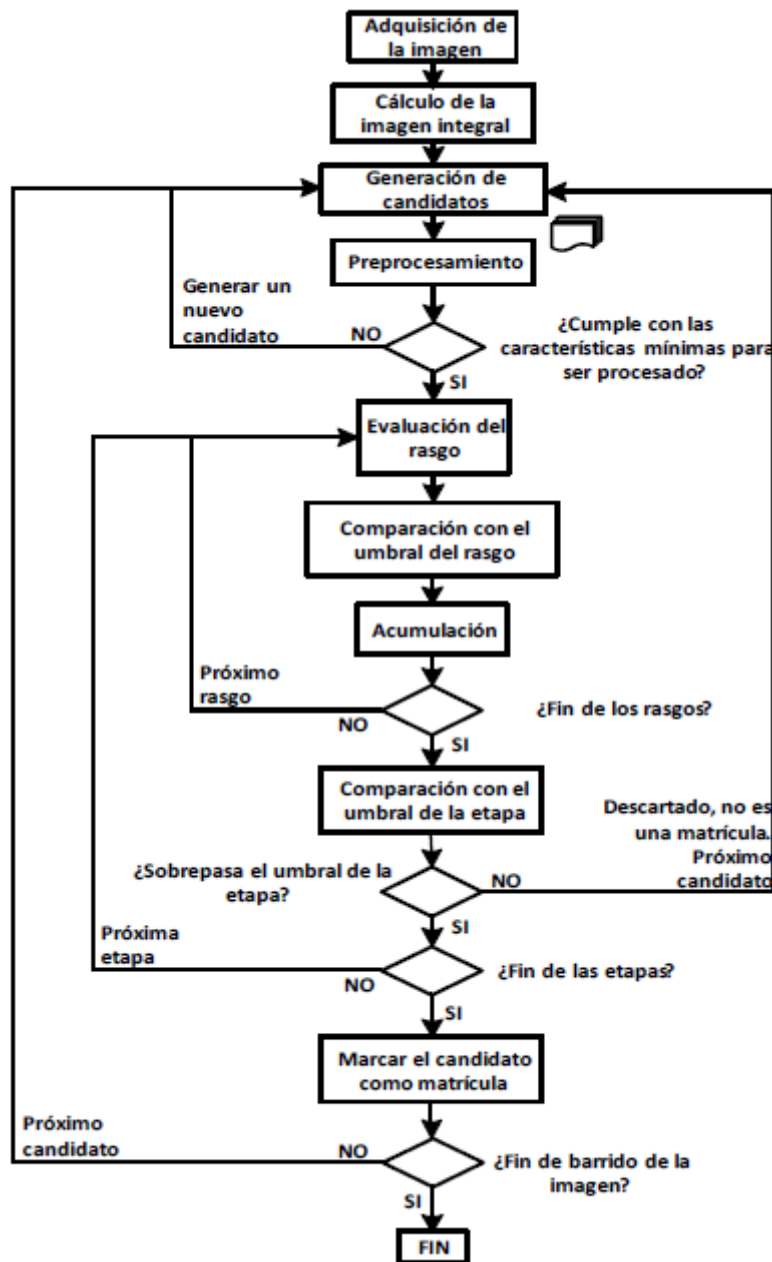


Figura 2. Diagrama de flujo del algoritmo de Viola-Jones para la detección de matrículas.

3. Resultado de la ejecución del algoritmo

Para comprobar el resultado de la ejecución del algoritmo de forma visual se utiliza un programa elaborado en MATLAB a partir de los resultados del proceso de detección. En la Figura 3 se observa que este dibuja un rectángulo azul por cada candidato que ha sido marcado como matrícula junto en la imagen correspondiente. La ejecución del algoritmo para esta imagen brinda tres candidatos como matrículas (dibujados como rectángulos azules). Como se puede apreciar, los tres candidatos coinciden correctamente con lo que sería una matrícula en la imagen y con una generación nula de falsos positivos.



Figura 3. Resultado de la ejecución del algoritmo.

La obtención de un grupo de candidatos positivos para una misma matrícula es característico cuando las matrículas están muy bien definidas en la imagen. Un sistema de post-procesamiento eliminaría este cúmulo de ventanas al agruparlas en una sola si comparten una gran porción de área común. El post-procesamiento permite eliminar también falsos positivos que, por lo general, no suelen aparecer en una misma área y agrupar las detecciones positivas en un solo candidato. En este trabajo no se realiza ninguna implementación de post-procesamiento. Esto se debe a que el algoritmo que se aplique para el post-procesamiento no depende de los resultados de la detección de matrículas, sino de la aplicación en donde se ejecute el algoritmo.

3.1 Efectividad de la implementación del algoritmo

Previo al desarrollo de este documento se realizaron una serie de experimentos para probar la efectividad de la aplicación en función de la detección de matrículas. De la base de datos brindada por CENATAV se tomaron de forma aleatoria un total de 100 imágenes las cuales fueron analizadas por el algoritmo. Los resultados de ello fueron muy alentadores, pues del total de imágenes procesadas, solo 4 de ellas dieron resultados desacertados, dando lugar a una efectividad de un 96 %. Las imágenes en las que no se detectó ninguna matrícula fue debido a la pobre calidad. Finalmente respecto a los falsos positivos, el sistema tuvo una eficiencia del 100%, pues en ninguna de las 100 imágenes se detectó una matrícula que no existiera.

3.2 Comparación de la implementación software del algoritmo de Viola-Jones en la FPGA Spartan-6 vs Zynq-7000.

Para realizar un análisis del tiempo de ejecución y de la efectividad del algoritmo de identificación, se realizaron mediciones empleando el mismo conjunto de imágenes que se utilizaron en las mediciones realizadas para la implementación en la FPGA Spartan-6. Los



resultados en la identificación de las matrículas en cuanto a efectividad son iguales, variando solamente el tiempo de ejecución del algoritmo. El resultado que se obtuvo fue de un tiempo de ejecución de 82,2 ms para la identificación de una matrícula en una imagen. Lo cual representa una reducción de 3,3 veces el tiempo de ejecución de la implementación software de tiempo de ejecución 279.2 ms realizada por nuestro grupo de investigación en una placa Atlys Spartan-6.

4. Conclusiones

Se obtuvo y validó una propuesta de implementación software de un algoritmo para la identificación de matrículas cubanas en lenguaje C sobre la placa de desarrollo Zedboard.

Las propuestas de solución consultadas que realizan la implementación de algoritmos de detección de matrículas se caracterizan en su mayoría por su complejidad computacional, a diferencia de la obtenida. El algoritmo mostró una efectividad de 96% y una demora en el procesado de una imagen de 82,2ms, siendo 3,4 veces más rápido que el resultado alcanzado en una FPGA Spartan-6 de Xilinx.

Para futuros trabajos se propone implementar vía hardware las partes más críticas del algoritmo explotando las capacidades de la placa Zedboard. Generar y adaptar un nuevo entrenamiento con menos etapas (entre 4 y 7) con el fin de acelerar el proceso de detección. Además se puede modificar el algoritmo para trabajar con más de una ventana de escaneo, siempre velando afectar lo menos posible la velocidad de ejecución del mismo.

5. Referencias bibliográficas

1. Anagnostopoulos CNE, A. I. (2006). A license plate-recognition algorithm for intelligent transportation system application. *IEEE Trans Intell Transp Syst*, 377–392.
2. Benkrid, B. (2002). Design and implementation of a 2D convolution core for video applications on FPGAs. *Third Int Workshop Digit Comput Video*, (págs. 85-92).
3. Comelli P, F. P. (1995). Optical recognition of motor vehicle license plates. *Trans Veh Technol*, 790-799.
4. Crockett LH, E. R. (2014). *The Zynq Book: Embedded Processing with the Arm Cortex-A9 on the Xilinx Zynq-7000 All Programmable Soc.* . UK: Strathclyde Academic Media.



5. Del Toro Hernández E, A. J.-S. (2012). Aceleración del algoritmo de detección de rostros de Viola-Jones mediante su implementación híbrida flexible sobre hardware reconfigurable.
6. Jing Y, Y. B. (2017). An efficient FPGA implementation of Optical Character Recognition for License Plate Recognition. 1-4.
7. Kahraman F, K. B. (2003). Computer and Information Sciences-ISCIS. 381–388.
8. Kolour, S. (2011). An Evaluation of License Plate Recognition Algorithms. *Int J Digit Inf Wirel Commun IJDIWC.* , 247-253.
9. Marek. (2008). Remarks on Hardware Implementation of Image Processing Algorithms. *Math Comput Science*, 105-110.
10. Rizvi S, P. D. (2017). Deep Classifiers-Based License Plate Detection. *Recognition on GPU-Powered Mobile Platform*, 66-94.
11. Rouhani B, M. A. (2017). An Automated Framework for Real-Time Intelligent Video Surveillance on FPGA. *ACM Transactions on Embedded Computing Systems*, 1-18.
12. S, D. (1997). A neural network based artificial vision system for licence plate recognition. *Int J Neural Syst*, 113–126.
13. Shi X, Z. W. (2005). Automatic license plate recognition system based on color image processing. 1159–1168.
14. Thooyamani K, K. V. (2013). Application of Pattern Recognition for Farsi License Plate Recognition. Vol. 19. Middle-East Journal of Scientific Research.
15. Wang SZ, L. H. (2003). A Cascade Framework for a Real-Time Statistical Plate Recognition System. *IEEE Trans Inf Forensics Secur.* , 267–282.
16. Wang T, N. F. (2004). Robust license plate recognition based on dynamic projection warping. 784–788.



II Convención Científica Internacional 2019
Universidad Central "Marta Abreu" de Las Villas
CIENCIA, TECNOLOGÍA Y SOCIEDAD. PERSPECTIVAS Y RETOS

17. Yan D, H. M. (2001). A high performance license plate recognition system based on the web techniqu. 325–329.