

Desarrollo de un algoritmo para el reconocimiento de objetos basado en FPGA aplicado a un Robot SCARA

Carlos Pillajo

Facultad de Ingeniería Electrónica
Universidad Politécnica Salesiana (UPS), Quito – Ecuador
cpillajo@ups.edu.ec

Resumen – El presente trabajo tiene por objetivo el desarrollo de un algoritmo para el reconocimiento de objetos basados en FPGA, algoritmo que será implementado en un robot tipo SCARA, el cual se encargara de seleccionar ó clasificar objetos, este sistema tiene una aplicación muy útil en muchas situaciones del mundo real como la detección de calidad del los productos, el seguimiento de objetos, la percepción 3D, Este proyecto se centra en la forma de detección para observar la imagen del objeto real, este trabajo comienza describiendo los conceptos básicos del procesamiento digital de imágenes, luego describe el método de conversión de imagen RGB a una señal binaria mediante el uso de MATLAB. Entonces la señal binaria se utiliza como una señal de entrada para la parte de detección. Algoritmo de detección utilizado en el diseño es un método fijo

Índices – Algoritmo FPGA's

I. INTRODUCCIÓN

Durante las últimas décadas, el interés en el uso de robots en la educación ha aumentado significativamente debido al potencial de afectar la naturaleza de la ingeniería y la ciencia la educación en todos los niveles [1], Aunque la apariencia de los robots industriales no ha cambiado significativamente y muchos modelos actuales tienen una estructura y funcionamiento similares entre sí, diferenciándose únicamente en su escala, lo cierto es que tanto en los aspectos mecánicos, como en el control y la programación, se ha visto el verdadero potencial de estos manipuladores, pudiendo ser susceptibles a cualquier modificación de tarea, sin alterar su estructura mecánica. Estos sistemas requieren de interfaces que depende de precisión, en tiempo real el reconocimiento de los objetos reales y la información digital relativa requieren de algoritmos de visión en tiempo real para el reconocimiento de objetos en 3D. [2]. Con el avance de la tecnología las FPGA's se han convertido en una herramienta indispensable en el diseño rápido y eficiente de proyectos que requieren procesamiento digital y la lógica programable [3]. La necesidad de obtener mayor rapidez y precisión en el desempeño del manipulador robótico, incluso bajo condiciones de restricción en el área de trabajo, lleva a

realizar un control basado en algoritmos del procesamiento digital de imágenes

Para ello, el presente trabajo se basará en técnicas de segmentación y operadores para la detección de bordes, los cuales se analizará para encontrar los que mejor se adapten al requerimiento planteado

II. DESARROLLO DEL TRABAJO

A. *Conceptos Generales del procesamiento digital de imágenes.*

El procesamiento digital de señales nos permite extraer información considerada relevante, o modificarla para darle un uso apropiado, empleando para ello herramientas computacionales. Se puede emplear el procesamiento digital de señales en una gran variedad de aplicaciones tales como: transmisión de información a través de canales de comunicación, restauración y mejoramiento de imágenes, reconocimiento de objetos en imágenes, medición de parámetros tales como la velocidad en objetos en desplazamiento capturados en video, etc.

Este proyecto empleará la detección de objetos basada a su vez en el proceso conocido como Segmentación de Imágenes.

Definición de Imagen Digital

Una imagen digital, puede ser definida como una función en dos dimensiones, $f(x,y)$, donde: x e y son coordenadas espaciales, la amplitud de f en el punto (x,y) es conocida como la intensidad de la imagen en ese punto. Tanto, los valores de las coordenadas x e y , como de la amplitud de la función f son valores finitos.

El término nivel de gris, se lo emplea para referirse a la intensidad en imágenes monocromáticas. Las imágenes a color, están formadas por una combinación de imágenes individuales en tres dimensiones.

Una imagen está compuesta por un número finito de elementos (puntos) con una ubicación y valor de intensidad particular, donde cada elemento es conocido como pixel.

Transformación de Intensidad [4]

Una transformación espacial opera directamente sobre los pixeles de una imagen. La transformación de intensidad en el

dominio espacial, es denotada por la expresión:

$$g(x,y) = T [f(x,y)] \quad (1)$$

Donde $f(x,y)$ es la imagen sobre la cual se aplicará la transformación, $g(x,y)$ es la imagen procesada y T es un operador sobre f , definido sobre un vecindario especificado cerca al punto (x,y) . T puede operar sobre un conjunto de imágenes, por ejemplo sumando K imágenes para provocar una reducción de ruido.

Filtrado espacial

El procedimiento que se lleva a cabo al realizar el filtrado espacial es el que sigue:

- Se escoge un punto central, (x,y) .
- Se desarrolla una operación que involucre solo los pixeles que rodean al punto central.
- El resultado de esa operación será la "respuesta" del proceso (filtrado) en ese punto.
- El proceso es repetido para cada punto en la imagen.

A medida que el punto central es movido, se crean nuevos vecindarios, (siendo los vecindarios, el conjunto de pixeles que rodean al punto central incluido dicho punto), uno para cada pixel en la imagen f . Si los cálculos realizados sobre los pixeles de un vecindario son lineales, la operación es llamada filtrado espacial lineal, de otra manera es llamada filtrado espacial no lineal. El filtrado espacial también es conocido como procesamiento de vecindario. Figura 1.

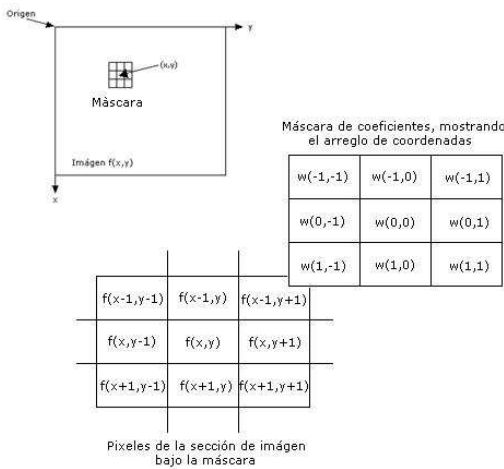


Fig. 1. Filtrado Espacial.

Existen dos conceptos muy importantes, relacionados estrechamente e involucrados en el filtrado espacial lineal. Uno es la correlación; el otro es la convolución. La Correlación es el proceso de pasar la máscara w por el arreglo de imagen f . La convolución es el mismo proceso, excepto que w es rotado 180° antes de ser pasada sobre f .

Teorema de Convolución y Correlación [5]

La convolución discreta de dos funciones $f(x,y)$ y $g(x,y)$ de tamaño $M \times N$ es denotada por:

$$f(x,y) * h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n)h(x-m,y-n).$$

Se conoce que el teorema de convolución consiste en la siguiente relación entre dos funciones y sus transformadas de Fourier

$$f(x,y) * h(x,y) \Leftrightarrow F(u,v)H(u,v) \quad ,y$$

$$f(x,y)h(x,y) \Leftrightarrow F(u,v) * H(u,v)$$

La correlación de dos funciones $f(x,y)$ y $h(x,y)$ es definida como:

$$f(x,y)^\circ h(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m,n)h(x+m,y+n)$$

Donde f^* denota el complejo conjugado de f . Normalmente se trata con funciones reales (imágenes), en cuyo caso $f^*=f$. La función de correlación tiene exactamente la misma forma que la función de la convolución con la excepción de que el complejo conjugado y el hecho de que el segundo término de la suma tiene positivos en lugar de signos negativos. Esto significa que h no se refleja sobre el origen. Todo lo demás en la aplicación de la correlación es idéntica a la convolución, incluyendo la necesidad de relleno.

Dada la similitud de la convolución y la correlación no es de sorprenderse que el teorema de correlación sea análogo al teorema de convolución, por lo tanto:

$$f(x,y)^\circ h(x,y) \Leftrightarrow F^*(u,v)H(u,v) \quad ,y$$

$$f^*(x,y)h(x,y) \Leftrightarrow F(u,v)^\circ H(u,v)$$

Estos dos resultados comprenden el teorema correlación. Se supone que todas las Funciones han sido debidamente extendidas. Como se sabe, la convolución es el vínculo entre el filtrado en los dominios espaciales en frecuencia. El uso principal de la correlación es para la coincidencia. En la coincidencia, $f(x,y)$ es una imagen que contiene objetos o regiones. Si queremos determinar si f contiene un objeto en particular o regiones (que normalmente llamamos esta imagen de una plantilla). Entonces, si existe una coincidencia, la correlación de las dos Funciones será máxima en el lugar donde h encuentra una correspondencia en f , el preprocesando así como el escalado y la alineación son necesario en la mayoría de las aplicaciones prácticas, pero la mayor parte del proceso lo realiza la correlación.

Segmentación de Imágenes

La segmentación, es un proceso en el cual una imagen es sub-dividida en las regiones u objetos que la componen. En este caso, la segmentación debería de concluir cuando los objetos de las imágenes, hayan sido aisladas.

Los algoritmos de segmentación para imágenes

monocromáticas generalmente están basados ya sea en la discontinuidad o en la similitud de los valores de intensidad en las imágenes. En la primera categoría, el enfoque es para particionar una imagen basado en cambios abruptos de intensidad, tal como ocurre en los bordes de una imagen. Los algoritmos basados en la detección de bordes, son los más básicos que existen en el tratamiento del tema de segmentación. Los principales algoritmos en la segunda categoría se basan en particionar la imagen en regiones que son similares de acuerdo a un conjunto de características predefinidas.

Detección de Discontinuidades basada en valores de Intensidad

La forma básica para encontrar discontinuidades, es pasar una máscara a través de la imagen, en la forma descrita en la sección anterior de filtrado, para una máscara de 3x3 este procedimiento involucra computar la suma de productos de los coeficientes con los niveles de intensidad contenidos en la región abarcada por la máscara. Esto es, la respuesta R de la máscara en cualquier punto (x, y) de la imagen está dado

$$R = \sum_{k=1}^9 W_k Z_k$$

$$R = W_1 Z_1 + W_2 Z_2 + \dots + W_9 Z_9$$

Donde z_k es la intensidad del pixel asociado con el coeficiente w_k de la máscara.

Detección de Bordes

Un borde, dentro de una imagen, es considerado un tipo de discontinuidad, por lo que este puede ser detectado usando derivadas de primer y segundo orden.

La derivada de primer orden para una imagen es el gradiente, el cual, para una función en dos dimensiones, $f(x,y)$ viene dado por:

$$\nabla f = [G_x \ G_y] = \left[\frac{df}{dx} \ \frac{df}{dy} \right]$$

La magnitud de este vector es:

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{\frac{1}{2}}$$

Esta cantidad se puede simplificar en algunos casos, usando el valor absoluto de los términos:

$$\nabla f = |G_x| + |G_y|$$

Esta aproximación, se comporta también como una derivada, esto quiere decir, que se hace cero en áreas de intensidad constante y sus valores son proporcionales al grado de cambio de intensidad, en áreas donde los valores de los pixeles son variables, Es común referirse a la magnitud del gradiente o su aproximación simplemente como "el gradiente"

III. TRABAJO A REALIZARSE

En este trabajo los algoritmos a implementarse se basan en la utilización de ventanas de píxeles dentro de la imagen, para realiza el procesamientos de los mismos y obtener la salida correspondiente. Los algoritmos basados en operadores de ventana se pueden dividir en 3 etapas principales. Una etapa de generación de ventana, un contador de filas y columnas y la etapa final específica del algoritmo a implementar. El objetivo del Generador de Ventanas es almacenar en un buffer la cantidad de píxeles necesarios para formar una ventana de $N \times N$, (en donde N define el tamaño de la ventana). Luego de esto, el generador de ventana deberá ir realizando un desplazamiento de la ventana a lo largo de toda la imagen hasta lograr el procesamiento de todos los píxeles. Luego, el contador de filas y columnas se encargará de determinar cuáles son los píxeles de los bordes de la imagen, esto se hace porque en los bordes de la imagen no se puede aplicar el procesamiento de ventana por falta de información.

Dentro de los algoritmos que utilizan estos operadores se desarrollarán los filtros de mediana, (muy útiles para eliminar el ruido sal y pimienta), los basados en operadores morfológicos y la convolución espacial.

Operadores Morfológicos

El término de Operadores Morfológicos en procesamiento de imágenes se refiere a la clase de algoritmos que están interesados en la estructura geométrica de una imagen. La morfología puede ser usada sobre imágenes binarias o en escala de grises, y usada en muchas áreas de procesamiento de imágenes, tales como esqueletización, detección de bordes, restauración y análisis de textura.

Hay 2 operaciones fundamentales en morfología: erosión y dilatación. Es común pensar erosión como la contracción de un objeto en una imagen. Dilatación es lo opuesto, agranda el objeto en la imagen. Ambos conceptos dependen del elemento estructurado y de cómo se ajusta en el objeto. La salida de una operación de dilatación es un píxel en primer plano para todos los puntos en el elemento estructurado a tal punto que el píxel original se ajusta en un objeto imagen.

La erosión y dilatación pueden ser representadas matemáticamente mediante las siguientes relaciones:

$$\text{Erosión} : A \ominus B = \{ x : (B)_x \subseteq A \}$$

$$\text{Dilatación} : A \oplus B = \cup \{ A + b : b \in B \}$$

Donde A es la imagen de entrada y B es el elemento estructurado

La erosión y dilatación en escala de grises puede ser obtenida también mediante un filtro de ordenamiento por rango. La erosión corresponde a un filtro de ordenamiento por rango de orden mínimo, y la dilatación corresponde a un filtro de ordenamiento por rango de orden máximo. La razón de esto

es que el resultado de un filtro de ordenamiento por rango de orden mínimo es el valor mínimo en la zona del píxel, que es exactamente lo que una operación de erosión hace. Esto también sigue siendo válido para un filtro de ordenamiento por rango de orden máximo y la operación de dilatación. De cualquier manera, el filtro de ordenamiento por rango solo trabaja como una operación morfológica con un elemento estructurado plano. Esto es así porque el filtro de ordenamiento por rango trabaja como una clase de elemento estructurado formada por todos unos. Aun así, esta es una característica poderosa, ya que la morfología en escala de grises usando elementos estructurados planos describe los usos más comunes de la morfología.

Filtros de Ordenamiento por Rango

Dentro de los filtros de Ordenamiento por rango se encuentra el filtro de mediana que se implementa mediante una ventana de píxeles que se va desplazando a los largo de la imagen. En cada desplazamiento se realiza el ordenamiento (ranking) de los píxeles y se reemplaza el valor del píxel central llamado origen por el valor medio determinado por el resultado del ordenamiento. En la figura 2 se muestra una ventana de nueve píxeles, en donde luego del ordenamiento de los mismos, se toma como salida el píxel que se encuentra en la posición 5, ya que este es un filtro de ordenamiento por ranking de orden 5 y de esta manera se implementa el filtro de mediana

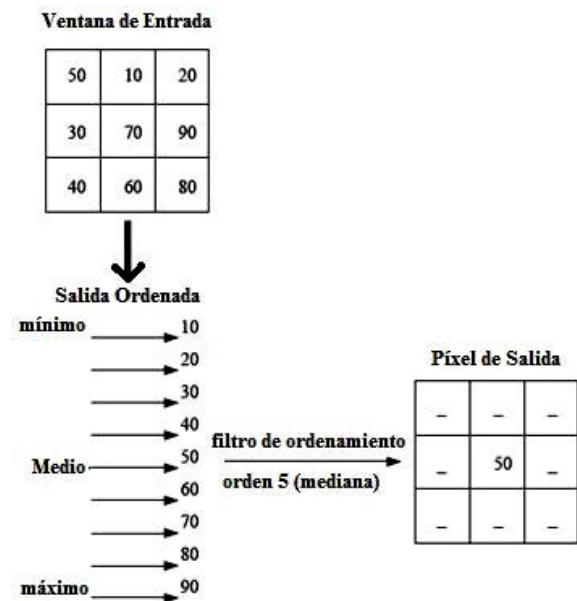


Fig. 2. Filtro de Mediana.

De manera similar pero tomando el píxel de mayor intensidad, se obtiene el filtro de máxima el cual permite realizar la dilatación que describen los operadores morfológicos. Tomando el píxel de menor intensidad se

obtendrá el filtro de mínima que permite implementar la erosión que también forma parte de dichos operadores.

Convolución Espacial

Los filtros espaciales usan una amplia variedad de máscaras, también conocidas como kernels, para calcular diferentes resultados, dependiendo de la función deseada. Por ejemplo, ciertas mascarar implementan el alisado, mientras otras realizan el filtrado pasa bajos o detección de bordes. La convolución se puede calcular multiplicando cada píxel de una ventana por la mascara (kernel), para luego realizar la suma de cada producto y por último dividir por el número de píxeles de la ventana. Este valor es el píxel de salida en la ubicación del píxel original en la imagen de salida. Matemáticamente la convolución se representa mediante la siguiente ecuación:

$$y(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} A(k_1, k_2)k(n_1 - k_1, n_2 - k_2)$$

Donde A es la imagen de entrada y K es la máscara de convolución.

La ventana del píxel de entrada es siempre del mismo tamaño que la máscara de convolución. El píxel de salida es redondeado al entero más cercano. La figura siguiente muestra la ventana de píxeles de entrada, la máscara de convolución, y el resultado de la salida. La máscara de convolución que se muestra en la Figura 3 es frecuentemente usada como filtro para eliminar ruido.

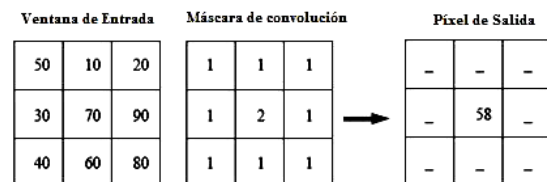


Fig. 3. Filtro de convolución

Descripción del Sistema

El sistema en la FPGA estará formado por 3 módulos principales; el Generador de Ventanas, el Contador de Filas y Columnas y el Módulo que implementa el algoritmo deseado. Por otro lado, el sistema consta de un módulo de comunicación serial para la adquisición y el envío de datos procesados .

El filtro de mediana, de Máxima y de Mínima se implementará en VHDL mediante un módulo generador de ventanas, el cual genera ventanas de 3x3 píxeles, un módulo de ordenamiento de píxeles que ordena los píxeles según su intensidad y un módulo contador de filas y columnas. Por otro lado, El algoritmo de la Convolución Espacial contiene el módulo de generación de ventanas, el contador de filas y columnas y un bloque específico que realiza la convolución entre cada una de las ventanas y la máscara a aplicar.

El la Figura 4 se muestra el diagrama en bloques del

sistema completo, en donde la imagen se convierte del formato Jpg a binario mediante el programa `jpg_a_bin.m`, luego se enviará a la FPGA mediante un software de comunicación serial, en donde se procesará y se envía dicha imagen nuevamente a la PC vía RS-232. Por último, mediante el programa `bin_a_jpg.m` se transforma la imagen procesada al formato original.

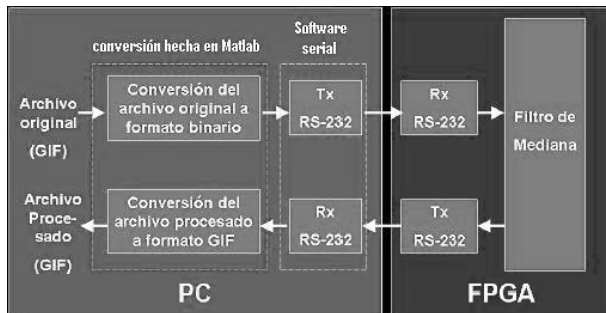


Fig. 4. Diagrama de Bloques del sistema a implementarse

A. Resultados.

Como resultado de esta primera parte ha sido el análisis de los conceptos básicos del procesamiento digital de imágenes que se necesitarán para implementar el sistema el cual servirá más adelante realizar el reconocimiento y la selección de objetos que serán manipulados por medio del brazo robótico tipo SCARA (Selective Compliant Articulated Robot Arm.).

B. Conclusiones.

Existe una amplia base teórica acerca del procesamiento digital de imágenes, para lo cual se necesita conocer el fundamento de los algoritmos que podrían implementarse.

El algoritmo a implementarse depende mucho de precisión que se desee en la aplicación específica.

El usos de los FPGA's permitirán realizar el procesamiento digital de imágenes casi en tiempo real, para lo cual también se necesita conocer el funcionamiento de la tarjeta FPGA a utilizarse.

Actualmente existen herramientas computacionales que permiten el procesamiento digital de señales con FPGAs, de una forma más amigable, tal como el LabView, siempre y cuando el Labview tenga la librería de manejo del FPGA utilizado.

IV. BIBLIOGRAFÍA

[1] J. Macedo, K. Colvin y D. Waldorf, «Machine vision course for manufacturing engineering undergraduate students,» *Journal of Manufacturing Systems*, vol. 24, nº 3, pp. 256-265, 2005.

[2] J. Zhang, F. Zhang y M. Ito, «Image processing based remote control with robot arm simulator,» 2009.

[3] R. J. A., «Desarrollo de Algoritmos de Procesamiento de Imágenes Basados en "Operadores de Ventana" sobre una FPGA,» 2010.

[4] C. B. Patricia., «Implementación de un algoritmo para la detección y conteo de células en imágenes microscópicas,» 2009.

[5] R. W. Rafael, *Digital Image Processing*, P. Hall, Ed., Tom Robinson, 2002.

[6] F. Faux y F. Luthon, «Theory of evidence for face detection and tracking,» *International Journal of Approximate Reasoning*, vol. 53, nº 5, pp. 728-746, 2012.

[7] M. Husin, F. Osman, M. Sabri, W. Abidin, A. Othman y A. Marzuki, «Development of shape pattern recognition for FPGA-based object tracking system,» 2010.

[8] V. J. Traver y A. Bernardino, «A review of log-polar imaging for visual perception in robotics,» *Robotics and Autonomous Systems*, vol. 58, nº 4, pp. 378-398, 2010.

[9] J. Zhang, F. Zhang y M. Ito, «Image processing based remote control with robot arm simulator,» 2009.

[10] J. DÑaz, E. Ros, R. AgÑas y J. L. Bernier, «Superpipelined high-performance optical-flow computation architecture,» *Computer Vision and Image Understanding*, vol. 112, nº 3, pp. 262-273, 2008.

[11] Y. Hirano, K. Kitahama y S. Yoshizawa, «Image-based object recognition and dexterous hand/arm motion planning using RRTs for grasping in cluttered scene,» 2005.

[12] E. N. Malamas, E. G. Petrakis, M. Zervakis, L. Petit y J.-D. Legat, «A survey on industrial vision systems, applications and tools,» *Image and Vision Computing*, vol. 21, nº 2, pp. 171-188, 2003.

[13] K. Kawamura, K. Hasegawa, Y. Someya, Y. Sato y K. Ikeuchi, «Robust localization for 3D object recognition using local EGI and 3D template matching with M-estimators,» 2000.