

Módulo de visión por computadora en tiempo real para el control de un sistema plano-esfera

Pérez-Olvera César A., Moreno-Armendáriz Marco A., Rubio-Espino Elsa, Zaragoza-Díaz de León J. O.

Centro de Investigación en Computación-IPN
Laboratorio de Reconocimiento de Patrones
Av. Juan de Dios Bátiz s/n. U.P. "Adolfo López Mateos",
col. Nva. Industrial Vallejo, c.p. 07738, México, D.F.
Contacto: cperezob06@ipn.mx
Paper received on 04/08/08, accepted on 08/09/08

Resumen: En este artículo, se muestra el funcionamiento de un módulo de visión por computadora que se utilizó como lazo de retroalimentación para el control de un sistema Plano-Esfera (PE). Este módulo de visión tiene como entrada una imagen de 10bits de resolución por píxel y como salida, las coordenadas del centroide de la esfera, basado en la imagen. A la imagen de entrada se le aplicó la función de umbral para separar el objeto de interés de la imagen de fondo. Se utilizó la transformación morfológica de *erosión*, para eliminar el ruido que resulta de dicha función de umbral. Una vez eliminado el ruido, se calcularon las coordenadas del centroide de la esfera. El comportamiento del módulo de visión fue descrito mediante un lenguaje de descripción de hardware y embebido en un circuito integrado; logrando así reducir el tiempo de procesamiento de la tarea de visión.

1 Introducción

La visión es uno de los cinco sentidos, con los que el cerebro humano puede formar una representación del mundo que lo rodea. Esta representación puede ser considerada como una imagen, en donde la luz se refleja en los objetos de su alrededor. A partir de esta imagen, podemos extraer información relevante para comprender e interactuar con el entorno. En el ser humano, esto es un proceso natural, sin embargo, aun no se ha podido comprender del todo como se realiza.

Con el surgimiento de nuevos paradigmas en las ciencias de la computación, como la inteligencia artificial, y con los avances tecnológicos tanto en software como en hardware, se ha tratado de desarrollar la capacidad de visión en una computadora de manera artificial, con el fin de realizar tareas de inspección, en donde la información que pueden brindar otro tipo de herramientas resulta insuficiente. Por ejemplo, si se pretende que una computadora realice la tarea de visión como la realiza el ser humano, ésta deberá estar provista de un algoritmo que incluya entre otras cosas, una aproximación al proceso de percepción y cognición visual, el modelo matemático del entorno, sensores conectados a ella, etc.

Debido a lo complejo de los procesos de visión por computadora y a sus aplicaciones en tiempo real, resulta difícil la ejecución de éstos en equipos que cuentan con procesadores convencionales [1]. Una opción para reducir el tiempo de ejecución de los procesos de visión por computadora, es formular un procesamiento concurrente y secuencial utilizando herramientas de diseño de hardware. Además de que se cuenta con la posibilidad de probar físicamente el funcionamiento mediante su equivalente circuito lógico embebido, en un circuito integrado que resulta de la tecnología FPGA (Field Programmable Gate Array).

El FPGA fue inventado en el año 1984 por Ross Freeman co-fundador de Xilinx Corporation, como resultado de dos tecnologías diferentes, los dispositivos lógicos programables (PLS - Programmable Logic Devices) y los circuitos integrados de aplicación específica (ASIC- Application-Specific Integrated Circuit). El FPGA, contiene bloques lógicos que son programables, por lo que puede reproducir desde operaciones lógicas básicas, hasta el conjunto de ellas para producir sistemas complejos como un procesador.

2 Sistema Plano-Esfera.

El sistema PE se muestra en la Fig. 1. Está compuesto por dos ejes perpendiculares entre si y se suponen desacoplados. Dos servomotores conectados a cada eje, darán movimiento al plano, en el cual se posicionará y estabilizará la esfera en la coordenada deseada.

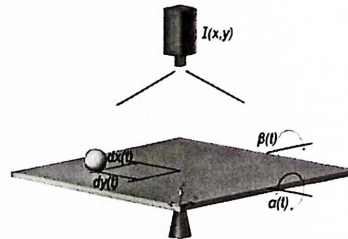


Fig. 1. Diagrama del sistema plano-esfera en la dirección x .

$dx(t)$	Posición de la esfera en la dirección x .
$dy(t)$	Posición de la esfera en la dirección y .
$\alpha(t)$	Posición angular de la mesa en la dirección x .
$\beta(t)$	Posición angular de la mesa en la dirección y .
$I(x,y)$	Imagen digital de la esfera sobre el plano.

Como se muestra en la Fig. 2, el sistema utiliza una cámara digital como sensor de retroalimentación para conocer la posición de la esfera sobre el plano. Esta cámara entrega una imagen digital $I(x,y)$, que un procesador embebido en un FPGA toma

como entrada, la imagen pasa por dos etapas: el pre-procesamiento, que consta de la aplicación de una función de umbral a la imagen, con el fin de separar el objeto de interés de la imagen de fondo, posteriormente se aplica una transformación morfológica de erosión, como operación no-lineal, logrando eliminar el ruido producido en la imagen por la función de umbral; la segunda etapa consta del cálculo de la posición de la esfera en el plano. Con el dato de la posición, se puede utilizar el modelo geométrico de la cámara digital para conocer la posición real de la esfera sobre el plano.

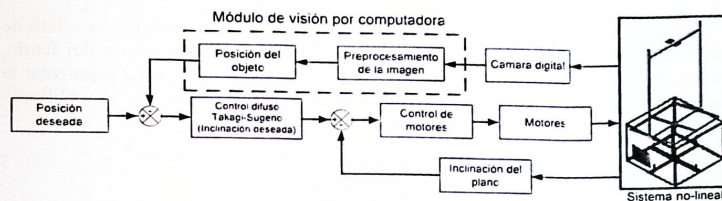


Fig. 2. Diagrama a bloques del sistema plano-esfera.

Como se observa en la Fig. 2, una vez que se obtiene la posición actual de la esfera y la deseada por el usuario, se retroalimenta a un al bloque de control. Se utiliza un control difuso de tipo Takagi-Sugeno [2], para calcular la posición angular deseada del plano y que la esfera se mueva en la dirección apropiada. Con la posición angular deseada y la actual proveniente del sensor de inclinación, se generan las señales de control para los servomotores que moverán el plano. Finalmente, la esfera rodará bajo la fuerza de gravedad hasta el punto de consigna.

Una vez descrito de manera general el funcionamiento del sistema PE, en este documento nos enfocaremos a describir el módulo de visión que se encuentra embebido en un circuito integrado.

3 Módulo de visión por computadora

El módulo de visión por computadora está conformado por tres componentes:

1. Función de umbral.
2. Transformación morfológica de erosión, y
3. Cálculo del centroide.

3.1 Función de umbral.

Es una de las transformaciones más sencillas que se puede aplicar a una imagen en niveles de grises $I(x,y)$ para obtener una imagen $U(x,y)$ en dos tonos (blanco y negro). También es una técnica de segmentación, ya que divide la imagen en subre-

giones. El valor de umbral B es el que decide cuales puntos de la imagen pasan al negro o al blanco. La elección del valor de umbral B depende de la complejidad de la imagen, ya que puede haber uno o varios puntos de umbral dependiendo de la segmentación que se quiera conseguir.

La expresión matemática de esta transformación es la siguiente:

$$U(x,y) = f(I(x,y)) = \begin{cases} 0, & 0 \leq I(x,y) \leq B \\ 1, & B \leq I(x,y) \leq 1023 \end{cases} \quad (1)$$

Con el fin de facilitar el procesamiento de la imagen, se consideró la esfera de color blanco y el plano de color negro para resaltar el objeto de interés del fondo, ver Fig. 3, por lo que sólo se necesita un valor de umbral B para poder segmentar la imagen y representar los componentes conexos de la esfera con $u(x,y)=1$ y el fondo con $u(x,y)=0$.

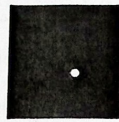


Fig. 3. Imagen $I(x,y)$, esfera sobre el plano.

La función de umbral fue implementada en el VHDL (Very High Speed Hardware Description Language) [3], que es un lenguaje mediante el cual es posible describir un circuito digital. Con ayuda del lenguaje, se tiene la posibilidad de abordar la descripción de una idea o diseño de forma estructural o comportamental [4]. La descripción estructural consiste en enumerar los componentes de un circuito y su interconexión. Con la descripción comportamental, es factible diseñar un circuito electrónico a partir de su comportamiento, en vez de los elementos de los que está conformado.

Para el caso de la función de umbral, se optó por la descripción comportamental mostrada en la Fig. 4. Los pixeles leídos de la cámara mayores que el valor de umbral B , corresponden al objeto y por debajo del umbral al fondo de la escena, lo que resultó en un circuito comparador combinatorio de 10bits.

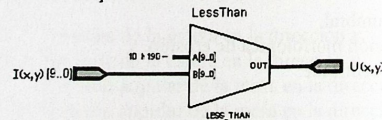


Fig. 4. Comparador de 10bits

3.2 Transformación morfológica de erosión.

La morfología se refiere al estudio de las formas y estructuras [5]. La morfología matemática descansa en la teoría de conjuntos, en el campo del procesamiento de imágenes, es empleada para la extracción de componentes, a su vez, es útil en la representación y descripción de la forma de una región.

Las definiciones básicas en morfología matemática son:

U conjunto de Z^2 cuyas componentes son $u = u(x,y)$.

- La traslación de U por un vector $a = a(m,n)$ se define como:

$$(U)_a = \{g \mid g = u + a, \text{ para } u \in U\} \quad (2)$$

- La reflexión de U se define como:

$$\hat{U} = \{a \mid a = -u, \text{ para } u \in U\} \quad (3)$$

- El complemento de U se define como:

$$U^c = \{a \mid a \notin U\} \quad (4)$$

El elemento estructurante es un conjunto de forma conocida, utilizado para la extracción de estructuras geométricas en los conjuntos sobre los que se opera. Su tamaño y forma va de acuerdo a la extracción de formas que se desea obtener.

La erosión y dilatación son las transformaciones morfológicas básicas sobre las cuales se construyen las demás.

U y W son conjuntos de Z^2 . La erosión de U con respecto del elemento estructurante W se define como:

$$U \otimes W = \{a \in Z^2 \mid a + w \in U, \forall w \in W\} \quad (5)$$

La transformación de erosión es el resultado de comprobar si el elemento estructurante W , pertenece completamente al conjunto U , de lo contrario el resultado es el vacío.

Considerando la imagen resultante de la función de umbral mostrada en la Fig. 5, se presentan píxeles que no pertenecen a la esfera, dichos píxeles podrían ser objetos en la escena si no conociéramos el origen de la imagen, pero en realidad son píxeles producto del reflejo de la iluminación en el plano. La imagen para ser procesada (cálculo del centroide de la esfera), requiere de eliminar todos los píxeles que no pertenecen al objeto de interés.



Fig. 5. Imagen binaria resultado de la función de umbral

Descripción de la transformación erosión [6]: Es una transformación que elimina los píxeles conexos con alguno que es parte del fondo, lo que supone eliminar una línea del contorno del objeto, esto en su forma más simple. El resultado de la

erosión puede ser expresado en la siguiente ecuación, usando los índices " m,n " como la posición de la componente conexa en la r -vecindad de la celda ' $u(x,y)$ '.

$$S(x,y) = \sum_{-r \leq m \leq r} \sum_{-r \leq n \leq r} w(m,n)u(x+m,y+n) \quad (7)$$

$$S(x,y) = f(S(x,y)) = \begin{cases} 0, & s(x,y) \neq i \\ 1, & s(x,y) = i \end{cases} \quad (8)$$

Donde ' $S(x,y)$ ' es el estado resultante de la celda $u(x,y)$. ' $u(x,y)$ ' es una componente de la imagen binaria proveniente de la función de umbral (1), para $0 \leq x \leq C$ y $0 \leq y \leq F$, donde ' C ' es el número de columnas y ' F ' el número de filas de la imagen. ' i ' es igual al número de pesos $w(m,n)$ con valor diferente de cero. ' r ' es la máxima distancia entre la celda de referencia $u(x,y)$ y sus vecinos $u(x+m,y+n)$. Si ' $r=1$ ', los índices (m,n) forman el siguiente conjunto $\{(-1,-1), (1,-1), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)\}$ y el elemento estructurante W , es definido con respecto a $w(m,n)$ como sigue:

$$W = \begin{bmatrix} w(-1,-1) & w(-1,0) & w(-1,1) \\ w(0,-1) & w(0,0) & w(0,1) \\ w(1,-1) & w(1,0) & w(1,1) \end{bmatrix} \quad (9)$$

' $w(m,n)$ ' son los pesos de control en la ecuación (7) que dan forma al elemento estructurante. El elemento estructurante (9) toma la forma y pesos de la ecuación (10) para realizar la tarea de erosión con $i=8$ y peso de referencia en $w(0,0)$.

$$W = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (10)$$

Comprendida la transformación de erosión expresada en (7) y (8), ahora puede ser descrita de forma estructural en VHDL. Para iniciar se define la siguiente notación:

$$\begin{array}{lll} A = u(x-1,y-1), & B = u(x-1,y), & C = u(x-1,y+1), \\ D = u(x+0,y-1), & E = u(x,y), & F = u(x+0,y+1), \\ G = u(x+1,y-1), & H = u(x+1,y), & I = u(x+1,y+1), \\ Q = q(x,y). \end{array}$$

Tabla 1: Tabla de verdad de $Q(x,y) = f(S(x,y))$

Entradas									Q
A	B	C	D	E	F	G	H	I	
1	1	1	1	1	1	1	1	1	1
*	*	*	*	*	*	*	*	*	0

* No importa

De la tabla 1, se obtiene la función Booleana (11), donde podemos observar que el operador lógico AND describe el comportamiento de la transformación morfológica de erosión:

$$Q = ABCDEFGHI \quad (11)$$

A partir de (11), se puede implementar el circuito digital equivalente mostrado en la Fig. 6 en un FPGA, lo que resulta en disminuir el costo computacional. Por otro lado un pre-procesamiento de imágenes binarias de este tipo, requiere de la convolución y un comparador en un equipo convencional (PC's), además de no contar con el ambiente concurrente que tiene VHDL.

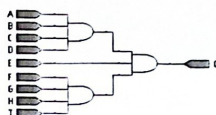


Fig. 6. Circuito equivalente de la transformación erosión.

3.3 Cálculo del centroide.

Este es el último bloque del módulo del sistema de visión, encargado de calcular el centroide de la esfera que se encuentra en el plano. Una vez que la imagen está libre de objetos que no pertenezcan a la esfera Fig. 8, es posible calcular su centroide. Los datos de la imagen que provienen del módulo de erosión se leen en un barrido progresivo de izquierda a derecha y de arriba hacia abajo, además se utilizó un contador para las columnas y uno para las filas, tomado estos dos como ejes de referencia (x,y) . Las coordenadas del centroide de la esfera representada por una región R de componentes conectados $q(x,y)=1$, pueden ser calculadas mediante los momentos geométricos estándar $m_{0,0}$, $m_{0,1}$, $m_{1,0}$ [7] de la forma siguiente:

$$Eje - X = \frac{m_{10}}{m_{00}} = \frac{\sum_{(x,y) \in R} xq(x,y)}{\sum_{(x,y) \in R} q(x,y)} \quad (12)$$

$$Eje - Y = \frac{m_{01}}{m_{00}} = \frac{\sum_{(x,y) \in R} yq(x,y)}{\sum_{(x,y) \in R} q(x,y)} \quad (13)$$

El centroide de la esfera será proporcionado al módulo de control, que calculará la inclinación necesaria para que la esfera ruede en la dirección apropiada y se estabilice en la posición deseada.

4 Resultados.

Para el sistema de visión se colocó el sensor digital de imágenes a una distancia fija sobre el plano de 0.638m. El tamaño del plano es de 0.5m x 0.5m, ver Fig. 7. El sensor digital de imágenes CMOS modelo MT9M011 [8], incorpora varias funciones de configuración, una de estas funciones especifica el tamaño de la imagen menor o igual al formato SXGA, la cámara fue configurada con una resolución de 1024H x 1024V para que la forma del plano coincida con la forma de la imagen, de esta manera se cubre toda el área de trabajo.



Fig. 7. Sistema plano-esfera

Debido al filtro de color Bayer [9] en el sensor, sería necesario interpolar para formar los planos de color verde, rojo y azul, con los que se forma una imagen a color. Para evitar esto, se utilizó el fondo negro y la esfera de color blanco, ya que el color blanco es la superposición de todos los colores, lo que no hace necesario la interpolación.

Se utilizó un FPGA Altera Cyclone II EP2C35F672C6 (el dispositivo se encuentra montado en la tarjeta DE2 de Altera Corporation) [10], en donde se implementaron las entidades que describen el comportamiento del módulo de visión. La primera tarea que realiza el módulo, es la de configurar el sensor digital de imágenes a través de una interfaz serie de dos vías. Una vez configurado el sensor, los datos de la imagen son enviados al FPGA en un barrido progresivo. Después, el módulo de captura se sincroniza con la cámara para recibir los datos y aplicar la función de umbral para segmentar. El ruido se elimina de la imagen a través de una transformación morfológica de erosión. Conforme este último módulo aplica el elemento estructurante a cada dato de la imagen, se detectan los píxeles que pertenecen a la esfera para determinar su centroide en el plano. Lo anterior, se describe en el lenguaje de descripción de hardware que es mostrado en forma de bloques en la Fig. 8.

En la etapa de diseño de los componentes que conforman al módulo de visión, la imagen resultante de cada componente es almacenada en una memoria RAM embebida en el FPGA. Con ayuda de Quartus II Versión 7.2 y con la herramienta In-System Memory Content Editor, se visualizó la imagen y verificó su correcto funcionamiento. En la Fig. 9, se muestran las imágenes resultantes del módulo de visión y en la Tabla 2. se presenta el cálculo de los centroides de algunas posiciones de la esfera sobre el plano.

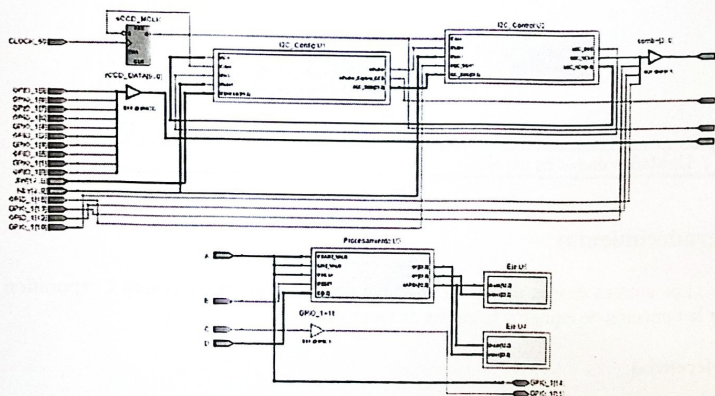
5 Conclusiones y trabajos futuros.

El procesamiento de imágenes binarias implementado en FPGA utilizado para el cálculo del centroide de la esfera dentro del plano, presenta una mínima desviación de un píxel con respecto de la posición esperada, debido a que las pruebas se realizaron en condiciones de iluminación naturales en el laboratorio; sin embargo, con iluminación controlada para el sistema PE, se espera mejorar el resultado.

Una vez que se terminó el módulo de visión y se verificó su correcto funcionamiento, se eliminó la memoria RAM embebida en el FPGA, ya que todo el procesamiento de la imagen capturada por la cámara se realiza en línea. El término en línea, nos dice que conforme se leen los datos de la imagen, estos son procesados en el FPGA sin necesidad de almacenarla en su totalidad, sólo se necesita almacenar tres líneas de la imagen en un registro de desplazamiento, por lo que para la obtención del centroide, se tiene un retraso del tiempo mínimo, lo cual representa una ventaja con respecto a equipos convencionales (PC's).

Como trabajo futuro continuará con la implementación y puesta a punto del controlador, del tipo Takagi-Sugeno, para cerrar el lazo de control basado en posición.

Utilizando la descripción empleada en la sección 3.2, se puede diseñar un circuito lógico que describa el comportamiento de la transformación morfológica de dilatación y realizar otro tipo de procesamiento.



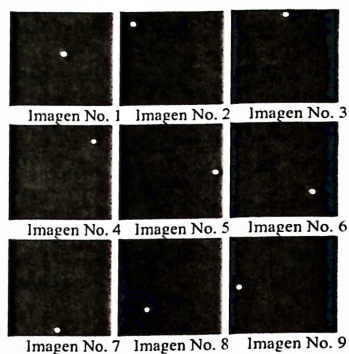


Fig. 9. Imágenes del sistema plano-esfera procesadas por el módulo de visión

Tabla 2. Cálculo de centroide de Fig. 9.

Imagen	Cálculo del centroide basado en imagen*	
	Esperado (x,y)	Módulo de visión (x,y)
1	(534,462)	(534,462)
2	(131,142)	(132,143)
3	(526,62)	(526,63)
4	(832,182)	(832,181)
5	(946,526)	(946,525)
6	(788,752)	(788,752)
7	(508,958)	(508,958)
8	(288,752)	(289,752)
9	(104,536)	(104,536)

*. Unidades dadas en pixeles

Agradecimientos

Los autores desean agradecer el apoyo de Víctor Maruri y Altera Corporation por la donación de equipo y licencias de Quartus II.

Referencias

1. Sánchez Martínez Miguel A., *Diseño de un circuito comparador de imágenes*. CINVESTAV del I. P. N., Departamento de ingeniería eléctrica, Sección de computación, México D.F., Julio 2005.
2. Hung T. Nguyen, Michio Sugeno, Richard Tong, Ronald R. Yager, *Theoretical aspects of fuzzy control, chapter 2*, John Wiley & Sons, Inc., February 1995.

3. IEEE Computer Society, *IEEE Standard VHDL Language Reference Manual*, pag. i – 290, ISBN: 0-7381-1948-2, January 2000.
4. Fernando Pardo Carpio, José A. Boluda Grau, *VHDL: Lenguaje para síntesis y modelado de circuitos*, Alfaomega RA-MA, 1999.
5. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, *Digital Image Processing using Matlab*, pag. 335-347, Prentice Hall, 2004.
6. Hsin-Chieh Chen, Yung-Ching Hung, Chang-Kuo Chen, The-Lu Liao, Chun-Kuo Chen. *Image-processing algorithms realized by discrete-time cellular neural networks and their circuit implementations*. Department of Engineering Science, National Cheng Kung University, Tainan 701, Taiwan, August 2005.
7. M.K. Hu, *Visual pattern Recognition by Moment Invariants*, IRE Trans. on Information Theory, vol. 8, pag. 179-187, February 1962.
8. Micron Technology, Datasheet MT9M011, 1/3-Inch Megapixel CMOS Active-Pixel Digital Image Sensor.
9. B. E. Bayer, *Color imaging array*, US Patent No. 3971065.
10. User manual, *Development and Education Board*, version 1.3, pag. 24-53, Altera Corporation, 2006.