

Teddy V. Rojas Z.⁽¹⁾, Wilmer Sanz F.⁽²⁾, Francisco Arteaga B.⁽³⁾

e-mails: tvrojas@uc.edu.ve wsanz@uc.edu.ve farteaga@uc.edu.ve

Se tienen tres puntos dados **A(1,2)** , **B(2,4)** y **C(3,6)**. Para cada punto existe una familia de rectas que pasan por él, de tal manera que con las coordenadas **(x,y)** del punto se generan combinaciones de los parámetros **a** y **b** de la familia de rectas **y = ax + b** que pasan por el punto en particular.

Tomando las coordenadas de cada punto y reordenando la ecuación de la recta en términos de los parámetros, cada punto produce una función paramétrica respectiva como sigue:

$$\begin{aligned} b &= f_1(a) = 2 - 1a \\ b &= f_2(a) = 4 - 2a \\ b &= f_3(a) = 6 - 3a \end{aligned}$$

Todo par de valores (a, b) de cada función paramétrica, con a variando de 1 a 5 por ejemplo (en relación al tamaño del mapa de bits), se traduce en votos dentro del arreglo de acumulación como se observa en las tablas 1 y 2.

Tabla 1. Votos para $b = f_1(a)$

a/b	-6	-5	-4	-3	-2	-1	0	1	2	3
1	0	0	0	0	0	0	0	1	0	0
2	0	0	0	0	0	0	1	0	0	0
3	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	0

Tabla 2. Votos para $b = f_1(a)$ y $b = f_2(a)$

a/b	-6	-5	-4	-3	-2	-1	0	1	2	3
1	0	0	0	0	0	0	0	1	1	0
2	0	0	0	0	0	0	2	0	0	0
3	0	0	0	0	1	1	0	0	0	0
4	0	0	1	0	1	0	0	0	0	0
5	1	0	0	1	0	0	0	0	0	0

Se observa en este resultado parcial que para $a = 2$ y $b = 0$, cuya celda en el arreglo de acumulación tiene la mayor cantidad de “votos”, corresponde a la recta $y = 2x + 0$ que conecta a los dos primeros puntos **A** y **B**.

Sucede que cuando se trazan todas las rectas posibles que pasan por el punto **A**, algunas también pasarán por **B** y por **C**. Si esto ocurre, las celdas del arreglo de acumulación se incrementan apropiadamente según las rectas que pasen a través de múltiples puntos. Al final del proceso de acumulación, la celda que tenga el mayor conteo de votos da la recta que pasa más cerca de la mayor cantidad posible de puntos alineados en la imagen de origen.

La Transformada de Hough plantea entonces que es en el espacio de los parámetros a y b o espacio de Hough, donde se determina la recta característica de una línea de puntos dados. Sin embargo, esta formulación tiene algunos inconvenientes numéricos: para rectas horizontales, cuando $a = 0$ y con rectas verticales, cuando $a \rightarrow \infty$. Para evitar estos inconvenientes se ha de reformular la transformada en la representación de coordenadas polares o radio-ángulo tal como se explica en la sección 2.2

2.2 Formulación en coordenadas polares

La transformada de Hough universalmente utilizada en la actualidad fue desarrollada por Richard Duda y Peter Hart en 1972 (Lee, 2006), quienes sugirieron el uso de los parámetros polares radio-ángulo en lugar de los parámetros pendiente-ordenada, para la representación de las rectas (Fig. 2)

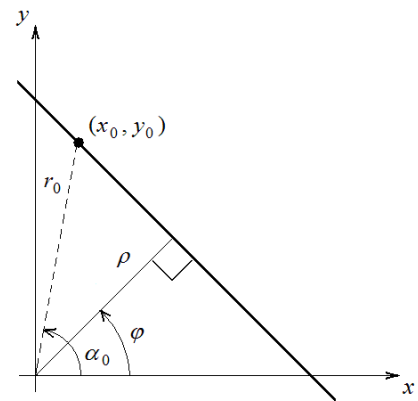


Fig. 2. Formulación en coordenadas polares

La nueva forma de parámetros de la recta es,

$$\rho = x \cos \varphi + y \sin \varphi \quad (1)$$

Con esta otra definición de parámetros se tiene el rango natural de $\varphi \in [0, 2\pi]$. Luego, se transforma cada punto (x, y) de la imagen de origen, en los puntos (φ_i, ρ_i) del espacio de Hough mediante la función de la recta en coordenadas polares. Cuando se genera la familia de rectas coincidentes para un punto en particular, se obtiene una curva sinusoidal (Fig. 3)

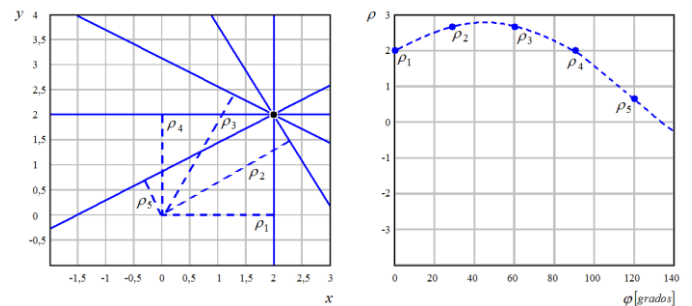


Fig. 3. Transformada de Hough de un punto

Supóngase ahora que se tienen los tres puntos **A(0,2)** , **B(2,0)** y **C(1,1)** en la imagen de origen. Con la transformada de estos tres puntos se producen las tres sinusoides indicadas a continuación,

$$\begin{aligned} \text{Para } \mathbf{A(0,2)} & \rightarrow \rho = 2\text{Sen}\varphi \\ \text{Para } \mathbf{B(2,0)} & \rightarrow \rho = 2\text{Cos}\varphi \\ \text{Para } \mathbf{C(1,1)} & \rightarrow \rho = \text{Cos}\varphi + \text{Sen}\varphi \end{aligned}$$

cuya gráfica es (Fig. 4)

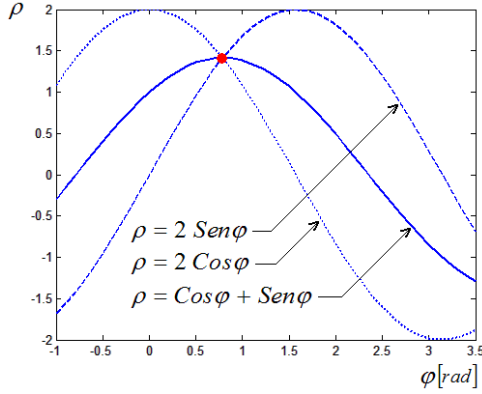


Fig. 4. Transformada de Hough de tres puntos

Se observa que las tres sinusoides se intersectan en $(\varphi, \rho) = (\pi/4, \sqrt{2})$ en el espacio de Hough, lo que significa que la recta que une los tres puntos **A** , **B** y **C** es $\sqrt{2} = x\text{Cos}(\pi/4) + y\text{Sen}(\pi/4)$, o lo que es igual a $x + y = 2$ en el espacio cartesiano original. Esta última operación se conoce como la Transformada Inversa de Hough.

2.3 Definiciones básicas de la transformada

Función de Parámetros de la Transformada: Es como se ha de describir el tipo o forma de las líneas a detectar en una imagen (rectas, círculos, elipses, etc.). En general, el tipo de línea a detectar puede ser descrito implícitamente por la ecuación,

$$f(x, y, \alpha_1, \alpha_2, \dots, \alpha_n) = 0 \quad (2)$$

o descrito explícitamente por la función

$$y = f(x, y, \alpha_1, \alpha_2, \dots, \alpha_n) \quad (3)$$

donde **x** e **y** son las coordenadas cartesianas de los puntos en la imagen, mientras que $\{\alpha_1, \alpha_2, \dots, \alpha_n\}$ es el conjunto de **n** parámetros que especifican el aspecto o tipo de líneas a detectar.

Transformada Directa de Hough: Cada punto (x, y) en el espacio de dos dimensiones de la imagen, es transformado en una hiper-superficie en el espacio de parámetros de Hough **n**-dimensional.

Transformada Inversa de Hough: Cada punto $(\alpha_1, \alpha_2, \dots, \alpha_n)$ en el espacio de parámetros describe una instancia específica del tipo de línea de interés en el espacio bidimensional de la imagen.

Sobre la base de lo anteriormente señalado, cuando la descripción de parámetros utilizada corresponde específicamente a la detección de líneas rectas, se puede afirmar que la intersección de sinusoides en el espacio de Hough determina los parámetros de la recta que une ciertos puntos en el espacio de la imagen, sólo aquellos puntos que han generado las sinusoides intersecadas.

2.4 Transformada circular de Hough

El nombre dado a la transformada se deriva del tipo específico de figuras a detectar en la imagen: *Círculos* (Rizon, et al, 2005). Para ello, la descripción de parámetros utilizada es,

$$f(x, y, x_0, y_0, r) = (x - x_0)^2 + (y - y_0)^2 - r^2 \quad (4)$$

donde la terna (x_0, y_0, r) conforma el conjunto de parámetros que expande el espacio de Hough a uno tridimensional; luego el arreglo de acumulación $H(x_0, y_0, r)$ será de dimensión tres.

La dirección del gradiente $\angle G$ puede ser utilizada para reducir los cálculos en el espacio de parámetros de Hough (Borovicka, 2003; Calic, 2005) y por tanto tener una más eficiente detección, limitando el proceso de votación sólo para los valores de los parámetros que sean más consistentes con la orientación de los bordes de las figuras buscadas en una imagen.

Así, en lugar de seleccionar votos para todas las rectas que pasan por un punto determinado, se puede escoger votar solamente por aquellas cuya orientación φ esté dentro de cierto rango $\phi - a \leq \varphi \leq \phi + a$, donde ϕ es la dirección del gradiente. Téngase en cuenta que la dirección del gradiente siempre es perpendicular a la dirección tangente de una curva dada por $f(x, y) = 0$, como se muestra en la figura 5 (Calic, 2005).

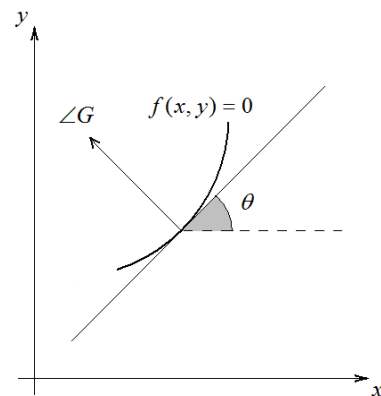


Fig. 5. Dirección del gradiente

De la figura se establece que,

$$\tan \theta = \frac{dy}{dx} = -\frac{f'_x(x,y)}{f'_y(x,y)} \equiv g(x,y) \quad (5)$$

siendo, $\phi = \angle G$ y $\theta = \phi \pm \pi/2$

En forma general, para cualquier tipo de función de parámetros resulta:

$$\begin{cases} f(x,y,\alpha_1,\alpha_2,\dots,\alpha_n) = 0 \\ g(x,y,\alpha_1,\alpha_2,\dots,\alpha_n) = -\cot \phi \end{cases}$$

Cualquier punto (x,y) en la imagen de origen, corresponde con una superficie en forma de cono en el espacio tridimensional, tal como se observa en la figura 6

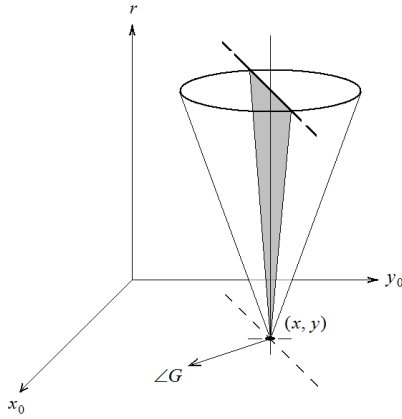


Fig. 6. Espacio 3D de la Transformada Circular

Sobre la base de esta figura y teniendo en cuenta la dirección gradiente $\angle G$ en los cálculos, según la ecuación 5, se tiene:

$$\begin{cases} (x - x_o)^2 + (y - y_o)^2 - r^2 = 0 \\ (x - x_o)/(y - y_o) = \cot \phi \end{cases} \rightarrow \begin{cases} x_o = x \pm r \cos \phi \\ y_o = y \pm r \sin \phi \end{cases} \quad (6)$$

$$\phi = \angle G = \tan^{-1} \left(\frac{y_o - y}{x_o - x} \right) \quad (7)$$

Ahora el algoritmo de detección de círculos será:

- I. Iniciar el arreglo de acumulación bidimensional $H(\phi, \rho)$ del espacio de parámetros de Hough, cuya cantidad de elementos va en función de la resolución de la imagen de origen.
- II. Para cualquier píxel que satisfaga $|G(x,y)| > T_g$ (valor de referencia del campo gradiente que es ajustable experimentalmente), que cumplan con las dos ecuaciones simultáneas:

$$\text{Para todo } r \text{ tal que: } \begin{cases} x_o = x \pm r \cos \phi \\ y_o = y \pm r \sin \phi \end{cases}$$

incrementar los elementos correspondientes del arreglo de acumulación, $H(x_o, y_o, r) = H(x_o, y_o, r) + 1$

- III. Cualquier elemento $H(x_o, y_o, r) > T_h$ (umbral de detección, también asignado experimentalmente) representará un círculo de radio r y centro en (x_o, y_o) contenido en la imagen de origen.

3. LA INTERFAZ VISION_XR3

En el diseño de la interfaz gráfica **Vision_XR3** en MATLAB para la aplicación de la Transformada Circular de Hough, es necesario tener cargado los complementos utilitarios siguientes:

Image Acquisition Toolbox,
Image Processing Toolbox
Signal Processing Toolbox.

La interfaz responde al diagrama de flujo de la figura 7

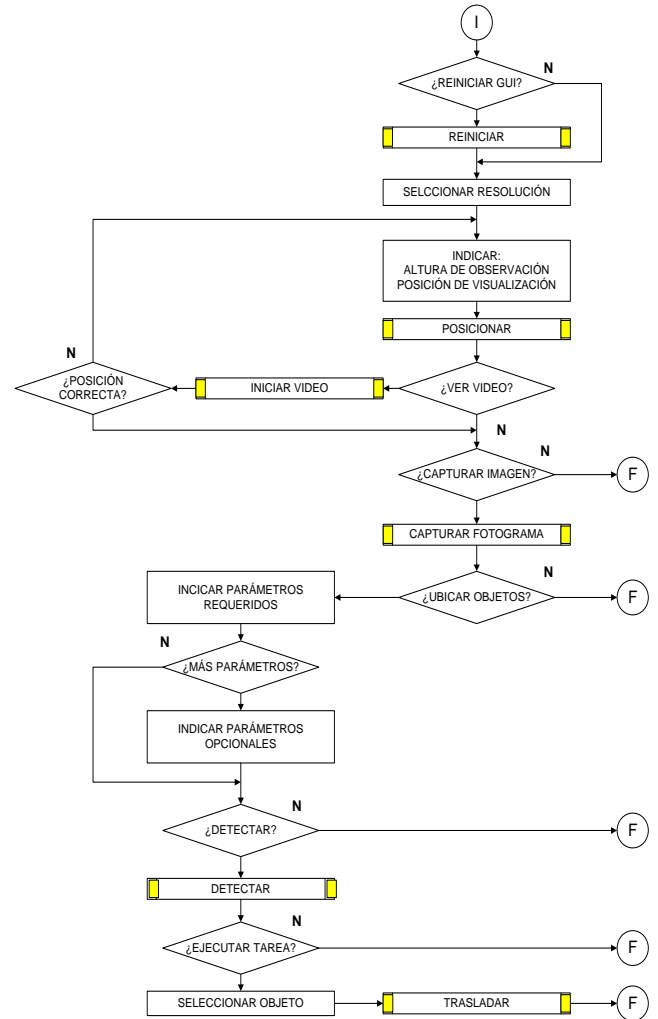


Fig. 7. Diagrama de flujo para Vision_XR3

En el diagrama de flujo se pretende destacar los ‘botones’ de ejecución presentes en la interfaz de usuario mediante rectángulos de extremos sombreados, para diferenciarlos de los bloques de acción generales. Luego, en la ventana de comandos de MATLAB, al invocar el comando **>>Vision_XR3**, aparece la interfaz mostrada en la figura 8

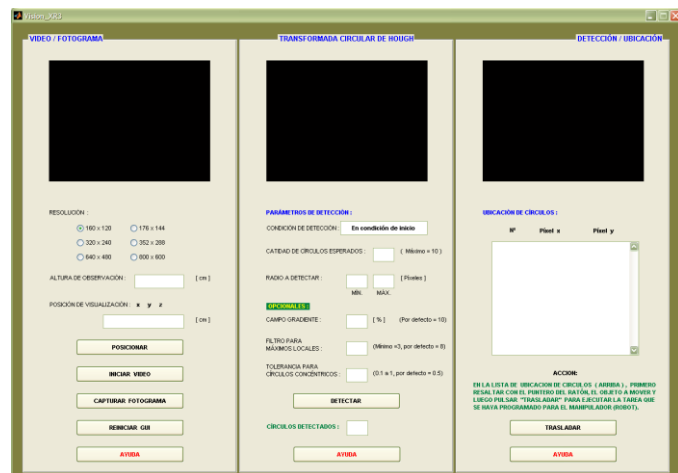


Fig. 8. Interfaz gráfica de usuario Vision_XR3.

La interfaz consta de tres paneles que agrupan las tareas que se han de seguir para lograr la ubicación de un objeto circular detectado. Estos paneles están identificados con los títulos:

VIDEO/FOTOGRAMA TRANSFORMADA CIRCULAR DE HOUGH DETECCIÓN/UBICACIÓN

Cada uno tiene un área rectangular a manera de visor, donde se puede ver lo que en el panel está ocurriendo, según las tareas ejecutadas. La idea es que el usuario siga una secuencia natural de pasos con un mínimo de de confusión.

El primer panel de tareas contiene las funciones requeridas para la configuración de la resolución de imagen, el inicio del video y la captura de un fotograma para procesar.

La resolución por defecto cuando se inicia la interfaz es de **160X120 píxeles**. Después de instalar el *controlador* suministrado por el fabricante de la cámara a utilizar, se sugiere que se establezca el funcionamiento monocromático para el dispositivo con la finalidad que la Transformada Circular de Hough procese una matriz de píxeles en la escala de grises (0 a 255 de intensidad, 0= Negro) en lugar de tres para el sistema de color RGB. Como el diseño de la interfaz gráfica supone un sistema de visión artificial donde la cámara estaría colocada verticalmente para tomar imágenes de un plano de trabajo, se proveen dos campos para asignar los valores de la altura de observación (desde la cámara al plano de trabajo) y la posición de observación, en coordenadas **(x, y, z)** respecto al sistema cartesiano en la base del soporte, todo en centímetros. Las coordenadas de posición de observación se suministran con los tres valores separados por espacios. El panel cuenta con los botones ‘INICIAR VIDEO’ para activar la cámara y ‘CAPTURAR FOTOGRAMA’ para

la obtención de una imagen digital. La transformada requiere de un archivo matricial específico de la imagen para su procesamiento. La creación, utilización y eliminación de ese archivo en el área de trabajo de MATLAB es totalmente transparente para el usuario.

El panel central es el “núcleo” de la interfaz y por supuesto de toda la aplicación. Aquí se ejecuta el código que detecta y ubica los objetos circulares presentes en una imagen digital, mediante la Transformada Circular de Hough.

Mientras en el panel de tareas “VIDEO/FOTOGRAMA” esté presente un fotograma capturado, se puede modificar cualquier parámetro para la transformada y realizar la detección. El fotograma es eliminado una vez culminado el proceso de detección. En este panel central están dispuestos seis campos para parámetros, tres requeridos y tres opcionales: Cantidad de círculos esperados, Rango (valor mínimo y máximo) del radio a detectar, Campo gradiente, Filtro para máximos locales y Tolerancia para círculos concéntricos. La cantidad de círculos esperados provee una cifra límite al programa para que cuando, ante imágenes muy complejas, se detecta una gran cantidad de círculos (muchos de ellos inexistentes), se puede dar el resultado de falla de detección de círculos reales. Un caso típico de esta situación ocurre cuando se intenta procesar una escena vacía (que no es lo mismo que una imagen en blanco con todos sus píxeles en 255): el programa puede devolver cantidades excesivas de círculos detectados (30, 40 o más). No es un defecto del algoritmo; recuérdese que la transformada intenta detectar contornos circulares mediante campos gradientes y estos no están claramente definidos en imágenes complejas o vacías. Por otro lado, la especificación de rango de los radios de los círculos a detectar provee la restricción necesaria para hacer más eficiente la búsqueda de los objetos en la imagen.

El primero de los parámetros opcionales es el campo gradiente, que establece el límite inferior para que el cambio de intensidad entre dos píxeles contiguos en la imagen no sea considerado como parte del contorno de un objeto: cambio de intensidad menor a este límite se considera como una continuidad de fondo de la imagen. El segundo parámetro opcional es el valor del radio del filtro aplicado para la búsqueda de máximos locales en el arreglo de acumulación de la transformada. El último parámetro opcional tiene que ver con la capacidad del programa para detectar círculos concéntricos presentes en una imagen dada.

Es en el tercer panel donde se presentan los resultados de la detección de objetos y se puede optar por la ejecución de la tarea de traslado de un objeto seleccionado. Este panel muestra la ubicación en píxeles con respecto a la esquina superior izquierda de la imagen, de los objetos circulares detectados. Una vez concluida la detección, quedan en el *WorkSpace* o área de trabajo de MATLAB cinco matrices, cuales son: 1) **accum** que es el *Arreglo de Acumulación* de la transformada, donde se encuentran los votos para los centros de los círculos detectados; se deja disponible al usuario para cualquier análisis que se quiera realizar a posteriori. 2) **circen** corresponde a las coordenadas de los centros de los círculos detectados (en píxeles) tal como se muestra en el panel de tareas, mientras que 3) **circen_base** y 4) **circen_imagen** son matrices que contienen las coordenadas de los mismos centros, en centímetros, referidos

a la base del robot y al el centro del plano de observación, respectivamente. Por último, 5) **cirrad** contiene los valores de los radios de los círculos detectados. Estos radios son necesarios para “dar volumen” a las coordenadas de los objetos, con la finalidad de que las pinzas del robot alcancen la altura correcta para tomar un objeto. El botón ‘TRASLADAR’ genera un archivo de datos de simulación para comprobar el funcionamiento de robots virtuales con el programa RoboWorks (Franco, Rodríguez, 2005), dado que la aplicación tiene una clara orientación a ser un sistema de detección de objetos para la robótica (Sanz, 2006; Craig, 2006).

4. ENSAYOS CON VISION_XR3

4.1 Ensayo sobre una una imagen plana

En esta sección se presentan algunos ensayos hechos con la interfaz **Vision_XR3**, con figuras que muestran los resultados que se obtuvieron en la experiencia. La captación de imágenes se logró a través de una cámara web adherida al elemento terminal de un robot antropomórfico de 5 grados de libertad (RHINO XR-3), ya que el objetivo original de esta investigación era dotar al nombrado robot de un sistema de visión artificial (Sanz, 2006). Primeramente se ha de seleccionar la resolución a utilizar. Cuando se inicia el GUI, la resolución por defecto es **160x120 píxeles**. Un aspecto importante a considerar es que a menor resolución, más rápido es el procesamiento por el algoritmo de la transformada programado, sin embargo aumenta la posibilidad de que no sean detectados círculos realmente existentes en la imagen. Por el contrario, a mayor resolución se mejora la cantidad de detección, tanto que es posible que se detecten círculos aparentemente inexistentes, además de aumentar el tiempo de procesamiento de la transformada. Un ejemplo de lo descrito se muestra en las figuras 9 y 10.

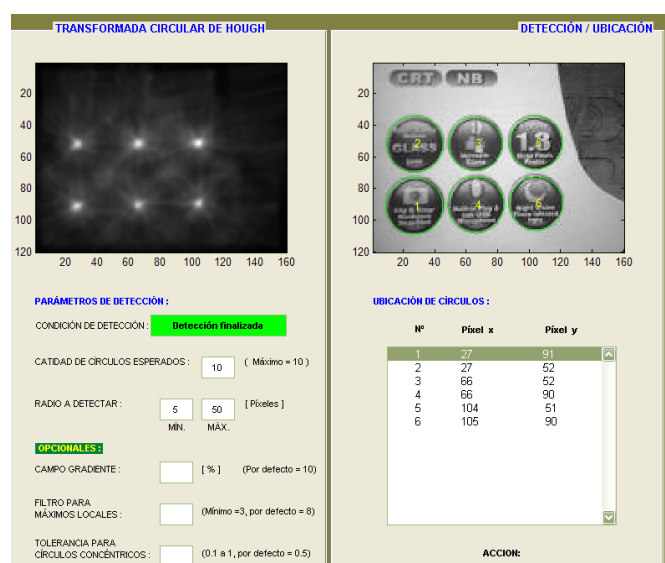


Fig. 9. Seis círculos detectados a una resolución de **160x120 píxeles**

La aplicación de la transformada a un fotograma capturado con una resolución de **160x120 píxeles** produjo la detección de seis círculos claramente visibles en la escena. La imagen es compleja en cuanto a presentar diversas formas y patrones (realmente es la imagen capturada de una sección de la caja de empaque de la cámara web utilizada). Ciertamente existen más formas circulares en el fotograma, pero a la resolución utilizada y con los parámetros por defecto dados al momento de correr el programa de detección, sólo los seis círculos mayores o más “evidentes” son detectados (Fig. 9).

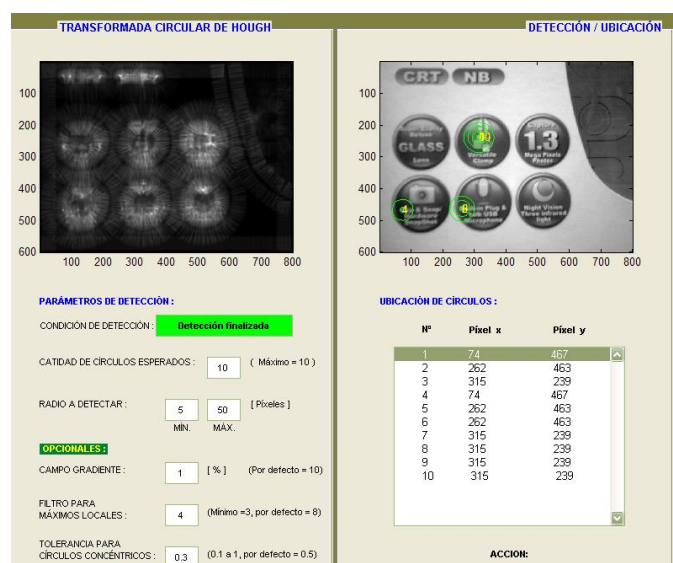


Fig. 10. Diez círculos detectados con una resolución de **800x600 píxeles**

Manteniendo la misma escena pero usando la mayor resolución de **800x600 píxeles**, la transformada con el nuevo fotograma, produce la detección de diez círculos todos aparentemente inexistentes (Fig. 10). Nótese que para lograr alguna detección “razonable” se cambiaron los valores de los parámetros opcionales, ya que al dejarlos en los valores por defecto, ocurren detecciones de hasta más de cien círculos. La experiencia indica que a la menor de las resoluciones y con los valores por defecto para los parámetros opcionales, la detección funciona bien para casi todas las condiciones de operación.

4.2 Ensayo de detección con una esfera real

En este ensayo se tiene la detección de un objeto esférico real, colocado junto a otros objetos tridimensionales comunes, según se muestra en las figuras 11 y 12 (se presentan los visores central y derecho de **Visión-XR3** simultáneamente, con la detección lograda a dos resoluciones distintas). La práctica indica que siempre se puede lograr un ajuste adecuado de los parámetros opcionales de la transformada para detectar una forma circular principal presente en el fotograma, no obstante lo compleja que pueda

parecer la escena. También la experiencia muestra que a mayor resolución se hace más difícil el ajuste óptimo de esos parámetros y la tendencia es a tener una detección exagerada de círculos, quizás irreales.

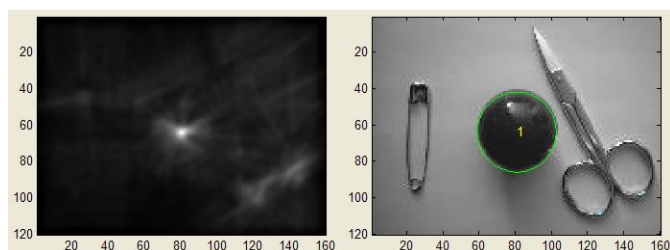


Fig. 11. Detección de una esfera real a la resolución de **160x120 píxeles**

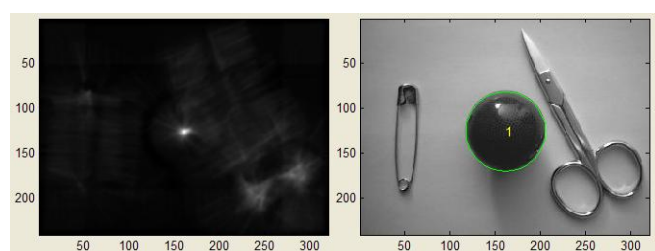


Fig. 12. Detección de una esfera real a la resolución de **320x240 píxeles**

El visor da la izquierda en ambas figuras muestra el espectro generado por la ejecución de la transformada y se pueden notar trazos difusos de los posibles centros de objetos circulares detectables; es el ajuste de los parámetros opcionales lo que evita que sean identificados como círculos presentes en la imagen. Resulta inútil pretender conseguir una más precisa ubicación de los objetos con mayores resoluciones de la imagen; en la práctica esto es irrelevante según el tamaño de los objetos a manejar y dada la propia resolución del movimiento del brazo del robot (también existe una resolución o “sensibilidad” para el movimiento de las articulaciones de un robot). En todo caso es seguro que a mayor resolución, mayor es el tiempo de procesamiento del programa y mayor el requerimiento de ajuste de los parámetros opcionales de la transformada.

De esta manera se logra comprobar el correcto funcionamiento de la interfaz gráfica de usuario **Vision-XR3** diseñada como sistema de visión artificial para la detección de objetos esféricos mediante la Transformada Circular de Hough.

5. CONCLUSIONES

Se comprobó la eficacia de la Transformada Circular de Hough como método numérico para la detección de objetos esféricos presentes en una imagen digital.

Se verificó la rapidez de la Transformada, siendo el tiempo empleado para la detección del orden de las décimas de

segundos, a la más baja resolución y con un computador personal con procesador Intel Centrino® de doble núcleo.

También se verificó que aumentar la resolución empleada no mejora sensiblemente la exactitud de la ubicación de los objetos, teniéndose como consecuencia directa un aumento del tiempo de procesamiento de la Transformada y una mayor dificultad de alcanzar un ajuste óptimo de los parámetros de detección.

REFERENCIAS

- Aguado, A. y Nixon M. (1995). A new Hough Transform mapping for ellipse detection. *University of Southampton Research Journal*. <http://www.ecs.soton.ac.uk/publications/rj/1995-1996/isis/asag93r/albrj.htm>
- Borovicka, J. (2003). Circle Detection using Hough Transforms. *University of Bristol*. U.K. <http://linux.fjfi.cvut.cz/~pinus/bristol/imageproc/hw1/report.pdf>
- Calic, J. (2005). Lecture on the Hough Transform. *University of Bristol*. U.K. <http://www.cs.bris.ac.uk/Teaching/Resources/COMS30121/slidesIP/Lecture04.pdf>
- Craig, J. (2006). *Robótica*. Prentice-Hall
- De La Escalera, A. (2001). *Visión por Computador*. Prentice-Hall
- Franco, A., Rodríguez, C. (2005). *Modelación y Animación con el RoboWorks*. Universidad EAFIT. Colombia
- González, J. (2000). *Visión por Computador*. Paraninfo
- Lee, K. (2006). Application of the Hough Transform. *University of Massachusetts Lowell*. http://teaching.cs.uml.edu/~heines/techrpts/Papers/LeeK_yewook_HoughTransform.pdf
- Peng, T. (2006). Detect circles with various radii in grayscale image via Hough Transform. *Matlab Central*. <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=9168>
- Rizon, M., Yaszid, H. y Otros. (2005). Object detection using Circular Hough Transform. *American Journal of Applied Sciences*. <http://www.scipub.org/fulltext/ajas/ajas2121606-1609.pdf>
- Sanz, W. (2006). Apuntes sobre Robótica y Visión Industrial. *Universidad de Carabobo*
- The MathWorks. (2006). *Image Acquisition Toolbox User's Guide*. MATLAB
- Torres, F. y Pomares J. (2002). *Robots y Sistemas Sensoriales*. Prentice-Hall