

Detección de Círculos Basada en la Orientación de Bordes y Triángulos Rectángulos

Alejandro Nava Acosta

Universidad Autónoma de Ciudad Juárez, Instituto de Ingeniería y Tecnología,
Av. Del Charro 450 Nte. Col. Partido Romero, C. P. 32310, (656) 688-21-00 ext. 4781
alex_bravery@hotmail.com

Francisco Javier Enríquez Aguilera

Universidad Autónoma de Ciudad Juárez, Instituto de Ingeniería y Tecnología,
Av. Del Charro 450 Nte. Col. Partido Romero, C. P. 32310, (656) 688-21-00 ext. 4781
fenrique@uacj.mx

Juan Cota-Ruiz

Universidad Autónoma de Ciudad Juárez, Instituto de Ingeniería y Tecnología,
Av. Del Charro 450 Nte. Col. Partido Romero, C. P. 32310, (656) 688-21-00 ext. 4781
jcota@uacj.mx

Ernesto Sifuentes de la Hoya

Universidad Autónoma de Ciudad Juárez, Instituto de Ingeniería y Tecnología,
Av. Del Charro 450 Nte. Col. Partido Romero, C. P. 32310, (656) 688-21-00 ext. 4781
esifuent@uacj.mx

Pablo Rivas-Perea²

Baylor University, One Bear Place #97356, C. P. 76798, Waco, TX
Pablo_Rivas_Perea@Baylor.edu

Resumen

La Transformada de *Hough* para Círculos es el método estándar para la detección de círculos en imágenes. Debido a sus desventajas, varias modificaciones y nuevos algoritmos han sido sugeridos. En este trabajo se presenta un método eficaz de detección de círculos basado en la orientación de bordes y triángulos rectángulos dentro de un

círculo. El método propuesto disminuye el requisito de almacenamiento y aumenta la velocidad para encontrar círculos en una imagen utilizando una técnica de segmentación y análisis entrelazado. Algunas imágenes sintéticas con diferentes niveles de ruido y algunas naturales se han utilizado para probar el rendimiento del método propuesto. Los resultados experimentales muestran que el método propuesto es más rápido y espacialmente más compacto que la Transformada de *Hough* para Círculos.

Palabras Claves: Círculos, Detección, *Hough*, Orientación.

1. Introducción

La detección de círculos en una imagen es parte fundamental en aplicaciones de procesamiento digital de imágenes, ya que puede ser utilizada para proporcionar la ubicación o la cantidad de objetos circulares en diversas aplicaciones industriales. Se han desarrollado varios algoritmos para la detección de círculos: la *Transformada de Hough* para Círculos (CHT por sus siglas en inglés), es uno de los algoritmos más utilizados. En éste algoritmo se obtienen primeramente los bordes de la imagen; posteriormente para cada punto de borde se traza un círculo de radio r . Para círculos de radio desconocido el algoritmo debe trazar círculos de todos los radios posibles, lo cual crea un arreglo acumulador de 3 dimensiones, donde 2 de esas representan el centro de los círculos y la tercera representa el radio. Los círculos trazados se traslaparán en el centro del círculo a detectar cuando el radio de ambos sea el mismo [1].

Las principales desventajas que presenta la CHT son la gran cantidad de almacenamiento y de procesamiento requeridos. Se han hecho varias modificaciones para mejorar la CHT. Xu *et al.* Presentaron una nueva propuesta que selecciona aleatoriamente 3 píxeles de borde no colineales y traza círculos en dichos píxeles [2]. Chen y Chung mejoraron el método de Xu *et al.*, seleccionando 4 píxeles de forma aleatoria [3].

Por otra parte, se han desarrollado varios algoritmos que no se basan en la CHT, sino en conocimientos previos de la imagen, y en características geométricas de los círculos. En

1995 Ho y Chen utilizaron las características geométricas del círculo para trazar 2 líneas ortogonales que se intersectan en el centro del círculo de la imagen [4]. Shang et al., detectaron el centro de un círculo haciendo triángulos rectángulos dentro de éste [5]. Por último, Rad et al., utilizaron la orientación de los bordes de la imagen para generar vectores pares de gradientes y comparar 2 ángulos que se forman a partir de dichos vectores para verificar que crucen por el centro del círculo [6].

Esta investigación toma como idea principal la orientación de los bordes y triángulos rectángulos dentro de un círculo para poder identificar objetos circulares, así como también plantea una técnica de segmentación y análisis entrelazado para reducir el tiempo de procesamiento, y otra técnica que utiliza las características geométricas de un círculo para reducir el almacenamiento en memoria.

Este trabajo está organizado de la siguiente manera. La sección 2 describe detalladamente el método propuesto. Los resultados experimentales son descritos en la sección 3. En la sección 4 se presenta la discusión sobre el método propuesto y por último, la conclusión y los planes para trabajo futuro se presentan en la sección 5.

2. Desarrollo

En esta sección se presenta el algoritmo basado en la orientación de los bordes y triángulos rectángulos. En un círculo que es más oscuro o más claro que el fondo, las direcciones de los bordes apuntan hacia adentro y fuera del círculo. Los siguientes párrafos describen los pasos principales del método propuesto.

a) Obtención de la orientación de bordes

El primer paso es obtener la orientación de los bordes de la imagen, para esto es necesario obtener dos gradientes de la imagen original: el Gradiente X y el Gradiente Y . Para la obtención de los gradientes se utilizan las dos máscaras de 3×3 del operador *Sobel* [7]. Una vez obtenidos los gradientes, se puede calcular la orientación de cada pixel de la imagen a partir de la siguiente ecuación:

$$\theta = \tan^{-1} \left(\frac{GY}{GX} \right) \quad (1)$$

Después, para eliminar información innecesaria de la imagen, se utiliza la detección de bordes. El método que se utiliza para detectar los bordes de la imagen es el de *Canny* [8]. Posteriormente a cada punto de borde se le asigna el valor que tiene la imagen de las orientaciones (ángulos), calculadas en ese mismo punto, mientras que al resto de la imagen donde no se encuentra un punto de borde, se le asigna un valor mayor a 90 para evitar que se confunda con un posible ángulo.

Debido a sus características geométricas, un círculo tiene pares de ángulos con el mismo valor (ver Fig. 1). Si se traza una línea entre un par de direcciones con el mismo valor, ésta cruzará por el centro del círculo.

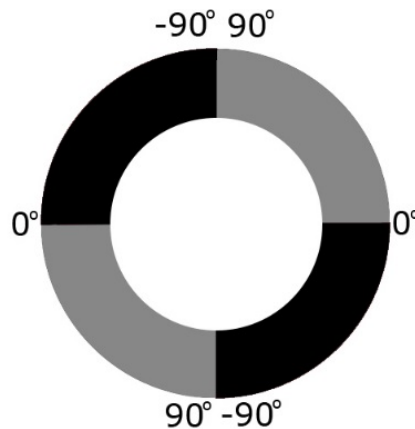


Fig. 1. Ángulos de un círculo obtenidos a partir de los gradientes.

b) Segmentación de la imagen

El siguiente paso es buscar pares de puntos de borde con el mismo valor de ángulo y realizar triángulos rectángulos entre éstos. Para esto primeramente se busca, de arriba hacia abajo y de izquierda a derecha, un punto de borde en la imagen y se almacena su posición; posteriormente se hace una segunda búsqueda de un segundo punto que

contenga el mismo valor que el primero para poder formar un triángulo rectángulo. Para disminuir el tiempo de procesamiento, se propone un análisis entrelazado para encontrar el segundo punto, esto es, buscar en columnas impares.

Además, para círculos cuyo radio sea menor a un tercio del lado menor de la imagen, se propone segmentar la imagen en nueve partes, para no tener que realizar la búsqueda en todas las filas y con esto disminuir aún más el tiempo de procesamiento. Se declara una constante para indicar el tamaño de los segmentos, a la que se le llama Rango. De esta manera, según el segmento en el que se encuentre el primer punto, se crea una ventana de tamaño de Rango para realizar la búsqueda del segundo punto dentro de ella únicamente (ver Fig. 2). Mientras más grande sea el rango de búsqueda los segmentos centrales se reducen y los cuatro segmentos de las esquinas se expanden (el rango de búsqueda tiene que ser menor a un medio del número de filas o columnas, dependiendo de cuál lado de la imagen sea menor).

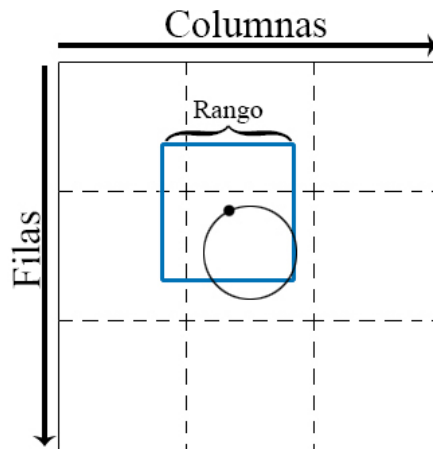


Fig. 2. Segmentación de la imagen

c) Triángulos rectángulos

Después, para cada par de puntos de borde con el mismo valor de ángulo se aplica el *Teorema de Tales*, el cual afirma que al trazar un triángulo rectángulo en el que la línea que une ambos puntos representa la hipotenusa del triángulo, si dicha línea cruza por el

centro de un círculo, también representa el diámetro del círculo (ver Fig. 3). Para calcular el radio del círculo se utiliza la siguiente ecuación:

$$R = \frac{\sqrt{(Y_2 - Y_1)^2 + (X_2 - X_1)^2}}{2} \quad (2)$$

Donde X_1, Y_1 representan la posición del primer punto y X_2, Y_2 representan la posición del segundo.

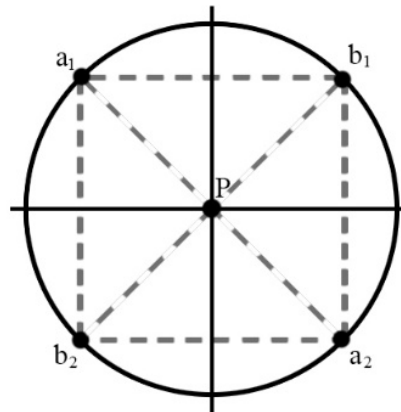


Fig. 3. Triángulos rectángulos en un círculo

Para acelerar el tiempo de procesamiento, se declaran dos constantes que representan el radio mínimo (R_{MIN}) y el radio máximo (R_{MAX}); si el radio calculado está entre el rango de radio mínimo y radio máximo, se obtiene el punto medio entre el par de puntos de borde, que podría ser el posible centro de un círculo.

Para saber si el punto medio calculado representa el posible centro de un círculo, se toman en cuenta las siguientes condiciones basadas en las características geométricas de un círculo:

$$|E_{RA}| \approx |E_{XA}| \quad (3)$$

$$|(X_0, Y_0 \mp r)| \approx 90^\circ \quad (4)$$

$$|(X_0 \mp r, Y_0)| \approx 0^o \quad (5)$$

Donde E_{RA} es el punto de borde ubicado en el ángulo recto (90 grados), del triángulo rectángulo; E_{XA} es el par de puntos de borde con el mismo valor; X_0, Y_0 representan el punto medio obtenido, y r es el radio calculado.

d) Almacenamiento y comparación de centros

Si se cumplen dos o más de las condiciones mencionadas, entonces el punto medio se considera el centro de un círculo y su posición se almacena en un arreglo acumulador bidimensional. Así mismo, también se guarda en una matriz de $n \times 3$, donde las primeras dos columnas representan el centro del círculo, y la tercera representa el radio.

En el último paso se normaliza el arreglo acumulador y los puntos que sean mayores a un valor de umbral elegido (T_{CP}), se comparan con los valores de las primeras dos columnas de la matriz de $n \times 3$; si ambos coinciden, se traza un círculo de radio igual al valor de la tercera columna de la matriz, a partir de las siguientes ecuaciones:

$$X_C = X_0 + r * \cos \theta \quad (6)$$

$$Y_C = Y_0 + r * \sin \theta \quad (7)$$

Donde X_C, Y_C son las coordenadas polares, y θ es el ángulo que va de 0 a 2π .

En resumen, la metodología que se propone se puede definir a partir del siguiente diagrama:

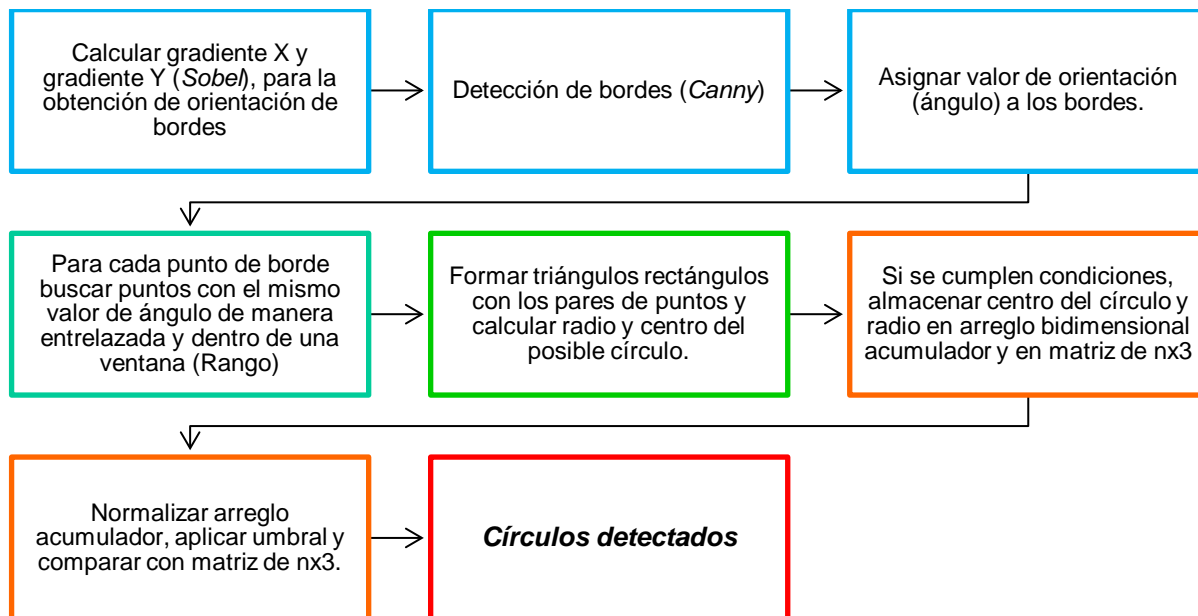


Fig. 4. Diagrama de metodología propuesta.

3. Resultados

Para probar el método propuesto se utilizaron alrededor de 25 imágenes de 256x256, en las que cada una contiene uno o más círculos de diferente radio. Todos los experimentos se realizaron en MATLAB 2010a con una computadora portátil con procesador AMD Athlon de 1.6 GHz.

El primer experimento consistió en utilizar una imagen sintética con fondo negro y cinco círculos blancos de distintos radios. Para probar el funcionamiento del método propuesto, se le añadió ruido sal y pimienta en distintos porcentajes a la imagen original (ver Fig. 5).

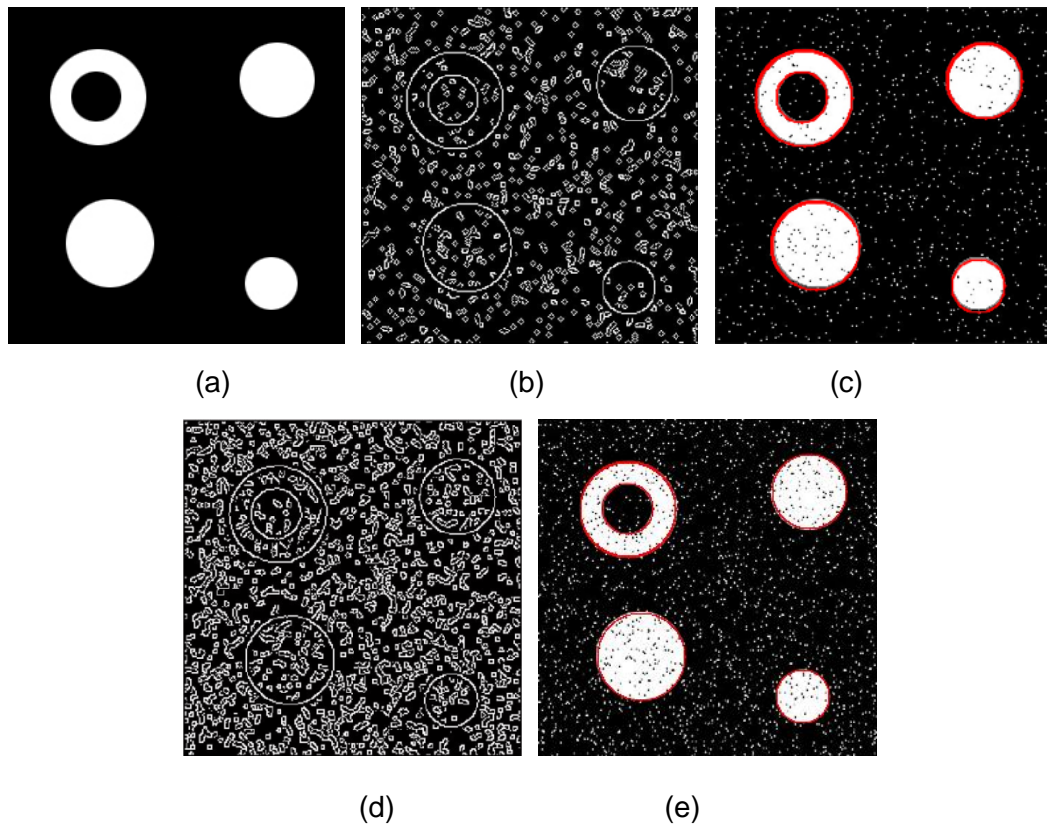


Fig. 5. (a) Imagen original; (b) bordes detectados de imagen con 3% de ruido sal y pimienta; (c) círculos detectados con 3% de ruido sal y pimienta; (d) bordes detectados de imagen con 7% de ruido sal y pimienta; (e) círculos detectados con 7% de ruido sal y pimienta.

Se hizo una comparación de los tiempos de ejecución con respecto al porcentaje de ruido con la CHT y el mismo método propuesto, pero sin la técnica de segmentación y análisis entrelazado. Para ambos métodos se utilizó un umbral de 50%, además cada imagen con un porcentaje de ruido se simuló 10 veces y se calculó el promedio del tiempo de ejecución (ver Fig. 6).

Como se puede observar en la figura 4, el ruido sal y pimienta afecta en gran cantidad a una imagen al momento de detectar sus bordes, esto debido a que agrega puntos de los dos extremos de la escala de grises (blanco y negro), y puede generar bordes falsos con facilidad. Aun después de agregar 7% de ruido a la imagen, con el método propuesto se

pueden detectar perfectamente los círculos, mientras que con la CHT la imagen se satura de círculos falsos detectados.

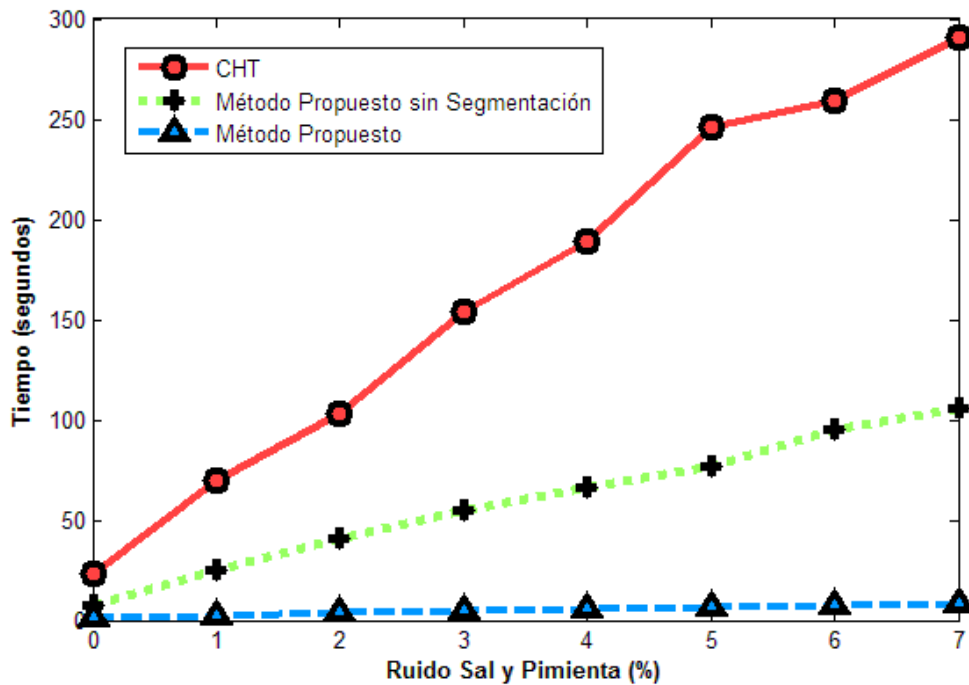


Fig. 6. Comparación de los tiempos de ejecución entre el método propuesto y la CHT.

Otro experimento que se realizó fue analizar el espacio en memoria que requiere el método propuesto para poder realizar la detección de círculos. Para esto, lo que se tomó en cuenta fue la matriz de $n \times 3$, la cual es dinámica; debido a que su tamaño es variable, se utilizaron 25 imágenes naturales con círculos de diferente radio y se obtuvo el promedio de la cantidad de bytes requeridos (ver Fig. 7).

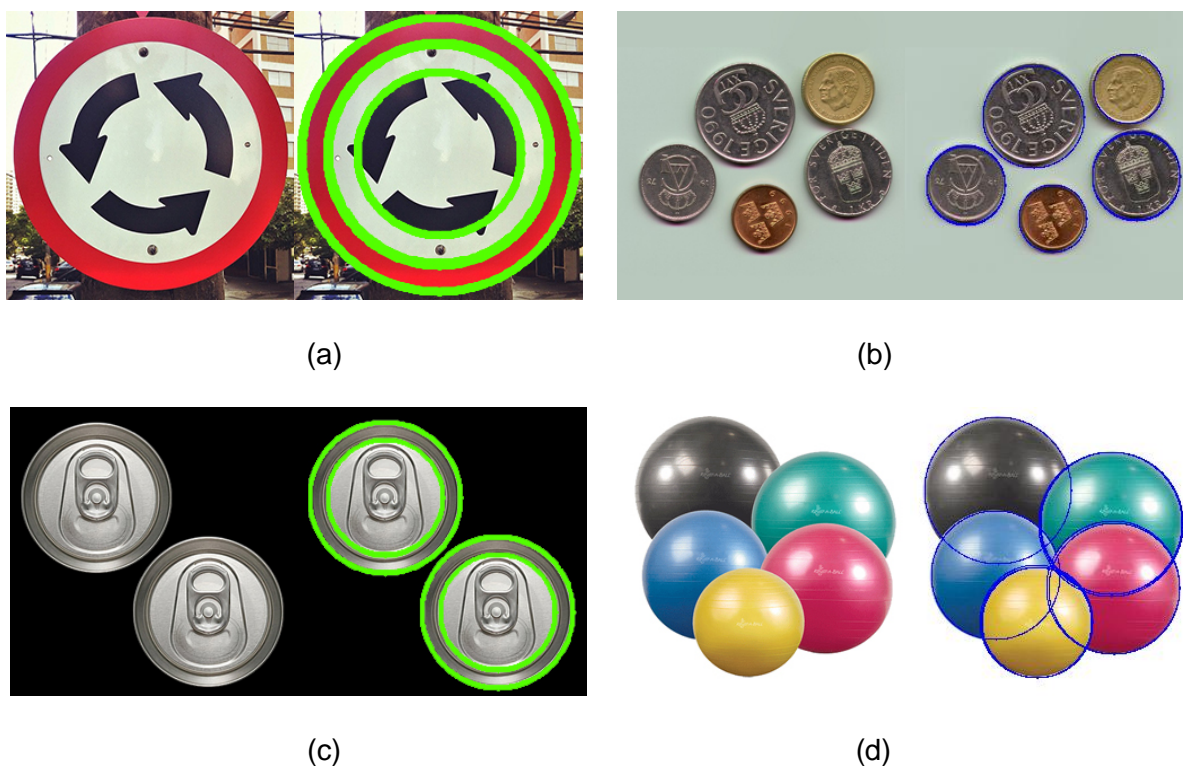


Fig. 7. (a) Círculos detectados de señalamiento; (b) Círculos detectados de monedas;(c) Círculos detectados de latas; (d) Círculos detectados de pelotas.

Después, al promedio obtenido se le sumó el tamaño en bytes del arreglo acumulador bidimensional, ya que éste es estático. Para la CHT se tomó en cuenta el arreglo tridimensional, que también es fijo. En ambos métodos se asumió desconocer el radio de los círculos en las imágenes, por lo que se consideró $r = 1, 2, \dots, 128$.

Algoritmo	Espacio en Memoria (kB)
CHT	67108.864
Método Propuesto	548.240

Tabla 1. Comparación de los algoritmos con respecto al espacio en memoria que requieren para detectar círculos.

Como se puede observar en la Tabla 1, el espacio que requiere el método propuesto para poder realizar la detección de círculos es poco mayor a 540kB, mientras que la CHT necesita 67MB aproximadamente, lo cual significa que el método propuesto reduce poco más de 120 veces el espacio requerido.

4. Discusión

En esta sección se discute el método propuesto haciendo una comparación con un método muy similar presentado en [9], y se comparan resultados obtenidos con imágenes naturales utilizadas en diversos artículos de detección de círculos. Además, se dan unas recomendaciones para mejorar la detección de círculos.

El método utilizado para comparar el propuesto también se basa en la información de los gradientes y las características geométricas de un círculo. La diferencia radica en que dicho método traza una línea para cada punto de borde, en dirección de su orientación. La desventaja que tiene este método es tener que calcular los puntos espaciales por los que cruza cada línea, lo cual incrementa el tiempo de procesamiento considerablemente. El método propuesto por el contrario, busca pares de puntos de borde con el mismo valor de ángulo dentro de una ventana y en columnas impares, lo cual ayuda a disminuir el tiempo de procesamiento y obtener los mismos o mejores resultados.

Entre las imágenes naturales que se utilizaron para poner a prueba el método propuesto y analizar los resultados, se escogieron imágenes manejadas en [9, 10, 11]. En todas las imágenes se logró detectar con éxito los círculos existentes. El método propuesto es capaz de no sólo detectar círculos completos, sino también círculos parciales (ver Fig. 8).

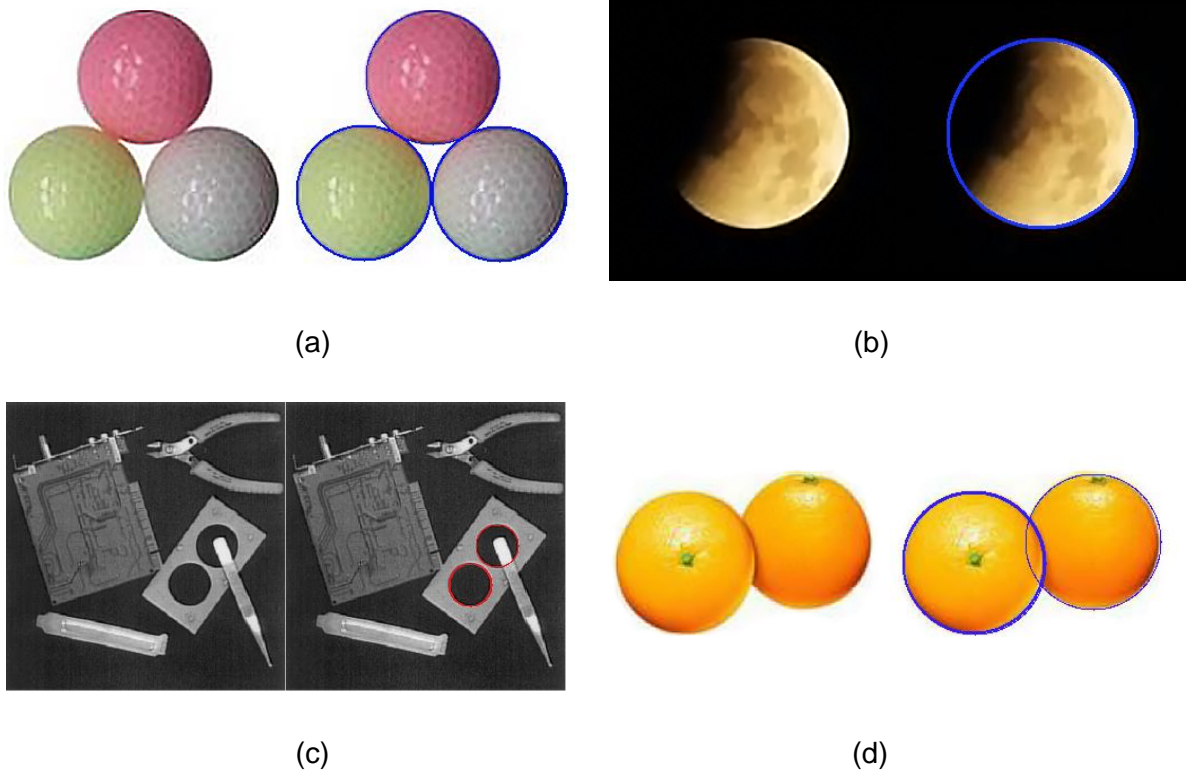


Fig. 8. (a) Círculos detectados de pelotas de golf; (b) Círculo detectado de luna parcial; (c) Círculos detectados de diferentes objetos; (d) Círculos detectados de par de naranjas.

Para poder tener una mejor detección de círculos, se recomienda reducir la distancia entre el radio mínimo y el máximo, y que el rango de búsqueda sea igual al radio máximo; de esta manera se disminuye la probabilidad de detectar círculos falsos y el tiempo de procesamiento.

Otros dos puntos que juegan un papel importante en el método propuesto son los umbrales de *Canny* (T_H y T_L) y el umbral del arreglo acumulador (T_{CP}); los primeros dos ayudan a controlar los bordes detectados en la imagen; si estos son muy bajos pueden provocar que se detecten bordes falsos, de lo contrario, si son muy altos, pueden eliminar bordes verdaderos, por lo que se debe de tener un buen balance en estos umbrales para obtener una buena detección. El umbral del arreglo acumulador ayuda a eliminar centros de círculos débiles, pero de igual manera que los umbrales de *Canny*, si se eleva mucho

el valor de umbral se pueden perder centros de círculos fuertes, o si se utiliza un valor muy bajo, puede conservar centros de círculos falsos. Un buen valor para este umbral debe ser mayor o igual a 30% y menor a 75%.

5. Conclusiones

En este trabajo se presentó un método eficaz basado en la orientación de los bordes y triángulos rectángulos dentro de un círculo para detectar círculos en imágenes. Los resultados experimentales muestran que el método propuesto es más rápido que la CHT y resiste más al incremento de puntos de borde en cuanto al tiempo de procesamiento. Además, también es más eficiente con el uso de recursos de memoria, ya que sólo utiliza un arreglo acumulador bidimensional y una matriz de $n \times 3$ que varía de tamaño según se cumplan las condiciones y características geométricas entre los pares de puntos de borde con el mismo valor de ángulo.

Las aplicaciones que puede tener este método pueden ser para la industria, o bien para la medicina, como por ejemplo el reconocimiento de señalamientos en carreteras para automóviles, el conteo de objetos como latas, monedas, células, etc. Se planea trabajar en un futuro con el método propuesto en arquitecturas reprogramables de bajo nivel para su uso en tiempo real en aplicaciones industriales.

6. Referencias

- [1] V. F. Leavers, *Shape Detection in Computer Vision Using the Hough Transform*. 1992. Londres. Springer London, pp. 138-172.
- [2] L. Xu, y E. Oja, "Randomized Hough Transform", vol. 3, 2009, pp. 1354–1361.

- [3] T.-C. Chen y K.-L. Chung, "An Efficient Randomized Algorithm for Detecting Circles", *Computer Vision and Image Understanding*, vol. 83, no. 2, Agosto 2001, pp. 172–191.
- [4] C. T. Ho y L. H. Chen, "A fast ellipse / circle detector using geometry symmetry", vol. 28, no. 1, 1995, pp. 117–124.
- [5] F. Shang, J. Liu, X. Zhang, y D. Tian, "An Improved Circle Detection Method Based on Right Triangles Inscribed in a Circle", 2009 WRI World Congress on Computer Science and Information Engineering, 2009, pp. 382–387.
- [6] A. A. Rad, K. Faez y N. Qaragozlou, "Fast Circle Detection Using Gradient Pair Vectors", *VIIth Digit. Image Comput. Tech. Appl.*, 2003, pp. 10–12.
- [7] S. Jansi and P. Subashini, "Optimized Adaptive Thresholding based Edge Detection Method for MRI Brain Images" *International Journal of Computer Applications*, vol. 51, no. 20, Agosto 2012, pp. 1–8.
- [8] J. Canny, "A Computational Approach to Edge Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, 1986, pp. 679–698.
- [9] L.-Q. Jia, H.-M. Liu, Z.-H. Wang, and H. Chen, "An effective non-HT circle detection for centers and radii," 2011 Int. Conf. Mach. Learn. Cybern. Julio 2011, pp. 814–818.
- [10] L. Jia, C. Peng, "A New Circle Detection Method Based on Parallel Operator", *Proceedings of the 2012 Int. Conference on Machine Learning and Cybernetics*, Julio 2012, pp. 1089.
- [11] H.-S. Kim, J.-H. Kim. "A two-step circle detection algorithm from the intersecting chords," *Pattern Recognition Lett.*, vol. 22, no. 6–7, Mayo 2001, pp. 787–798.

7. Autores

Alejandro Nava Acosta es estudiante de la carrera de Ingeniería en Sistemas Digitales y Comunicaciones en el Instituto de Ingeniería y Tecnología de la Universidad Autónoma de Ciudad Juárez (UACJ). Sus áreas de interés son DSP, FPGA e Instrumentación.

M. en C. Francisco Javier Enríquez Aguilera es profesor en la UACJ, obtuvo su maestría en Electrónica del Instituto Nacional de Astrofísica, Óptica y Electrónica. Sus áreas de interés son DSP, FPGA, Adquisición de Datos, Instrumentación y Control.

Dr. Juan Cota-Ruíz es profesor en la UACJ, obtuvo su Doctorado en la Universidad de Texas en El Paso (UTEP) en 2011 y su Maestría en el Instituto Tecnológico de Chihuahua en el 2003. El Dr. Cota ha publicado en revistas nacionales e internacionales en el área de Wireless Sensor Networks. Sus áreas de interés son Redes Inalámbricas de Sensores, Optimización Numérica, Instrumentación, Control y DSP.

Dr. Ernesto Sifuentes de la Hoya es profesor en la UACJ, obtuvo su Doctorado en la Universidad Politécnica de Catalunya (UPC) en 2009 y su Maestría en el Instituto Tecnológico de Chihuahua en 2002. Sus actuales intereses de investigación son Circuitos de Interfaz de Sensores, Instrumentación Virtual, Sistemas Embebidos, Sensores Autónomos, y Redes de Sensores Inalámbricas.

Dr. Pablo Rivas-Perea obtuvo su Doctorado en UTEP, El Paso, Texas en 2011 y su Maestría en el Instituto Tecnológico de Chihuahua. Ha trabajado en la industria por 8 años, recientemente hizo un verano de investigación en la NASA y actualmente se encuentra haciendo un Post-Doctorado en el Departamento de Ciencias Computacionales en la Universidad de Baylor, en Waco, Texas.