Detección de círculos usando Autómatas de aprendizaje

Erik Cuevas, Fernando Wario, Daniel Zaldivar y Marco Pérez-Cisneros

Departamento de Ciencias Computacionales
Universidad de Guadalajara, CUCEI
Av. Revolución 1500, Guadalajara, Jal, México
{erik.cuevas, fernando.wario, daniel.zaldivar, marco.perez}cucei.udg.mx

Resumen

La detección de círculos en imágenes digitales ha recibido una atención considerable en los últimos años en el área de visión por computadora. Por otra parte los Autómatas de Aprendizaje (LA por su nombre en inglés "Learning Automata") son un método heurístico que puede ser utilizado para encontrar la solución a complejos problemas de optimización. LA es una técnica para la adaptación de parámetros en optimización, donde la búsqueda se lleva a cabo en el espacio probabilístico en lugar del espacio de parámetros como sucede en otros algoritmos de optimización tradicionales. En este trabajo, se presenta un detector de círculos basado en LA en el cual el proceso de detección es abordado de manera similar a un proceso de optimización. El algoritmo utiliza la combinación de tres puntos que forman parte de un borde como los parámetros de círculos candidatos (x, y, r), mientras que una función objetivo determina si estos círculos candidatos se encuentran en realidad presentes en la imagen. De esta manera el esquema de parámetros reduce el espacio de búsqueda evitando probar con círculos que resultan poco probables o imposibles. El algoritmo resultante permite detectar círculos a alta velocidad con precisión a nivel sub-pixel, incluso en condiciones complicadas.

1. Introducción.

El problema de detección de círculos se presenta en diferentes áreas del procesamiento de imágenes y es de particular importancia en aplicaciones de tipo industrial como la inspección automática de productos manufacturados, vectorización asistida de dibujos, detección de objetivos, etc. Los problemas de detección de objetos son resueltos principalmente por dos tipos de técnicas: Las primeras son técnicas determinísticas, incluyendo aquellos métodos basados en la transformada de Hough y técnicas de ajuste de modelos. Por otro lado están las técnicas estocásticas, incluyendo técnicas de muestreo aleatorio y algoritmos genéticos (GA por su nombre en inglés: "Genetic Algorithms").

Tradicionalmente, la detección de círculos en imágenes digitales se lleva a cabo con el método de la transformada circular de Hough [1]. Un método típico basada en la transformada de Hough comienza obteniendo el mapa de bordes de la imagen, la información obtenida se utiliza para deducir la ubicación de los centros (x, y) y los valores de los radios (r), por último se lleva a cabo la detección de máximos. La desventaja de este método es la alta demanda de recursos computacionales. La exactitud en los parámetros obtenidos por esta clase de sistemas es pobre, particularmente en presencia de ruido y debido a la alta demanda de recursos computacionales esta opción es totalmente inadecuada para procesamientos en tiempo real. Con la intención de evitar estos problemas, algunos investigadores han propuesto nuevas variaciones a la transformada de Hough, entre ellas la "probabilistic Hough transform" [2]. Como una alternativa a las técnicas basadas en la transformada de Hough, el problema de detección de formas en el área visión por computadora ha sido también abordado con métodos de búsqueda estocásticos. En particular GA, los cuales han sido utilizados recientemente en importantes tareas de reconocimiento de formas e.g. Roth y Levine propusieron el uso de GA en la extracción de primitivas en imágenes [3]. Yao et al propusieron un GA multi-población para detectar elipses [4]. Ayala-Ramirez et al presentaron un detector de círculos basado en GA [5]. Su propuesta es capaz de detectar múltiples círculos en imágenes reales, pero falla frecuentemente en la detección de círculos imperfectos. El interés de utilizar los algoritmos de LA surge del hecho de que en ellos la búsqueda del valor óptimo se realiza en un espacio de probabilidades en lugar de un espacio de parámetros como sucede en otros algoritmos de

optimización. Cabe destacar que los algoritmos de LA han sido utilizados para resolver diferentes problemas de ingeniería [6]. Recientemente, se han propuesto algunos algoritmos basados en LA que resultan efectivos para la optimización de complejas funciones con múltiples mínimos (vea [7]). Además, se ha demostrado experimentalmente que el desempeño de estos algoritmos de optimización es comparable o mejor que el de los algoritmos genéticos en [7].

2. Autómatas de Aprendizaje

En la figura 1 se muestra la arquitectura típica de un sistema LA. El autómata selecciona de manera probabilística una acción (X). Dicha acción es aplicada al ambiente, y la función de evaluación del desempeño provee una señal de refuerzo β. Esta señal es utilizada para actualizar la distribución de probabilidades interna del autómata, de forma tal que aquellas acciones que obtienen un desempeño favorable son recompensadas incrementando su probabilidad, mientras que aquellas con un desempeño no favorable son penalizadas o su probabilidad permanece sin cambios, dependiendo del método de aprendizaje empleado. Con el tiempo, el desempeño promedio del sistema se verá mejorado hasta llegar a un límite. En términos de optimización, la acción que resulte con la mayor probabilidad representara el mínimo global.

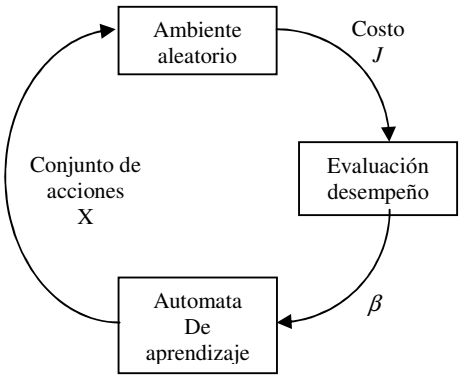


Fig 1. Sistema de aprendizaje por reforzamiento

Existen una gran cantidad de algoritmos para el aprendizaje en los LA, uno de los más utilizados es el de "linear reward/inaction (L_{RI}) scheme", del cual ha sido demostrada su convergencia [7]. En respuesta a la acción $\mathbf{x}_i$ seleccionada en el instante n, las probabilidades son actualizadas de la siguiente manera:

$$\begin{aligned} p_i(n+1) &= p_i(n) + \theta \cdot \beta(n) \cdot (1 - p_i(n)) \\ p_j(n+1) &= p_j(n) - \theta \cdot \beta(n) \cdot p_j(n), \text{ si } i \neq j \end{aligned} \tag{1}$$

Donde θ es un parámetro de aprendizaje y $0 < \theta < 1$ y $\beta \in [0,1]$ es la señal de refuerzo; si $\beta = 1$ significa que se obtuvo la mejor respuesta del ambiente, de manera contraria $\beta = 0$ es una respuesta nula. Eventualmente la probabilidad de acciones exitosas se incrementará hasta aproximarse a la unidad. En caso de que una sola acción resulte exitosa durante mucho tiempo, se considera que el autómata ha convergido.

3. Detección de círculos con LA

La primera tarea del algoritmo consiste en obtener el mapa de bordes por medio del algoritmo de Canny, los puntos obtenidos son los únicos candidatos potenciales para la definición de círculos. Con el objetivo de mejorar la velocidad del algoritmo se eligen de manera aleatoria un porcentaje representativo de los puntos-borde (alrededor del 5%) de la imagen, las

coordenadas de estos puntos son archivados en un vector $P=\{p_1,p_2,\ldots,p_{N_p}\}$ con N_p como el número de puntos-borde tomados. Con el fin de construir cada círculo candidato (acción en el caso del enfoque para LA), se generan todas las combinatorias posibles de 3 elementos de los puntos-borde, así se asume que el contorno del círculo (acción) pasa por los puntos elegidos p_i, p_j y p_k, las acciones generadas hasta el momento son codificadas en las coordenadas del centro x_0, y_0 y radio *r* del círculo que pasa por los tres puntos, estos parámetros pueden ser calculados con las siguientes ecuaciones:

$$x_0=\frac{\det(\mathbf{A})}{4((x_j-x_i)(y_k-y_i)-(x_k-x_i)(y_j-y_i))},\ y_0=\frac{\det(\mathbf{B})}{4((x_j-x_i)(y_k-y_i)-(x_k-x_i)(y_j-y_i))}, \quad (2)$$

y

$$r=\sqrt{(x_0-x_d)^2+(y_0-y_d)^2}, \quad (3)$$

donde

$$\mathbf{A}=\begin{bmatrix} x_j^2+y_j^2-(x_i^2+y_i^2) & 2\cdot(y_j-y_i) \\ x_k^2+y_k^2-(x_i^2+y_i^2) & 2\cdot(y_k-y_i) \end{bmatrix}\ \mathbf{B}=\begin{bmatrix} 2\cdot(x_j-x_i) & x_j^2+y_j^2-(x_i^2+y_i^2) \\ 2\cdot(x_k-x_i) & x_k^2+y_k^2-(x_i^2+y_i^2) \end{bmatrix}, \quad (4)$$

det() es la operación determinante y $d\in\{i,j,k\}$. La figura 2 muestra los parámetros definidos por las ecuaciones (2) y (3).

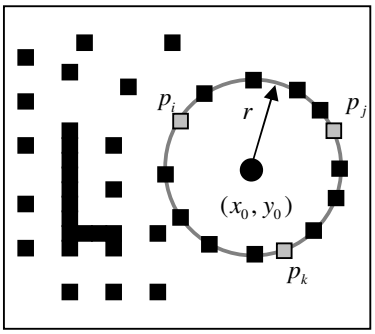


Fig. 2. Circulo candidato (acción) formado por la combinación de los puntos p_i, p_j y p_k.

El conjunto de acciones resultante se filtra para eliminar aquellos círculos con un radio fuera de los limites de interés (eliminando círculos pequeños que generalmente son ruido), además, el algoritmo comprueba que todos los círculos considerados como acciones sean distintos, de esta manera se reduce la cantidad de acciones y por consiguiente el tiempo de ejecución del algoritmo, pues se evita que dos acciones con los mismos parámetros compitan entre ellas. La solución del LA al problema de detección es encontrada mediante la evolución de las probabilidades asociadas a cada circulo (acción), las cuales son modificadas de acuerdo a la señal de refuerzo $\beta(C)$ obtenida. Para calcular $\beta(C)$ se generan las coordenadas de los puntos de la circunferencia correspondiente a la acción elegida. Las coordenadas son generadas por el "midpoint circle algorithm (MCA)" [8] y se colocan en el vector $S=\{s_1,s_2,\ldots,s_{N_s}\}$, con N_s como el número de puntos que forman la circunferencia. Después se revisa si los puntos del vector S coinciden con puntos en el mapa de bordes y se evalúa la siguiente formula:

$$\beta(C)=\frac{\sum_{i=1}^{N_s}E(x_i,y_i)}{N_s} \quad (5)$$

Donde $E(x_i, y_i)$ representa la cantidad de puntos de S que coinciden con puntos en el mapa de bordes; N_s es el número de pixeles en el perímetro del círculo que corresponde a la acción *C* que se está evaluando. De ahí que el algoritmo busca maximizar a $\beta(C)$, dado que un valor mayor representa una mejor respuesta de "circularidad" por parte del operador. Se ejecuta el algoritmo del LA con un limite de ciclos preestablecido, si durante la ejecución de estos ciclos una de las acciones elegidas genera un error por debajo de un limite previamente establecido se toma esta acción como la solución a este problema, de manera similar, si alguna de las acciones presenta un valor de probabilidad suficientemente alto como para no ser revertido en los ciclos restantes, el entrenamiento se interrumpe y se toma dicha acción como la optima; en caso contrario se ejecutan todos los ciclos y se toma como solución a nuestro problema de detección de circulo aquella acción con la probabilidad más alta.

4. Resultados experimentales

Se realizaron una serie de experimentos con el fin de evaluar el desempeño de nuestro detector de círculos. A continuación se presentan algunos de los resultados obtenidos con el sistema. En la figura 3 se muestra el circulo detectado en la imagen en color rojo, además de la distribución de probabilidades de las acciones a lo largo de las iteraciones, se puede apreciar la convergencia del algoritmo hacia una sola acción.

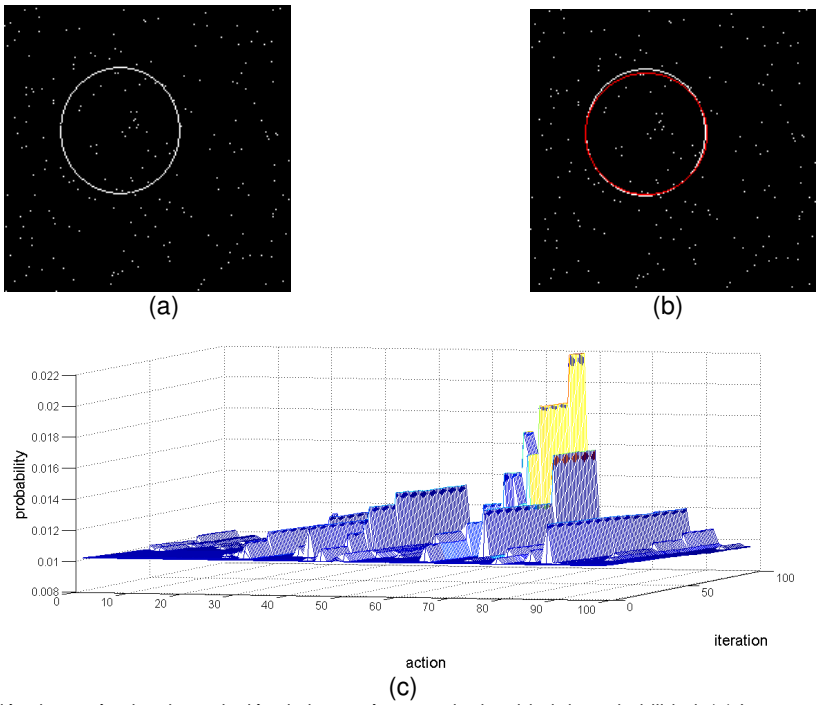
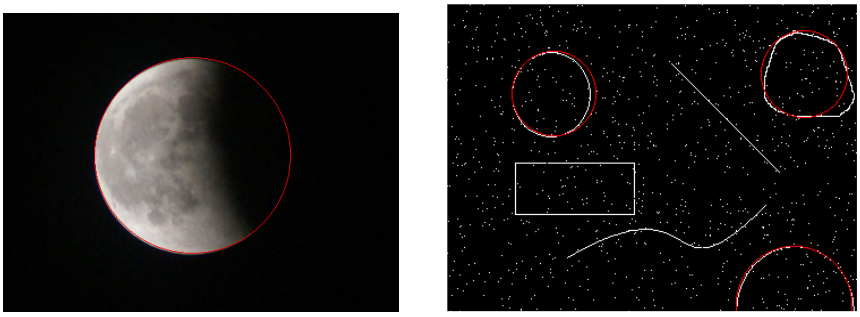


Fig. 3. Detección de un círculo y la evolución de los parámetros de densidad de probabilidad. (a) Imagen original, (b) el círculo detectado se muestra en rojo, (c) Evolución de las densidades de probabilidad.

Una de las ventajas de considerar el problema de detección con un enfoque de optimización, es la posibilidad de poder detectar arcos de círculos, círculos ocluidos o bien círculos que presenten imperfecciones de forma. Esto se debe a que el algoritmo LA es capaz de encontrar los parámetros del círculo que de acuerdo al valor de probabilidad $p(A_i)$ mejor aproximen al arco, al circulo ocluido o bien al circulo imperfecto. La figura 4a muestra uno de estos ejemplos, en el cual la figura circular, en este caso la luna muestra parte de su circunferencia ocluida. Otro de los resultados obtenidos se presenta en la imagen 4b, la imagen consta de 6 objetos, entre ellos 3 círculos, el primero de ellos un círculo casi perfecto, el segundo un círculo que por encontrarse en una de los extremos de la imagen aparece ocluido y por último una figura circular dibujada a mano.



(a) (b)

Fig. 4. Detección de círculos en condiciones complejas. (a) Detección de círculos ocluidos y (b) círculos dibujados a mano.

5. Conclusiones

En este artículo hemos presentado un nuevo método para detección de círculos basado en LA. El método es capaz de detectar círculos con precisión a nivel sub-pixel tanto en imágenes reales como artificiales. El detector de círculos es capaz de detectar de manera confiable figuras circulares inclusive si estas se encuentran significativamente ocluidas, también figuras circulares dibujadas a mano y arcos. La principal debilidad de nuestro sistema surge con la búsqueda de círculos pequeños. Este problema se debe a la capacidad de nuestro sistema de trabajar con círculos imperfectos, es decir, cuando un pequeño sector circular pueda ser inscrito, será detectado como un imperfecto pero probable círculo. El comportamiento de nuestro sistema con respecto a pequeños círculos puede ser ajustado al contexto de trabajo.

Referencias

[1] Muammar, H., Nixon, M., 1989. Approaches to extending the Hough transform. In: Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP_89, vol. 3, pp. 1556–1559.

[2] Shaked, D., Yaron, O., Kiryati, N., 1996. Deriving stopping rules for the probabilistic Hough transform by sequential analysis. Comput. Vision Image Understanding 63, 512–526.

[3] Roth, G. and Levine, M. D.: Geometric primitive extraction using a genetic algorithm. *IEEE Trans.Pattern Anal. Machine Intell*. 16 (9), 901–905, 1994.

[4] Yao, J., Kharma, N., and Grogono, P.: Fast robust GA-based ellipse detection. *In: Proc. 17th Int. Conf. on Pattern Recognition ICPR-04*, vol. 2, Cambridge, UK, pp. 859–862, 2004.

[5] Ayala-Ramirez, V., Garcia-Capulin, C. H., Perez-Garcia, A., and Sanchez-Yanez, R. E.: Circle detection on images using genetic algorithms, *Pattern Recognition Letters*, 27, 652–657, 2006.

[6] Seyed-Hamid Z., 2008. Learning automata based classifier, Pattern Recognition Letters.

[7] Najim, K., Poznyak, A.S., 1994. Learning Automata - Theory and Applications. Pergamon Press, Oxford.

[8] J.E. Bresenham: A Linear Algorithm for Incremental Digital Display of Circular Arcs. Communications of the ACM 20, 100-106. (1987).