

A cooperative method for image segmentation based on cellular automata

Jorge Parra, Niriaska Perozo*

Unidad de Investigación en Inteligencia Artificial, Decanato de Ciencias y Tecnología, Universidad Centroccidental Lisandro Alvarado, Barquisimeto 3001-Venezuela.

Abstract.-

A very important stage in the artificial vision process consists in segmenting images for extracting data of interest and reducing their complexity and processing time. A simple way to segment an image is by edge detection. In this paper, a cooperative method based on cellular automata is proposed for detecting edges in images, where cells compare neighboring RGB components to determine the existence of edges. In this sense, an application was implemented using the object-oriented paradigm and tests were performed to compare the segmentation's quality achieved by the proposed method, yielding better results in more scenarios than other existing edges detection methods.

Keywords: Computer vision; image segmentation; edge detection; cellular automata.

Un método cooperativo para la segmentación de imágenes basado en autómatas celulares

Resumen.-

Una fase muy importante para el proceso de visión artificial es la segmentación de las imágenes, debido a que en esta fase se puede reducir la cantidad de datos de la imagen y obtener sólo los datos de mayor interés acerca de los objetos que la componen, lo que permite disminuir la complejidad de la imagen y el tiempo de cómputo para su procesamiento. Una manera sencilla de segmentar una imagen es mediante el proceso de detección de bordes. En el presente trabajo, se propone un método cooperativo para detectar bordes en imágenes basándose en autómatas celulares, en el que las células comparan los componentes RGB de la vecindad cooperando entre ellas para determinar si existen bordes. En este sentido, se implementó una aplicación utilizando el paradigma orientado a objetos, con la que se realizaron pruebas para comparar la calidad de segmentación lograda por el método propuesto, mostrando en la mayoría de los escenarios mejores resultados que los obtenidos con otros métodos de detección de bordes.

Palabras clave: visión artificial; segmentación de imágenes; detección de bordes; autómata celular.

Recibido: noviembre 2015

Aceptado: febrero 2017

1. Introducción

La detección de bordes es el primer paso en la segmentación de imágenes: un método que

se utiliza para determinar la composición de una imagen. Los bordes de una imagen suelen contener mucha información sobre la imagen y describen dónde se encuentran los objetos, sus formas y tamaños, e información sobre su textura. El estudio de los bordes se ha convertido en una tarea muy importante en el procesamiento de imágenes médicas o biológicas, ya que se puede reducir significativamente la cantidad de

*Autor para correspondencia

Correo-e: nperozo@ucla.edu.ve (Niriaska Perozo)

datos de una imagen, filtrar la información inútil y preservar las características importantes de la misma, lo que permite acelerar su procesamiento. En base a esto, en el presente trabajo se propone la implementación de un método para la segmentación de imágenes basado en autómatas celulares, que se puede utilizar para mejorar el proceso de visión artificial. Específicamente, este método permite la detección de bordes en las imágenes, ya que las células del autómata pueden determinar cooperativamente junto con las células vecinas cambios en las tonalidades de color, comparando los componentes RGB de la imagen con respecto a un umbral de reconocimiento establecido. Este umbral puede ser ajustado para obtener una mejor calidad de detección de los bordes en las imágenes analizadas, según se verificó en los diferentes escenarios de prueba planteados.

2. Trabajos Relacionados

A continuación, se presentan algunos trabajos donde se implementan sistemas de visión artificial utilizando autómatas celulares.

Pritan et al [1] proponen el diseño y desarrollo de una aplicación para el procesamiento de imágenes basado en una unidad de procesamiento de gráficos (GPU, *Graphics Processing Unit* en inglés) en lugar de utilizar una unidad central de procesamiento (CPU, *Central Processing Unit* en inglés), para alcanzar un gran incremento en la velocidad del procesamiento de imágenes. Adicionalmente para aprovechar la memoria gráfica del computador utilizan una arquitectura unificada de dispositivos de cómputo (CUDA, *Compute Unified Device Architecture* en inglés), el lenguaje C++, principios de ingeniería de software, patrones de diseño y Qt Framework. De acuerdo con este planteamiento podemos ver que un problema frecuente en el procesamiento de las imágenes es la velocidad, y que los autores intentan dar una solución a nivel de hardware. El método propuesto en esta investigación trata de dar una solución a nivel de software basado en los autómatas celulares, ya que estos poseen una gran rapidez al momento de realizar operaciones de cómputo según [2].

En [3], Cifuentes y colaboradores proponen una técnica para la segmentación de imágenes digitales por textura operando sobre una rejilla de píxeles, basándose en el método de Lattice-Boltzmann, un autómata celular que representa el estado de cada celda (pixel) con 9 escalares que representan poblaciones de partículas transportándose entre celdas vecinas. La población de partículas por celda viene definida por una función de densidad que define la probabilidad de que las partículas se transporten a una dirección específica, representando una regla de evolución del estado de la celda. A través de la experimentación con texturas en imágenes sintéticas y naturales, lograron alcanzar una calidad de segmentación entre 85 % y 95 % .

De forma similar, en [4] se presenta un método para segmentar imágenes satelitales generadas por el sensor de un radar de apertura sintética (SAR, *Synthetic Aperture Radar* en inglés), para ello proponen un algoritmo, el cual combina la información sobre la intensidad y la entrada de la textura de la imagen a través del aprendizaje de los autómatas celulares. Para la detección de bordes, las celdas del autómata deben calcular la distancia euclídeana entre los componentes RGB de cada uno de los píxeles de color y de todos sus vecinos. El estudio demostró que con la realización de dos segmentaciones, una simple y otra compleja, se puede utilizar el autómata celular para adaptar su comportamiento al entorno, modelando un comportamiento asociado a la textura. En el método propuesto en esta investigación, a diferencia del método antes descrito, se determina la distancia de cada uno de los componentes RGB por separado, y cada uno es comparado con un umbral diferente.

Por otro lado, en [5] y [6] los autores realizan pruebas con diferentes métodos demostrando que aquellos basados en autómatas celulares, se comportan mejor en casi todos los escenarios que otros métodos para la segmentación y detección de bordes en imágenes, entre los cuales se encuentran Priors Label, GrowCut, Watershed, el método de Otsu, el detector de bordes Prewitt, y los operadores de Sobel, Roberts y Canny. Similarmente, en el presente trabajo se realizó una comparación de los resultados obtenidos con el método planteado y los obtenidos por [5] para evaluar la efectividad

del método propuesto.

Finalmente, en [7] se presenta un método basado en autómatas celulares difusos, que aplica 32 reglas generadas por un motor de inferencia, a cada una de las células, estudiando su vecindad de 8 celdas, para determinar la pertenencia a alguno de los 3 tipos de píxeles estudiados (ruido, borde o fondo). Este método es utilizado para eliminar el ruido en las imágenes y para la detección de bordes, demostrando eficiencia en la clasificación de los píxeles de la imagen a través de las reglas difusas definidas para tal fin. A diferencia de dicho método, las células del autómata propuesto en este trabajo evalúan una vecindad de 4 celdas, y las reglas para detección de bordes que lo rigen no son generadas por un motor de inferencia, sino que dependen de los cambios en los componentes RGB de las celdas comparadas para dar un mejor tiempo de respuesta.

3. Aspectos Teóricos Relevantes

3.1. Autómatas Celulares

Un autómata celular (AC) se define según [3] como un modelo matemático para un sistema dinámico, compuesto por un conjunto de células que adquieren distintos estados o valores que son alterados de un instante a otro en intervalos regulares de tiempo.

Un AC está caracterizado por los siguientes elementos básicos:

- Arreglo regular: constituye el espacio de evolución en un plano 2D o en un espacio n -dimensional, donde cada división homogénea de arreglo es llamada célula.
 - Conjunto de estados: un conjunto finito de valores o colores, también denominado alfabeto, del que las células del arreglo toman sus valores.
 - Configuración inicial: son los valores de estado asignados inicialmente a las células del espacio inicial de evolución del sistema.
 - Vecindades: define el conjunto contiguo de células y sus posiciones relativas entre sí;
- Función local: regla de evolución que determina el comportamiento del AC. Está conformada por una célula central y sus vecindades. Define el cambio de estado de cada célula dependiendo de los estados anteriores de sus vecindades, y puede ser una expresión algebraica o un grupo de ecuaciones.

a cada vecindad diferente corresponde un elemento del conjunto de estados.

3.2. Sistema de Visión Artificial

La visión por computador comprende la obtención, caracterización e interpretación de las imágenes. En concordancia, un sistema de visión artificial está compuesto por un sensor óptico para capturar la imagen, y un computador para almacenar las imágenes y ejecutar los algoritmos de pre-procesamiento, segmentación y reconocimiento [8].

Fases de un Sistema de Visión. De acuerdo a Ortiz [8], tenemos que las fases más importantes de un sistema de visión son las siguientes:

- Captación: obtención de una imagen mediante una cámara o dispositivo de captura.
- Pre-procesamiento: aplicación de técnicas que facilitan el procesamiento posterior; incluye la reducción de ruido y realce de detalles.
- Segmentación: división de una imagen en regiones de interés.
- Descripción: caracterización y diferenciación de distintos tipos de objetos.
- Reconocimiento: identificación de objetos pertenecientes a una escena.
- Interpretación: asignación de significado a un conjunto de objetos reconocidos en una imagen.

4. Descripción del Método Propuesto

En el presente trabajo se propone inicializar cada célula del autómata con un pixel de la imagen, de tal manera que un pixel en la posición (x, y) es asignado a la célula en la posición (x, y) del autómata. Posteriormente, cada una de las células evalúa de forma simultánea su vecindad para determinar si existen bordes, de acuerdo con el algoritmo mostrado en la Figura 1.

```

si cambioColor vArrCen o vCenIzq entonces
|  marcar borde;
en otro caso
|  si cambioColor vArrIzq o vArrDer entonces
|  |  marcar borde diagonal;
|  en otro caso
|  |  no hay borde;
|  fin
fin

```

Figura 1: Algoritmo para determinar si existen bordes cada una de las células.

Cada una de las células del autómata recibe un píxel de la imagen seleccionada y evalúa, a través de los umbrales establecidos, si existe o no un borde dentro de la imagen al comparar con los píxeles de las células vecinas, asignando un color según la pertinencia y tipo de borde: si no existe un píxel asignado a la célula el color mostrado es el azul, para bordes horizontales o verticales la célula es de color negro, y gris para bordes diagonales; en caso de que no se detecte ningún borde, la célula será de color blanco (ver Figura 2).

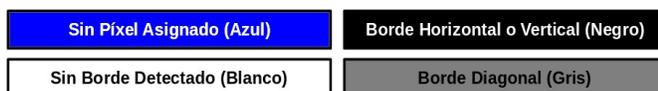


Figura 2: Colores mostrados por la célula de acuerdo a la evaluación de los píxeles.

Para evaluar la existencia de bordes en una célula, se toma en cuenta los valores RGB que tiene el píxel asignado a la célula, y se compara de acuerdo a los valores establecidos para cada umbral. Los umbrales indican la diferencia mínima que debe existir en los componentes RGB de dos células para determinar si existe o no un borde. Existen 3 umbrales diferentes, uno para cada componente RGB: el umbral R representa la

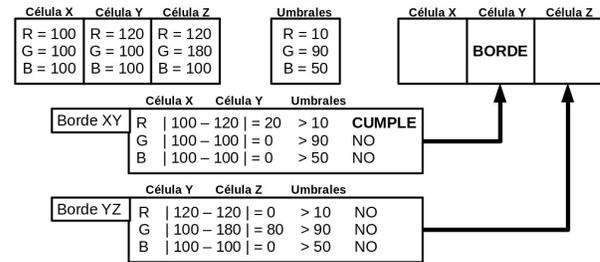


Figura 3: Detección de borde a partir de la comparación de los componentes de los píxeles evaluados.

diferencia para el componente R (red - rojo), el umbral G para el componente G (green - verde) y el umbral B para el componente B (blue - azul). Para determinar si existe o no un borde, la célula compara cada uno de los componentes RGB de su píxel con el de su célula vecina y si la diferencia entre los valores de alguno de los componentes supera el umbral para dicho componente, se determina la existencia de un borde (ver Figura 3). Un borde existe si para al menos uno de los componentes RGB entre una célula X y una célula Y se cumple la ecuación(1)

$$|A - B| \geq U \quad (1)$$

donde:

$$\begin{aligned}
 A &= \text{Célula } X_{(R,G,B)}, \\
 B &= \text{Célula } Y_{(R,G,B)} \text{ y} \\
 U &= \text{Umbral}_{(R,G,B)}
 \end{aligned}$$

La comparación se realiza tomando en cuenta para la evaluación solamente 4 células de su vecindad (ver Figura 4.a): la célula superior izquierda, la superior central, la superior derecha y la célula izquierda. Esta vecindad fue elegida debido a que al ser aplicada sobre todas las células, se obtienen las comparaciones de forma cooperativa desde las 8 direcciones posibles (ver Figura 4.b), de tal manera que no se realiza la comparación más de una vez como sucede en los métodos clásicos (ver Figura 4.c y 4.d). El algoritmo utilizado para verificar si existe un cambio de color entre una Célula X y una Célula Y es mostrado en la Figura 5.

5. Interfaz de la Aplicación

Para el desarrollo del método propuesto se implementó una aplicación basada en autómatas

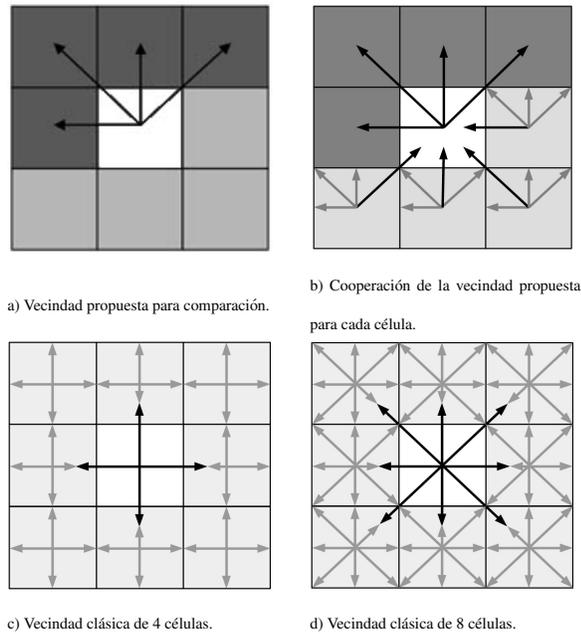


Figura 4: Vecindades.

```

Entrada: CelulaX, CelulaY
Salida: Hay cambio de color
difR = abs(CelulaX.cR - CelulaY.cR);
si difR ≥ umbralR entonces
    | devolver Verdadero
en otro caso
    | difG = abs(CelulaX.cG - CelulaY.cG);
    | si difG ≥ umbralG entonces
    | | devolver Verdadero
    | en otro caso
    | | difB = abs(CelulaX.cB - CelulaY.cB);
    | | si difB ≥ umbralB entonces
    | | | devolver Verdadero
    | | fin
    | fin
fin
devolver Falso
    
```

Figura 5: Algoritmo para verificar si existe un cambio de color entre una Célula X y una Célula Y.

celulares, para segmentar imágenes. Específicamente, cada célula del autómata debe ser capaz de leer un pixel de la imagen, evaluar su vecindad y de acuerdo con los umbrales establecidos, definir las diferentes regiones dentro de la imagen y los bordes detectados en las mismas.

La aplicación se programó con el lenguaje Java, siguiendo el paradigma orientado a objetos y definiendo una interfaz gráfica a través de la librería Swing para la selección de la imagen, la personalización del proceso de segmentación y

renderizado de la imagen.

El primer paso para el proceso de segmentación consiste en elegir la imagen que se desea segmentar, utilizando las opciones definidas en la interfaz de usuario (Figura 6). Los umbrales definidos para la evaluación de cada uno de los componentes RGB de la imagen, pueden ser modificados en cualquier momento por cada color o de manera conjunta (umbral general). Los botones de movimiento permiten cambiar la posición de la imagen con respecto al área de dibujo (canvas), es decir asigna otros píxeles a las células del autómata para observar otra región de la imagen, lo cual es útil cuando la imagen mostrada tiene grandes dimensiones.

6. Resultados Obtenidos

Para evaluar la capacidad de detección del método propuesto, se experimentó con imágenes de diferente nivel de detalle para calibrar los valores de los umbrales, analizar los tiempos de cómputo del algoritmo y determinar visualmente la calidad de la segmentación obtenida. De igual manera, se realizó una comparación de los resultados obtenidos con respecto a los proporcionados por otros métodos existentes.

6.1. Calibración inicial de umbrales

Se utilizó una imagen compuesta por círculos en los que varía sólo uno de sus componentes RGB con respecto a sus vecinos, para calibrar el valor de los umbrales (ver Figura 7).

En el primer caso, la aplicación detectó sin problema los bordes utilizando la imagen original para cada componente RGB con un umbral igual a 40. Sin embargo, al variar el umbral de alguno de los componentes, la aplicación obvia la detección de bordes entre colores con componentes comunes. Como ejemplo, para el segundo caso (umbral R=255) se evita la detección de bordes entre el rojo (255, , 0) y el negro (0, 0, 0), y entre el cyan (0, 255, 255) y el blanco (255, 255, 255); en el tercer caso (umbral B = 255), no se detecta bordes entre el azul (0, 0, 255) y el negro (0, 0, 0), y entre el amarillo (255, 255, 0) y el blanco (255, 255, 255).

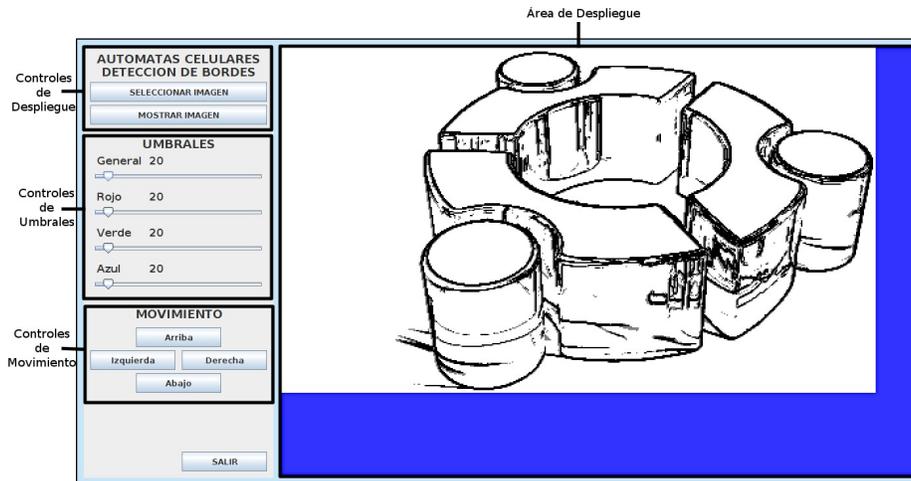


Figura 6: Vista de la Interfaz de Usuario de la Aplicación.

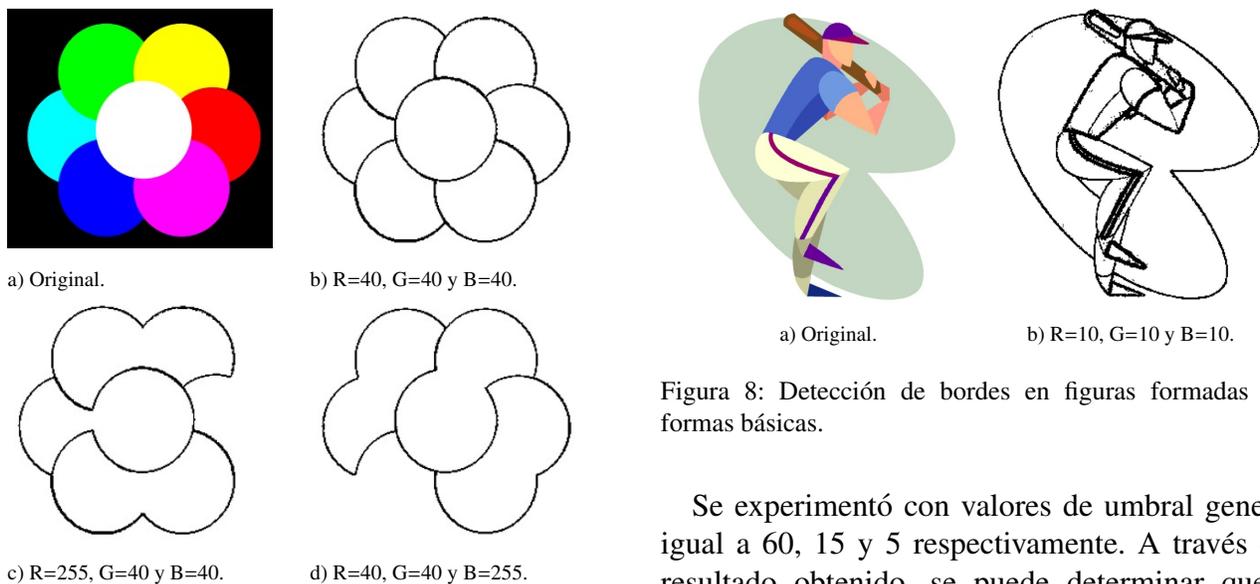


Figura 8: Detección de bordes en figuras formadas por formas básicas.

Figura 7: Calibración de los valores de umbrales por componentes.

6.2. Segmentación de formas sencillas

Se utilizó una imagen constituida por formas sencillas (ver Figura 8); los resultados demuestran que el autómata celular puede segmentar los bordes de figuras simples con gran precisión.

6.3. Disminución progresiva de umbrales (calibración 2)

Para evaluar la segmentación, se ejecutó una prueba en una imagen con sombras y gradiente de colores (ver Figura 9), disminuyendo progresivamente el valor de los umbrales.

Se experimentó con valores de umbral general igual a 60, 15 y 5 respectivamente. A través del resultado obtenido, se puede determinar que a medida que el umbral es mayor se pueden detectar bordes sólo cuando existe gran diferencia entre los componentes de los colores de la imagen, mientras que cuando el umbral es menor se detectan cambios pequeños entre los colores.

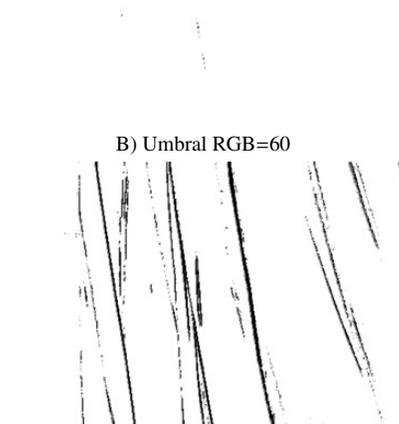
6.4. Segmentación de formas complejas

En esta sección se evaluó el reconocimiento de bordes en imágenes complejas con gran nivel de detalle (ver Figura 10). Se experimentó colocando los valores del umbral general en 2, 10 y 50 respectivamente.

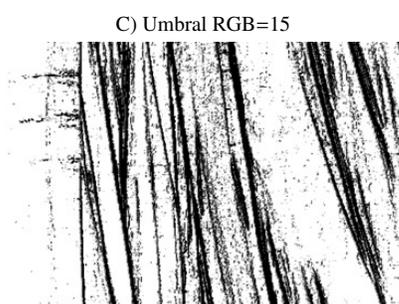
En los siguientes escenarios, se realizan comparaciones entre el método planteado en la presente investigación y los resultados obtenidos en la investigación realizada por [5], donde a su vez



A) Original



B) Umbral RGB=60



C) Umbral RGB=15



D) Umbral RGB=5

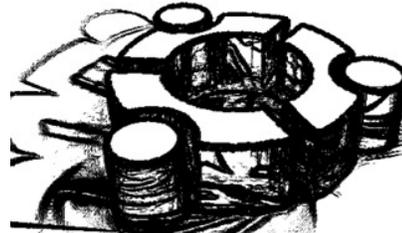
Figura 9: Detección de bordes en imagen con gradientes y sombras con umbral general igual a 60, 15 y 5.

se compara un método de detección de bordes basado en autómatas celulares con los métodos de Roberts, Prewitt, Sobel y Canny. Se adicionó a sus resultados aquellos obtenidos en este trabajo, utilizando el método planteado con autómatas celulares pero comparando los componentes RGB entre las células vecinas no más de una vez, como sucede en los métodos clásicos.

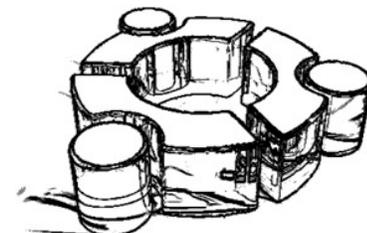
Para esta tarea, se realizaron algunos cambios al método planteado en cuanto a la presentación de la



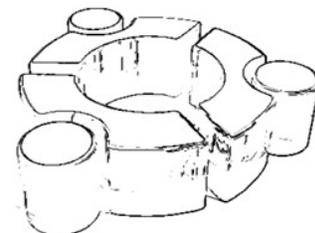
A) Original



B) Umbral RGB=2



C) Umbral RGB=10



D) Umbral RGB=50

Figura 10: Detección de bordes en imágenes con alto nivel de detalle con umbral general igual a 2, 10 y 50.

imagen segmentada para poder realizar una mejor comparación de los resultados entre los trabajos. Anteriormente se utilizaba el color negro para representar los bordes horizontal y verticalmente, gris para bordes diagonales y blanco para indicar que no se detecto ningún borde, pero en los siguientes experimentos se utilizará el color negro para indicar ausencia de bordes, gris para bordes diagonales y blanco para bordes horizontales y verticales. Sin embargo, el método para la detección de los bordes sigue el mismo procedimiento ya planteado en la sección 4 de esta investigación.

6.5. Imagen del fotógrafo, comparación de resultados

En este experimento se tomó para segmentar la imagen del fotógrafo (ver Figura 11) y comparar los resultados obtenidos con el método propuesto en este trabajo y otros métodos existentes.

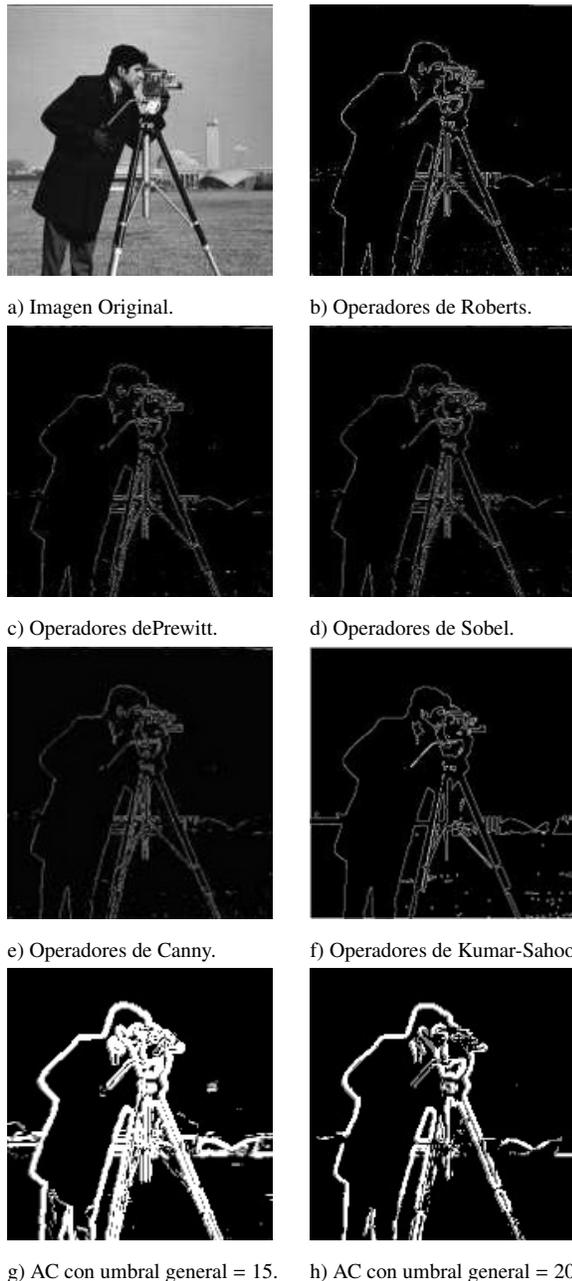


Figura 11: Resultados de detección de bordes fotógrafo.

En la segmentación obtenida se puede apreciar la figura principal a través de los bordes detectados. Además, el hecho de que los componentes

RGB cercanos a los bordes presentan una diferencia mayor al umbral permite generar bordes con un grosor mayor al de los otros métodos, lo que contribuye a una mejor segmentación de los objetos que componen la imagen; sin embargo, este grosor evita que se puedan detectar bordes en pequeños detalles de la imagen. Igualmente como en los otros trabajos, la diferencia entre los componentes de los píxeles del fondo de la imagen es muy pequeña y por tanto no se diferencian los objetos ubicados al fondo.

6.6. Imagen de Lena, comparación de resultados

Finalmente, se segmentó la imagen Lena (ver Figura 12) para comparar el comportamiento de los métodos de detección de bordes de forma similar a la sección 6.5.

A través de este escenario, se puede observar que el método propuesto define mayor número de zonas en la imagen analizada, permitiendo reconocer otros objetos que no se pueden diferenciar en los resultados obtenidos en los otros trabajos. También se puede ver que existe mayor continuidad en los bordes detectados y una mejor definición de las zonas debido al grosor de los bordes en los resultados obtenidos. Para esta comparación se aplicó nuestro método de detección de bordes utilizando 3 umbrales generales diferentes (10, 15 y 20), como se aprecia en los resultados mostrados en la Figura 12 literales (g), (h) e (i), respectivamente. En las secciones 6.5 y 6.6 se evaluaron los tiempos computacionales al ejecutar el algoritmo para la detección de bordes. Para obtener los tiempos de ejecución del algoritmo planteado se ejecutaron las pruebas en una PC con un CPU de 400 MHz y el tiempo se midió solamente sobre el algoritmo sin incluir las operaciones de entrada y salida, al igual que en la investigación realizada por [5]. La Tabla 1 y Figura 13 muestran los tiempos obtenidos por el algoritmo propuesto al evaluar las imágenes Camarógrafo (sección 6.5) y Lena (sección 6.6) con un tamaño de 256 x 256 píxeles en comparación con los de otros métodos evaluados (Roberts, Prewitt, Sobel, Canny y Kumar-Sahoo).

Del análisis realizado sobre los tiempos de cómputo, se puede destacar que los métodos en

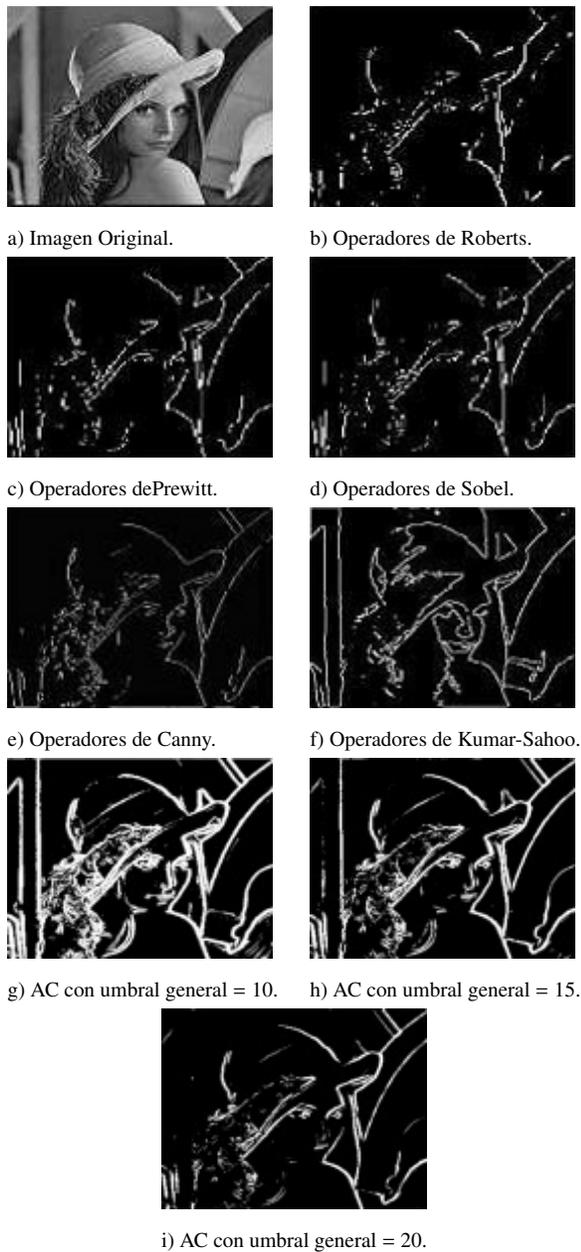


Figura 12: Resultados de detección de bordes Lena.

Tabla 1: Comparación de los tiempos de cómputo obtenidos por los métodos de detección de bordes.

	Métodos: Tiempo de Cómputo(seg)					
	Roberts	Prewitt	Sobel	Canny	Kumar-Sahoo	Método Propuesto
(1)	0.1804	0.2149	0.2306	0.2803	0.1406	0.0687
(2)	0.2006	0.2105	0.2505	0.2906	0.1506	0.0824

(1) Imagen Lena
(2) Imagen fotógrafo

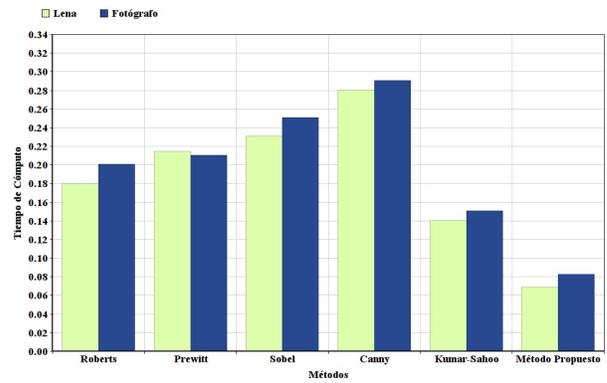


Figura 13: Gráfico comparativo de tiempos de cómputo obtenidos por los métodos de detección de bordes.

propuesto y método de Kumar-Sahoo) permiten obtener tiempos más bajos que el resto de los métodos. Sin embargo, los tiempos de cómputo obtenidos con el método propuesto son menores en un factor cercano a la mitad que los del método de Kumar-Sahoo. Esto se debe a que el trabajo cooperativo de las células vecinas en nuestra propuesta disminuye la cantidad de comparaciones realizadas por cada célula, trabajando con una vecindad formada por 4 células en lugar de 8 como lo hace el método de Kumar-Sahoo.

7. Conclusiones y trabajo futuro

El uso de autómatas celulares para la detección de bordes, es un método muy eficiente para realizar la fase de segmentación de imágenes, ya que permiten disminuir notablemente los tiempos de cálculo. Aunque se podría perder un poco el nivel de detalle, aún se sigue cumpliendo el objetivo de dividir la imagen en áreas de interés que finalmente puedan ser procesados y reconocidos como objetos por el computador. Esto puede ser utilizado para permitir al computador el reconocimiento de objetos, e incluso, la detección de texto dentro de las imágenes.

Por otro lado, la detección de bordes permite convertir imágenes con muchos colores y detalles, en imágenes simples con sólo dos o tres colores, donde resaltan los objetos que existen en la realidad, lo cual además de reducir el tamaño de las imágenes, permite disminuir el nivel de complejidad de la misma ya que elimina la

los cuales se utilizan autómatas celulares (método

información menos útil de la imagen, y en consecuencia es más sencillo procesarla por el computador.

De acuerdo con las imágenes resultantes del procesamiento se concluye que a mayor valor del umbral se detectan bordes en áreas que resaltan a simple vista por su forma y color, mientras que a medida que el valor del umbral decrece se detectan hasta los más pequeños detalles en cuanto a sombras y degradados de color. Sin embargo, cuando el umbral es muy pequeño, la gran cantidad de bordes encontrados puede causar pérdida en la calidad de la imagen resultante y en consecuencia, dificultad para detectar áreas que representan objetos específicos en la imagen. Dado que los componentes RGB cercanos a los bordes presentan una diferencia mayor al umbral, se generan bordes con un grosor mayor que el obtenido por otros métodos, contribuyendo en general con una mejor segmentación de los objetos contenidos en la imagen.

Adicionalmente, la comparación entre los diferentes métodos de segmentación indica que el método propuesto presenta alta capacidad para la detección de bordes y facilita la personalización de la segmentación al permitir la graduación del umbral de diferencia de los componentes RGB analizados por el autómata celular. Esto ayuda a calibrar el umbral hasta obtener el resultado esperado, eligiendo desde una imagen segmentada con bordes bien definidos y gran cantidad de zonas detectadas, hasta una imagen con bordes menos continuos pero con gran nivel de detalle.

Para futuras investigaciones se propone la calibración del umbral a través de alguna técnica, como por ejemplo un algoritmo genético, que permita elegir de forma óptima y completamente automática el valor del umbral de segmentación que mejor se adapte a cada imagen.

Referencias

- [1] Pritam Prakash Shete, PPK Venkat, Dinesh M Sarode, Mohini Laghate, SK Bose, and RS Mundada. Object oriented framework for cuda based image processing. In *Communication, Information & Computing Technology (ICCICT), 2012 International Conference on*, pages 1–6. IEEE, 2012.
- [2] Paul L. Rosin and Xianfang Sun. *Cellular automata in image processing and geometry*, chapter Edge detection using cellular automata, pages 85–103. Springer International Publishing, Cham, 2014.
- [3] Virginia Cifuentes, Emmanuel Larussi y Alejandro Clause. Método de Lattice-Boltzmann para segmentación de texturas de imágenes. *Mecánica Computacional*, XXXI:3027–3036, 2012.
- [4] Gholamreza Akbarizadeh. Segmentation of SAR satellite images using cellular learning automata and adaptive chains. *Journal of Remote Sensing Technology*, 1(2):44–51, 2013.
- [5] Tapas Kumar and G Sahoo. A novel method of edge detection using cellular automata. *International Journal of Computer Applications*, 9(4):38–44, 2010.
- [6] Priyanka Shotrya and Sanjeev Bhardwaj. Image segmentation using cellular automata: A technical survey. *International Journal Of Engineering And Computer Science*, 2(4):1268–1272, 2013.
- [7] Khadijeh Mirzaei, Hodayun Motameni, and Rasul Enayatifar. New method for edge detection and denoising via fuzzy cellular automata. *International Journal of Physical Sciences*, 6(13):3175–3180, 2011.
- [8] Fernando Ortiz Renilla. Creación de mapas visuales panorámicos mediante técnicas de visión artificial. Trabajo especial de grado, Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III de Madrid, España, 2012.