

# Algoritmo rápido de la transformada de Hough para detección de líneas rectas en una imagen

Fast algorithm of the Hough transform for straight line detection in an image

Luis Canul-Arceo,<sup>1</sup> José López-Martínez,<sup>1\*</sup> Lizzie Narváez-Díaz,<sup>1</sup>

<sup>1</sup> Facultad de Matemáticas, Unidad Multidisciplinaria Tizimín, Universidad Autónoma de Yucatán.  
Calle 48 s/n. CP 97700. Tizimín. Yucatán, México

\* Correo-e: jlopezm@uady.mx

## PALABRAS CLAVE:

transformada de Hough,  
detección de líneas, paralelismo,  
descomposición piramidal

## RESUMEN

La transformada de Hough es uno de los métodos más comunes para detectar formas (por ejemplo líneas) en el procesamiento digital de imágenes; sin embargo, la complejidad computacional de la transformada es alta si se realiza de forma secuencial (utilizando un solo procesador). En este trabajo se presenta un algoritmo rápido de la transformada de Hough para detectar líneas rectas en una imagen mediante una técnica de descomposición en la imagen de entrada. Implementado en forma paralela, esto permite cargas de trabajo balanceadas en los procesadores participantes, para evitar la sobrecarga computacional. Se presentan y discuten simulaciones por computadora, las cuales demuestran la eficiencia del algoritmo rápido propuesto con algunos ejemplos de imágenes.

## KEYWORDS:

Hough transform, line detection,  
parallel processing,  
pyramidal decomposition

## ABSTRACT

The Hough transform is one of the most common methods used to detect shapes (i.e. lines) in digital image processing. However, the computational complexity of the transform is high if performed sequentially (using a single processor). In this paper, we present a fast algorithm of the Hough transform to straight lines detection in an image, which use an image decomposition technique. Implemented in parallel computing, this technique enables balanced workload for the processors involved to avoid the computational overhead. Computer simulations are presented and discussed to show the efficiency of proposed algorithm with some images as examples.

## 1 INTRODUCCIÓN

Para los humanos es relativamente sencillo determinar patrones como líneas o curvas en los objetos que nos rodean; sin embargo, hacer que las computadoras puedan reconocer dichos patrones es una tarea un poco más complicada. La transformada de Hough (TH) es uno de los métodos más importantes en las áreas de procesamiento de imágenes y visión computacional, se utiliza ampliamente para detectar diversas formas, como líneas y curvas [1-3]. Aunque no es el único método para el reconocimiento de líneas, es una herramienta muy fácil de implementar y robusta para el ruido en imágenes reales [4]. Generalmente, la TH consiste en las siguientes fases: votación, localización de picos, determinación de los parámetros actuales y verificación, cada una de las cuales ha sido ampliamente descrita en la literatura [5]. En contraste, una de sus principales desventajas radica en el alto costo computacional que requiere para realizar los cálculos y el almacenamiento de los datos [4]. Debido a lo anterior, existe una gran variedad de trabajos que buscan reducir la complejidad computacional utilizando métodos paralelos [1, 6].

En este trabajo se propone un enfoque diferente de la paralelización de la TH, mediante una técnica de descomposición de la imagen de entrada. Dicha descomposición asegura una carga de trabajo homogénea entre los procesadores participantes en la paralelización. Lo anterior permite que los tiempos de trabajo en los procesadores sean aproximadamente iguales, de tal manera que se aproveche al máximo el tiempo de procesamiento en cada uno de ellos.

Para demostrar los resultados, el artículo se organiza de la siguiente forma: en la sección 2 se describe brevemente la TH, en la sección 3 se presentan los detalles de la paralelización, en la sección 4 se describen y discuten la simulación por computadora y los resultados experimentales, finalmente, en la sección 5 se muestran las conclusiones.

## 2 EL ALGORITMO DE LA TH

La TH se utiliza para detectar líneas en aplicaciones de procesamiento de imágenes, para lo cual, transforma los puntos del plano cartesiano a un espacio de parámetros. La literatura clasifica la TH en dos grandes grupos, dependiendo de la parametrización usada para representar líneas [7].

El primer grupo utiliza los parámetros de la pendiente  $a$  y el parámetro de intersección con la ordenada  $b$ , para lo cual se apoya en la representación de la recta dada por la siguiente ecuación:

$$y=ax+b, \tag{1}$$

Por cada punto del plano pasan infinitas rectas que satisfacen la ecuación anterior para diferentes valores de  $a$  y  $b$ . Al considerar el plano  $ab$ , denominado espacio de parámetros, se obtiene una única recta para cada punto; por ejemplo, si consideramos dos puntos  $(x_p, y_p)$  y  $(x_j, y_j)$ , cada punto tendrá asociada una única recta, que al intersectarse darán los parámetros de  $a$  y  $b$  de la recta que contiene los puntos colineales de  $(x_p, y_p)$  y  $(x_j, y_j)$ , tal como se muestra en la figura 1.

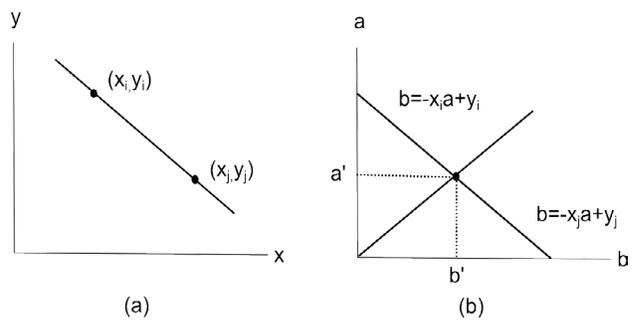


Figura 1. (a) Recta en el plano  $xy$ ; (b) espacio de parámetros

En la implementación de la TH se utilizan unas celdas acumuladoras que dividen el espacio de parámetros, esto permite evaluar todos los posibles valores por cada punto y de acumular los frecuentes (fase de votación). El resultado es la creación de una malla; mientras mayor sea el valor en cada casilla, significa que se ha encontrado una línea recta (fase de localización de picos en la malla)[8]. El inconveniente de utilizar el espacio de parámetros es la dificultad para detectar líneas verticales debido a que los parámetros pueden llegar a ser infinitos a medida que la línea se hace vertical.

El segundo grupo utiliza coordenadas polares y está compuesto por los parámetros  $\rho$  y  $\theta$ , con la siguiente ecuación:

$$\rho=x \cos\theta +y \sin\theta, \tag{2}$$

donde  $\rho$  es la distancia de la línea con el origen y  $\theta$  es el ángulo entre el vector con respecto al eje de las

abscisas. El comportamiento es similar a lo realizado en el primer grupo, la diferencia radica en que en vez de rectas, por cada punto estará asociado un senoide en el plano  $\rho\theta$ , como se muestra en la figura 2. Utilizando las coordenadas polares se soluciona el inconveniente del primer grupo.

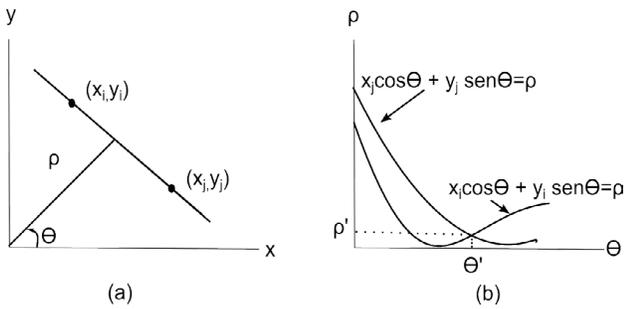


Figura 2. (a) Parametrización de las líneas en el plano  $xy$ ; (b) curvas sinusoidales en el plano  $\rho$

### 3 EL ALGORITMO RÁPIDO PROPUESTO

A continuación se presenta una técnica de descomposición de imágenes que se utiliza en el algoritmo propuesto. La técnica conocida como *decimation technique* consiste en descomponer una imagen con un factor  $L$  a través de filas y columnas en un conjunto de  $L^2$  imágenes, como se muestra en la figura 3.

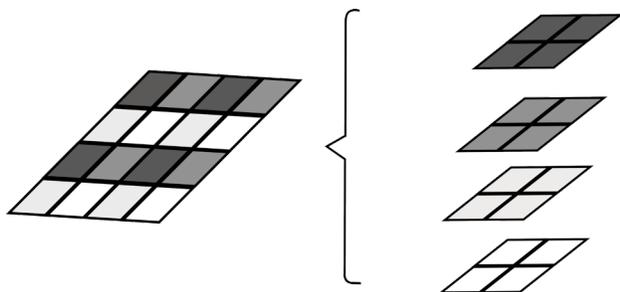


Figura 3. Descomposición de la imagen utilizando la técnica *decimation* con  $L=2$

Los pasos del algoritmo se resumen a continuación:

1. La imagen de entrada es de tamaño  $N \times N$  píxeles y se le aplica un filtro *Canny* [8]. El objetivo es extraer sus bordes principales.
2. Se realiza la descomposición de la imagen

anterior con un factor  $L=2$ , y se asigna su ejecución a  $p_i, i=1 \dots 4$  procesadores.

3. Cada procesador  $p_i$  realiza la TH en el espacio de parámetros de  $\rho\theta$  con la subimagen que se le asignó. Cada  $p_i$  procesadores tiene sus propias celdas acumuladoras, lo que genera una matriz de votación  $M_i$ . El tamaño de la  $M_i$  tiene los siguientes rangos  $-90 < \theta < 90$  y  $-N < \rho < N$ .
4. Al finalizar el procesamiento de los  $p_i$  debido a que la carga es balanceada, se ahorra tiempo en el proceso de sincronización, ya que los  $p_i$  consumirán aproximadamente el mismo tiempo al realizar su trabajo. Posteriormente, se realiza la unión de los resultados de las celdas acumuladoras, es decir,  $M_T = \sum_{i=1}^4 M_i$ . Por lo tanto, se obtiene un acumulador global.
5. Finalmente, a partir de la  $M_T$  se procede a encontrar los picos de las celdas mediante un umbral conocido, se obtienen los parámetros actuales y se verifican en la imagen de entrada.

En la tabla 1 se presenta el pseudocódigo del algoritmo propuesto, donde  $I$  es la imagen de entrada a la cual se le aplica el filtro *Canny* con la finalidad de obtener los bordes de la imagen. El resultado de aplicar el filtro se almacena en  $I_c$ . Posteriormente,  $I_c$  se divide aplicando la técnica de descomposición piramidal mencionada; como consecuencia, las imágenes resultantes se almacenan en  $I_D(i)$ , a la cual se le aplica la TH, y el resultado se guarda en  $M(i)$ , respectivamente. Como  $M(i)$  contiene los

Tabla 1. Pseudocódigo del método propuesto

```

I ← ImagenEntrada()
Ic ← AplicarFiltroCanny(I)
For i ← 1 to 4 do in paralelo
    ID(i) ← Descomponer-Ima(Ic)
    M(i) ← TFHough(ID(i))
    MT ← MT + M(i)
End For
P ← Hough-picos(MT)
IL ← Hough-lineas(I, P)
    
```

acumuladores parciales de la imagen de entrada, se suma al acumulador global  $M_T$ . Es muy importante señalar que tanto la descomposición, la aplicación de la TH, así como la suma de los resultados, se efectúan en paralelo. Como se había mencionado, la para-

lización se realiza utilizando cuatro procesadores. Para obtener los acumuladores máximos y trazar las líneas en la imagen original, se utilizan las funciones Hough-picos y Hough-líneas, respectivamente. El vector  $\mathbf{P}$  contiene los parámetros necesarios para trazar las líneas y la imagen original con las líneas detectadas; se encuentra en  $\mathbf{I}_L$ .

#### 4 RESULTADOS EXPERIMENTALES

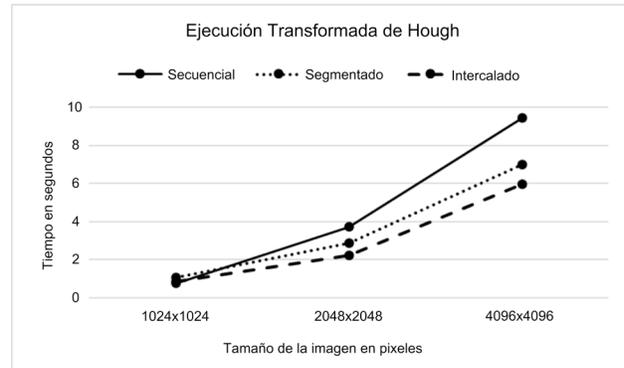
En esta sección se presentan los resultados experimentales de aplicar el algoritmo con una imagen de  $N \times N$  con diferentes tamaños en píxeles,  $N = \{1024, 2048, 4096\}$ . Los experimentos se realizaron en una computadora con procesador Intel Xeon con 4 núcleos, con 8GB de RAM. Se utilizó el software Matlab con el *toolbox* de paralelismo, en particular, la función *parfor* para realizar la descomposición de la imagen y la TH utilizando cuatro procesadores. Las imágenes de prueba se muestran en las figuras 4 y 7, las cuales corresponden al primer y segundo experimento, respectivamente.



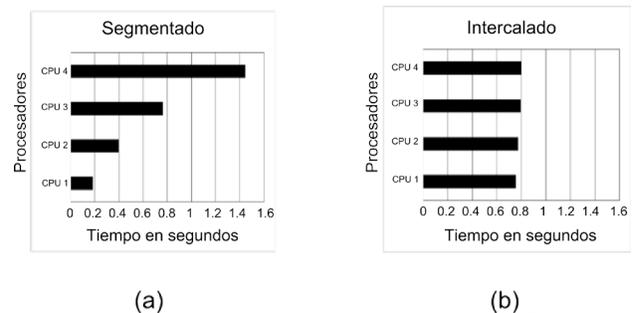
**Figura 4.** Imagen de González y Wood [8] utilizada en el primer experimento.

Con el objetivo de medir la eficiencia del método propuesto, el algoritmo se comparó con la TH secuencial y con el método estándar de dividir la imagen de entrada en cuatro cuadrantes (división por

segmentos). A la propuesta de método la llamaremos: división por intercalación, para diferenciarlo del anterior. Se puede ver en la figura 5 que el método propuesto es más rápido que la TH secuencial y que el método de descomposición mediante segmentos. El tiempo que se presenta en la figura 5 incluye la suma de los acumuladores de cada procesador a la matriz global de acumuladores.



**Figura 5.** Rendimiento del método de división por intercalación en términos de segundos y tamaño de la imagen de entrada para la figura 4.



**Figura 6.** Utilizando la figura 4 con un tamaño de 2048 x 2048 píxeles: (a) carga de trabajo para el algoritmo que utiliza el método de descomposición por segmentos; (b) carga de trabajo para el algoritmo con el método de descomposición intercalado.

En la figura 6 se muestran los resultados de las cargas de trabajo del método de división por segmentos contra el de división por intercalación para una imagen de 2048 x 2048 píxeles. Se puede observar que éste proporciona carga de trabajo homogénea a los procesadores, lo que minimiza el tiempo total de ejecución de la TH. Esto se debe a que, al realizar el método de descomposición por segmentos, es

posible que existan pocas líneas dentro de uno de los segmentos, lo que origina una carga heterogénea para los procesadores.

En la figura 8 se presenta el rendimiento del método por intercalación en comparación con el método secuencial y la descomposición por segmentos para la figura 7. En la figura 9 se muestra la comparativa entre el método propuesto y el de descomposición por segmentos utilizando una imagen de entrada de tamaño 2048 x 2048 pixeles; se puede apreciar claramente que la carga de trabajo y el tiempo de procesamiento es menor al método de descomposición por segmentos.

### 5 CONCLUSIONES

En este artículo se presentó un algoritmo rápido para la transformada de Hough utilizado para la detección de líneas rectas en una imagen de entrada. El algoritmo utiliza una técnica de descomposición intercalada aplicada a la imagen de entrada, lo que permite que la carga de trabajo entre los procesadores sea homogénea. Las simulaciones por computadora muestran que el algoritmo propuesto es más rápido que la TH secuencial y que otra técnica de descomposición de imágenes más común. Para un trabajo futuro, el algoritmo de descomposición intercalada será implementado en GPU para mejorar su tiempo de respuesta.



Figura 7. Imagen utilizada para el segundo experimento

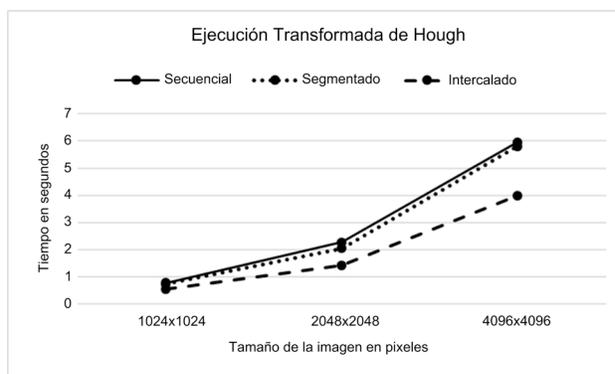


Figura 8. Rendimiento del método propuesto (intercalado) en términos de segundos y tamaño de la imagen de entrada para la figura 7

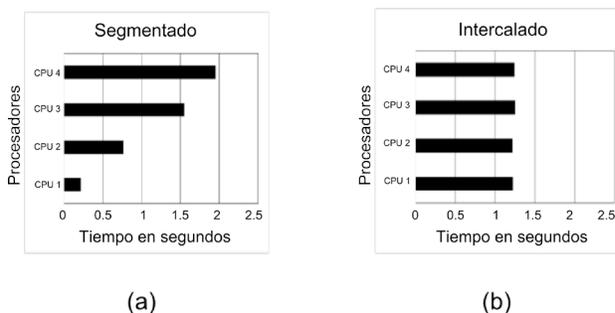


Figura 9. Utilizando la figura 7 con un tamaño de 2048 x 2048 pixeles, (a) carga de trabajo para el algoritmo que utiliza el método de descomposición por segmentos; (b) carga de trabajo para el algoritmo propuesto, el cual utiliza el método de descomposición intercalado.

## REFERENCIAS

1. Chen, L., Chen, H., Pan, Y., Chen, Y. A fast efficient parallel Hough transform algorithm on LARPBS. *The Journal of Supercomputing*. 2004, 29, 185-195.
2. Duda, R.O., Hart, P.E. Use of the Hough transform to detect lines and curves in pictures. *Commun. ACM*. 1972, 15(1), 11-15.
3. Zhu, T., Jeong-Hyun, K., Dong-Joong, K. Ellipse detection: a simple and precise method based on randomized Hough transform. *Optical Engineering*. 2012, 51(5).
4. Atiquzzaman, M. Multiresolution Hough transform: an efficient method of detecting patterns in images. *IEEE Transactions on pattern analysis and machine intelligence*. 1992, 14(11), 1090-1095.
5. Ji, J., Chen, G., Sun, L. A novel Hough transform method for line detection by enhancing accumulator array. *Pattern recognition letters*. 2011, 32, 1503-1510.
6. Hanahara, K., Matuyama, T., Uchiyama, T. A real-time processor for the Hough transform. *IEEE transactions on pattern analysis and machine intelligence*. 1988, 10(1), 121-125.
7. Gauil, N., Villalba, J., Zapata, E. A fast Hough transform for segment detection. *IEEE transactions on image processing*. 1995, 4(11), 1541-1548.
8. Gonzalez, R.C., Woods, R. E. *Digital Image Processing*. Upper Saddle River: Pearson / Prentice Hall, 2008.

### Acerca de los autores



Luis Canul-Arceo es pasante de la Licenciatura en Ciencias de la Computación de la Facultad de Matemáticas, Unidad Multidisciplinaria Tizimín de la Universidad Autónoma de Yucatán (UADY). Entre sus líneas de investigación se encuentran el procesamiento de imágenes, cómputo paralelo y reconocimiento de patrones.



José Luis López Martínez es Licenciado en Ciencias de la Computación, por la Universidad Autónoma de Yucatán, y recibió el grado de Doctor en Ciencias de la Computación en 2011 en el Centro de Investigación Científica y de Educación Superior de Ensenada, México. Actualmente es profesor en la Facultad de Matemáticas de la UADY. Sus áreas de investigación incluyen: el procesamiento de imágenes, cómputo paralelo y reconocimiento de patrones.



Lizzie Narváez-Díaz es Licenciada en Ciencias de la Computación por la Universidad Autónoma de Yucatán, y obtuvo la Maestría en Ciencias de la Computación en 2006 en el Instituto Tecnológico de Estudios Superiores de Monterrey, campus Cuernavaca. Actualmente es profesora de la Facultad de Matemáticas de la UADY. Su área de interés son las redes de computadoras.