

MÉTODO DE LATTICE-BOLTZMANN PARA SEGMENTACIÓN DE TEXTURAS DE IMÁGENES

Virginia Cifuentes^{a,c}, Emmanuel Iarussi^c y Alejandro Clause^{b,c}

^a*Comisión de Investigaciones Científicas de la Prov. de Buenos Aires (CIC-PBA),*

^b*Comisión Nacional de Energía Atómica (CNEA),*

^c*Instituto PLADEMA, Facultad de Ciencias Exactas, UNCPBA, Paraje Arroyo Seco-Campus Universitario, Tandil, Argentina, <http://www.pladema.gov.ar>*

Palabras clave: Lattice-Boltzmann, segmentación de imágenes, detección de texturas

Resumen. Se propone una técnica para la segmentación de imágenes digitales por textura basada en el método de Lattice-Boltzmann (LB) operando sobre la grilla de píxeles. LB es un autómata celular que representa el estado de cada celda (píxel) con 9 escalares que representan poblaciones de pseudo partículas transportándose entre celdas vecinas. En la presente aplicación se utilizó un autómata LB con un kernel lineal local de mezcla entre poblaciones. Esto implica que las partículas no interactúan entre sí sino con el medio en el que se propagan, el cual se representa usando las distribuciones de gris de los píxeles de cada vecindad. La distribución estacionaria de las poblaciones de fotones es usada para caracterizar las texturas por región. Se encontró que cada textura induce una relación particular entre la entropía y la velocidad media local, como una especie de ecuación de estado textural, la cual puede ser aplicada como criterio de segmentación por regiones. En experimentos numéricos preliminares sobre texturas imágenes sintéticas y naturales se alcanzaron calidades de segmentación del 95% y 85% respectivamente.

1 INTRODUCCIÓN

La identificación automática de texturas en imágenes digitales es una operación difícil de lograr usando técnicas clásicas de segmentación de imágenes, como umbralado o modelos deformables. Algunas aplicaciones de esta operación son el diagnóstico médico o la inspección industrial no destructiva asistidas por imágenes, la zonificación de regiones (Haralick et al, 1973), la discriminación de áreas forestadas (Oliver, 2000), entre otros. Se han propuesto diferentes algoritmos para la detección de texturas, como la combinación de métodos de segmentación por umbralado combinado con filtros de suavizado (Shapiro & Stockman, 2001). En general hasta el presente los resultados exitosos se restringen a dominios muy específicos y limitados.

Una técnica interesante para la caracterización de texturas en imágenes digitales es la aplicación de autómatas celulares a la matriz de píxeles que forma cada imagen. Los primeros trabajos en esta dirección se remontan a la década del 70 con aplicaciones de caminadores aleatorios (Wechsler & Kidode, 1979). Recientemente Grady (2006) experimentó con un algoritmo de caminadores en imágenes médicas.

Una alternativa más flexible que los caminos aleatorios son los autómatas de Lattice Boltzmann (LB), que representan el estado de cada celda con múltiples poblaciones de partículas que se mueven entre vecindades siguiendo reglas determinísticas (Chen & Doolen, 1998). Aunque las principales aplicaciones de LB están en la simulación de fluidos, recientemente se lo ha utilizado en la creación de modelos de iluminación (Geist & Westall, 2011; Maso et al, 2010). En principio es posible aplicar LB a grillas formadas por píxeles de una imagen, en forma análoga a los caminadores aleatorios. Los estudios preliminares han mostrado resultados interesantes en esta dirección (Iarussi et al, 2011).

En este trabajo se presenta un autómata de LB con un kernel lineal local de mezcla entre poblaciones. Esto implica que las partículas no interactúan entre sí sino con el medio en el que se propagan, el cual se representa aquí en función de las distribuciones de gris de los píxeles de cada vecindad. La distribución estacionaria de las poblaciones de fotones es usada luego para caracterizar las texturas por región.

2 MÉTODO DE LATTICE-BOLTZMANN

Lattice Boltzmann (LB) es una clase de modelos cinéticos discretos que incorpora mecanismos de procesos mesoscópicos a nivel de vecindades, de manera que las propiedades macroscópicas promediadas obedezcan las ecuaciones de transporte deseadas (Chopard & Droz, 1998; Frisch et al., 1986 ; Giraud et al., 1998; He et al., 1998; Higuera et al., 1989). Recientemente se ha extendido la aplicación de LB en numerosas ramas de la ingeniería (García et al., 2010; Golbert et al, 2012; Rinaldi et al, 2012).

Esencialmente, LB consiste en una representación discreta del espacio mediante una grilla regular de celdas cuyo estado está caracterizado por una población de partículas. El estado de cada celda se expresa con un campo escalares, llamado función distribución, $f_i(x,t)$ (no observable directamente) que representa la cantidad de partículas en la celda x en el tiempo t moviéndose en direcciones discretas e_i (tratadas como variables internas). Los observables físicos son variables macroscópicas que se generan a partir de los momentos de $f_i(x,t)$ respecto de la variable e_i .

La función distribución cambia de estado según reglas explícitas que se emulan mecanismos de transporte y colisión de partículas. Hay diferentes formulaciones matemáticas de estas reglas. Para fluidos en general la regla requiere el cálculo intermedio de los momentos de la función distribución. También pueden expresarse las reglas de evolución

como una transición de estados, en forma análoga a como se define un autómata finito, es decir:

$$f_i(\vec{x} + \vec{e}_i, t + 1) = \sum_{j=0}^9 m_{ij}(\vec{x}) f_j(\vec{x}, t) \quad (1)$$

donde $f_i(\vec{x}, t)$ es la población de partículas en la celda \vec{x} en el paso temporal t que se desplazan en la dirección \vec{e}_i y m_{ij} es la probabilidad de transición entre las direcciones i y j (Maso et al 2010). El conjunto de vectores \vec{e}_i define el vecindario de un píxel, y sus valores son: $\vec{e}_0 = (0,0)^T$, $\vec{e}_1 = (1,0)^T$, $\vec{e}_2 = (1,1)^T$, $\vec{e}_3 = (0,1)^T$, $\vec{e}_4 = (-1,1)^T$, $\vec{e}_5 = (-1,0)^T$, $\vec{e}_6 = (-1,-1)^T$, $\vec{e}_7 = (0,-1)^T$ y $\vec{e}_8 = (1,-1)^T$.

En el presente trabajo, se determinaron los valores de m_{ij} a partir de la textura de imagen digital estudiada. El objetivo es que las texturas se manifiesten en la función de distribución $f_i(\vec{x}, t)$ a través de la iteración del autómata LB dado por la Ec. (1). Cada píxel \vec{x} de la imagen se asocia una matriz m_{ij} de dimensiones 9×9 , que se construye en base a los valores de intensidad de los pixeles correspondientes del vecindario de \vec{x} , es decir:

$$m_{ij}(\vec{x}) = F[g(\vec{x} + \vec{e}_i), g(\vec{x} + \vec{e}_j)] \quad (2)$$

donde $g(\vec{x})$ es la intensidad de gris del píxel \vec{x} .

Se pueden definir variantes de m_{ij} que tengan en cuenta vecindarios más grandes. En nuestro caso esto se obtiene mediante un suavizado por promedio espacial m_{ij}^r con un radio de influencia r , es decir:

$$m_{ij}^r(\vec{x}) = \sum_{\rho=0}^r \sum_{k=0}^9 w_r(\vec{x} + \rho \vec{e}_k) m_{ij}(\vec{x} + \rho \vec{e}_k) \quad (3)$$

donde $w_r(\vec{x})$ es un coeficiente de ponderación que puede usarse para dar mayor peso a un píxel de acuerdo a su intensidad — es decir, $w_r(\vec{x})$ proporcional a $g(\vec{x})$.

Una vez construida las matrices $m_{ij}(\vec{x})$ se itera la Eq. (1) hasta llegar al estado estacionario. A partir de la distribución estacionaria $f_i(\vec{x})$ se pueden definir imágenes transformadas $g'(\vec{x})$ que serán usadas para investigar zonificaciones texturales. Las transformaciones estudiadas fueron las siguientes:

$$n(\vec{x}) = \sum_{i=0}^9 f_i(\vec{x}) \quad \text{Densidad} \quad (4.1)$$

$$\vec{u}(\vec{x}) = \frac{1}{n(\vec{x})} \sum_{i=0}^9 \vec{e}_i f_i(\vec{x}) \quad \text{Velocidad} \quad (4.2)$$

$$E(\vec{x}) = \sum_{i=0}^9 [\vec{u}(\vec{x}) - \vec{e}_i]^2 f_i(\vec{x}) \quad \text{Energía interna} \quad (4.3)$$

$$H(\vec{x}) = - \sum_{i=0}^9 \frac{f_i(\vec{x})}{n(\vec{x})} \ln \left[\frac{f_i(\vec{x})}{n(\vec{x})} \right] \quad \text{Entropía} \quad (4.4)$$

3 MÉTODO DE LATTICE-BOLTZMANN

Para evaluar la utilidad de los campos transformados dados por las Ecs. (4.1) se investigaron las siguientes instancias de la Ec. (2):

$$m_{ij}(\vec{x}) = N[g(\vec{x} + \vec{e}_i) + g(\vec{x} + \vec{e}_j)] \quad (5.1)$$

$$m_{ij}(\vec{x}) = N[g(\vec{x} + \vec{e}_i) g(\vec{x} + \vec{e}_j)] \quad (5.2)$$

$$m_{ij}(\vec{x}) = N[|g(\vec{x} + \vec{e}_i) - g(\vec{x} + \vec{e}_j)|] \quad (5.3)$$

$$m_{ij}(\vec{x}) = N[g(\vec{x} + \vec{e}_i)/g(\vec{x} + \vec{e}_j)] \quad (5.4)$$

donde $N[\cdot]$ indica la operación de normalización sobre el índice i , es decir:

$$N[a_{ij}] = a_{ij} / \sum_{i=0}^9 a_{ij}$$

lo cual asegura que m_{ij} sea una matriz estocástica y que, por lo tanto, se conserva el número total de partículas.

Se estudiaron varias figuras sintéticas y naturales, las cuales fueron procesadas previamente de manera que la intensidad total de cada vecindario de 3×3 sea uniforme. De esa manera se asegura que no es posible segmentar por umbralado directo del campo de grises. En los bordes de las imágenes se impusieron condiciones de contorno periódicas.

En la [Figura 1\(a\)](#) se muestra el primer caso estudiado, que consiste en un fondo sintético sobre el cual se superpuso una región de una foto de follaje natural. La mejor discriminación se obtiene con la Ec. (5.1) (operador suma) y el filtro por promedio (Ec. (3)) sobre un $radio r = 4$ (es decir sobre vecindarios de 11×11), usando pesos $w_r(\vec{x}) = g(\vec{x})$.

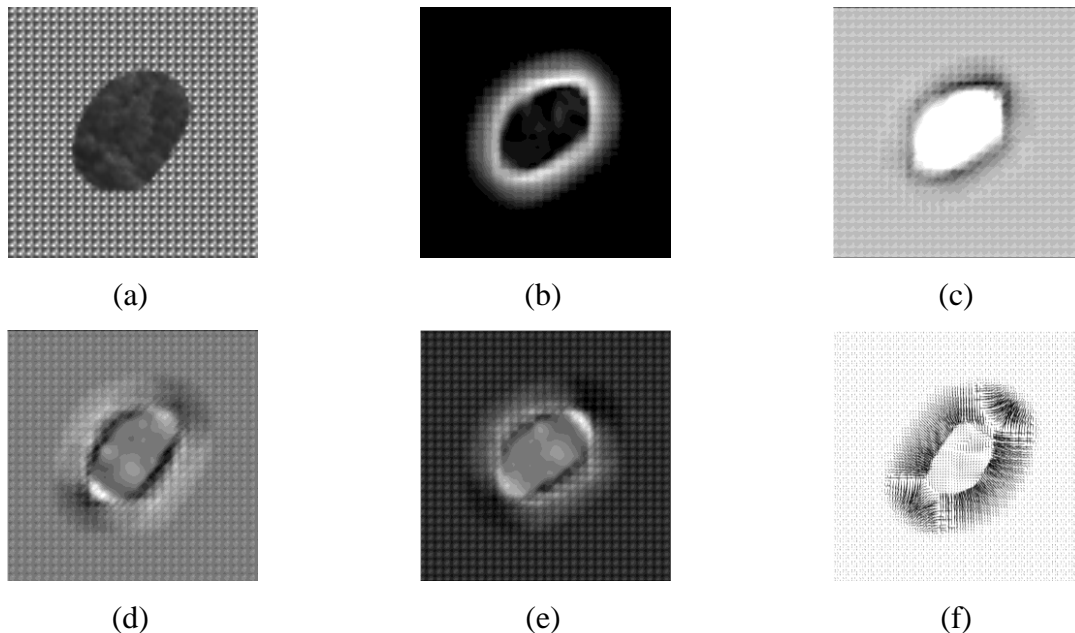


Figura 1 Caso de estudio 1. Imagen original (a), módulo de la velocidad (b), entropía (c), energía (d), densidad (e), vector de velocidades (f).

Las [Figura 1 \(b\) a \(f\)](#) muestran las imágenes transformadas con las magnitudes dadas por las Ecs. (5.1): el módulo de la velocidad (b), la entropía (c), la densidad (d), la energía interna (e) y el vector de velocidades (f). Si bien la frontera entre ambas texturas no se logra definir claramente debido al promediado por vecindario que hace que las matrices generadas en los

bordes entre texturas sean una mezcla de las texturas que lindan con el borde, las variables calculadas muestran la detección de los bordes presentes en la imagen, evidenciando claramente la separación entre las dos regiones.

Dado que los momentos calculados con las Ecs. (5.1) pueden asociarse al estado termodinámico y fluidodinámico de la población de pseudo-partículas representadas por la función $f_i(\vec{x})$, se investigaron las correlaciones entre los diferentes momentos. En la [Figura 2](#) se muestra el gráfico de la entropía en función del módulo de la velocidad obtenidos con la Ec. (5.1) y radios de filtrado $r = 3$ y $r = 4$. Cada punto corresponde a un pixel del fondo (negros) o de la textura superpuesta (rojos). Es muy interesante observar que cada textura parece inducir un “estado termofluidodinámico” en la población de equilibrio de las pseudo-partículas, y que ese estado tiende a seguir una ley del tipo:

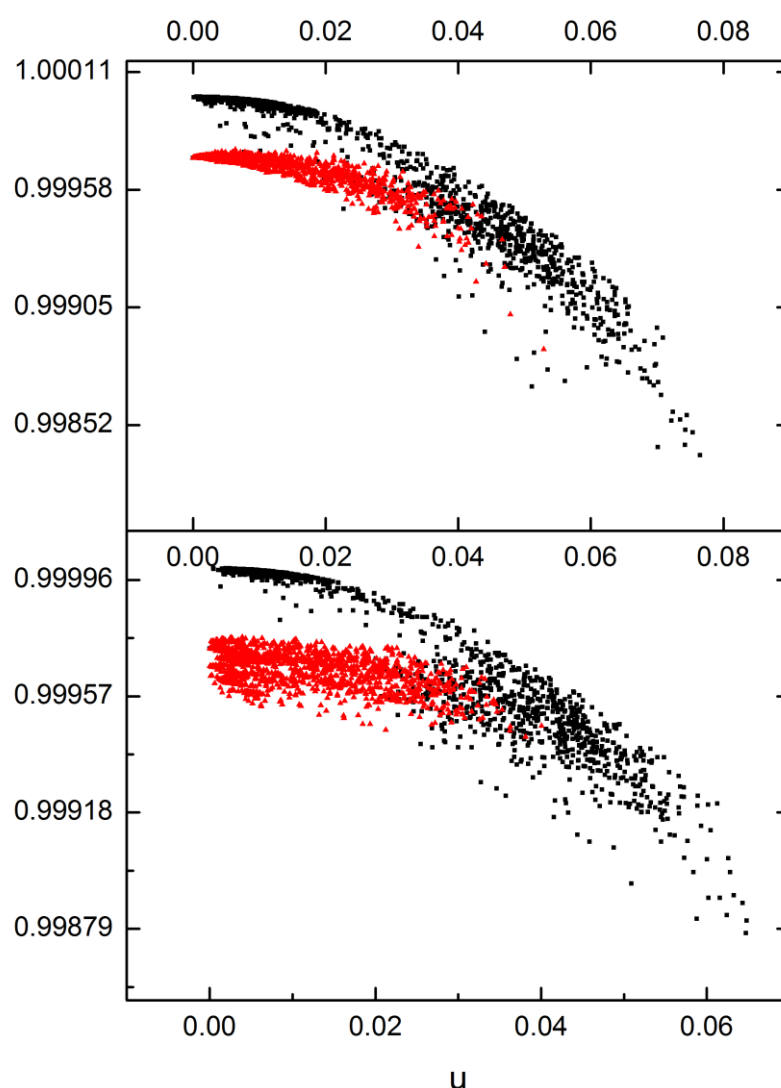


Figura 2 Mapas de entropía y velocidad correspondiente a la imagen de la [Figura 1](#). Vecindarios de 9x9 pixeles (arriba), 11x11 pixeles (abajo)

$$H = H_o \left(1 - \frac{u^2}{u_o^2} \right) \quad (6)$$

Esta ley es notablemente similar a la ecuación de Bernoulli que relaciona la presión con la

densidad de energía cinética de una línea de corriente de un fluido.

Los valores de H_o y u_o pueden ser utilizados para discriminar texturas por umbralado. Es decir, se genera una imagen binaria $I(\vec{x})$ de acuerdo a:

$$I(\vec{x}) = \begin{cases} 0 & \text{si } \frac{H}{H_o} + \frac{u^2}{u_o^2} < 1 \\ 1 & \text{en otro caso} \end{cases} \quad (7)$$

Variando los valores de los parámetros H_o y u_o se obtienen diferentes segmentaciones. Cada segmentación puede ser ranqueada usando el índice de calidad *AOM* (del Fresno, 2008) que mide el grado de intersección de cada segmentación con una segmentación de referencia (que en este caso se conoce por construcción). La Figura 3 muestra la variación del índice *AOM* con el parámetro u_o . Se puede observar que el mejor ajuste de la segmentación para radios de filtrado iguales a $r = 3$ y $r = 4$ se obtiene cuando $H_o = 0.99987$ y $H_o = 0.99997$ respectivamente. El máximo índice *AOM* es 0.94. A la derecha de la Figura 3 se muestra la segmentación óptima correspondiente a $u_o = 1.829$ ($r = 3$) y $u_o = 1.454$ ($r = 4$).

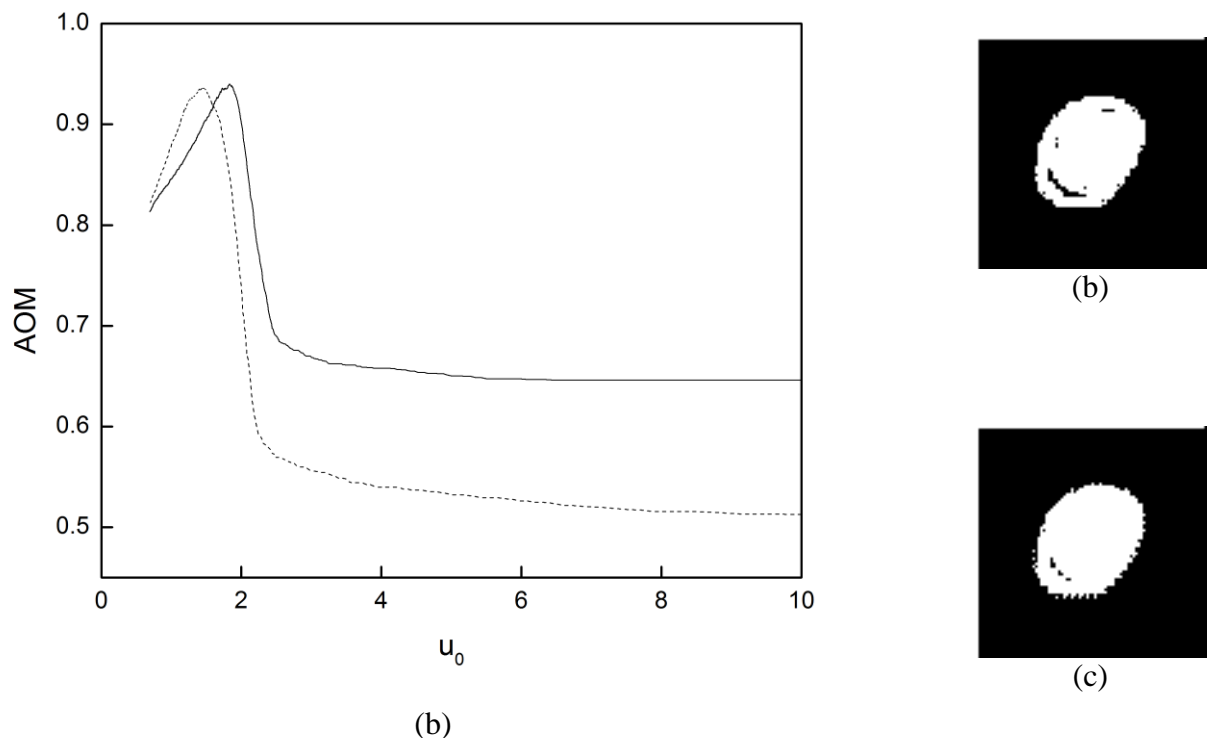


Figura 3 Selección del tamaño del vecindario. Abajo, segmentaciones correspondientes al máximo *AOM* en $r = 3$ (línea sólida, b) y $r = 4$ (líneas de puntos, c).

El segundo caso que se estudió fue la imagen digital natural mostrada en la Figura 4. Se aplicó la metodología propuesta observando que el tamaño del vecindario guarda relación directa con la discriminación de ambas texturas. En la Figura 5 se muestra el mapa de entropía y módulo de la velocidad, donde se puede observar que la separación entre ambas texturas se hace más notoria al adoptar tamaños de vecindario mayores.



Figura 4 Imagen natural compuesta por textura de pino y eucaliptus.

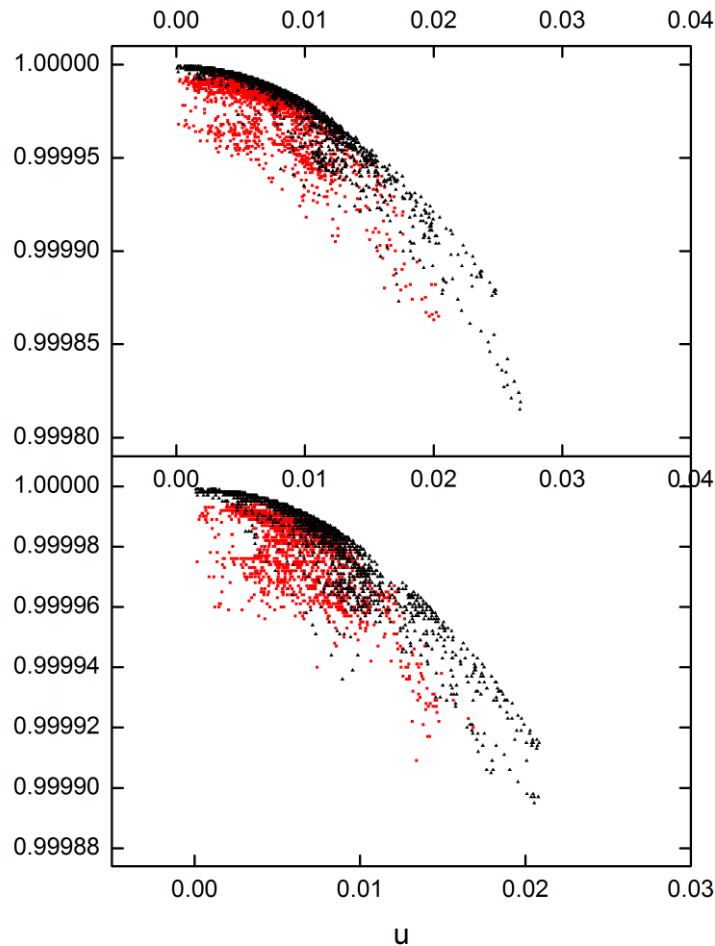


Figura 5 Mapas de entropía y velocidad correspondiente a la imagen de la [Figura 4](#). Vecindarios de 17x17 píxeles (arriba) y de 21x21 píxeles (abajo).

Se calculó el índice AOM cuando $H_o = 0.99998$ para diferentes valores de u_o . El mejor ajuste de la segmentación se logró para valores de $u_o = 2.094$ para un promedio de 17x17 píxeles y de $u_o = 2.167$ para un promedio de 21x21 píxeles ([Figura 6](#)). En la [Figura 7](#) se muestra la segmentación final para cada máximo AOM para promedios de 9x9, 11x11, 13x13, 15x15, 17x17 y 21x21 píxeles.

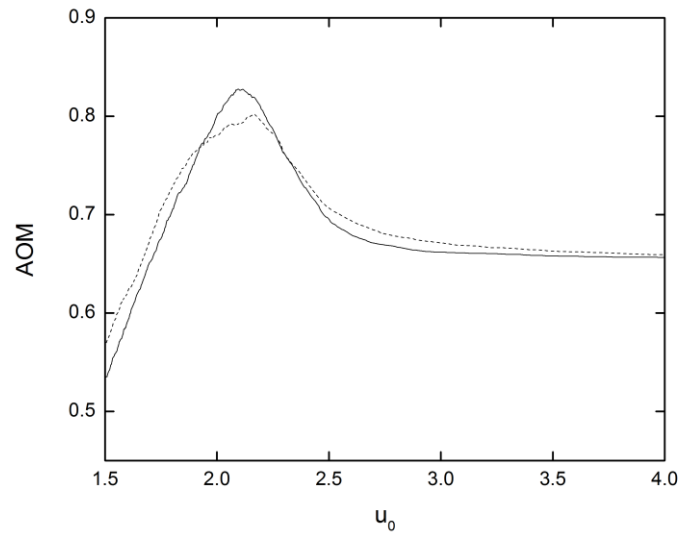


Figura 6 Selección del tamaño de vecindario para promedios de 17x17 (sólido) y 21x21 (línea de puntos) píxeles.



Figura 7 Segmentación de la Figura 4 para promedios de 9x9, 11x11, 13x13, 15x15 y 17x17 píxeles.

La tercera imagen estudiada está compuesta sólo de texturas naturales de un bosque de pinos. Puede apreciarse que, en el sector superior derecho de la imagen, se incorporó a la imagen una textura de pinos aún más densificada cuya forma coincide con una herradura invertida (ver Figura 8). En la Figura 9 se muestra el mapa de entropía y módulo de la velocidad obtenida con un promedio de 17x17 píxeles. La segmentación puede observarse en la Figura 10.

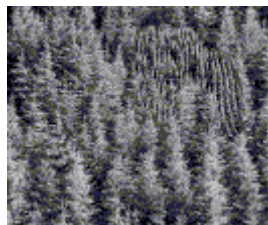


Figura 8 Imagen natural compuesta por texturas a distinta resolución.

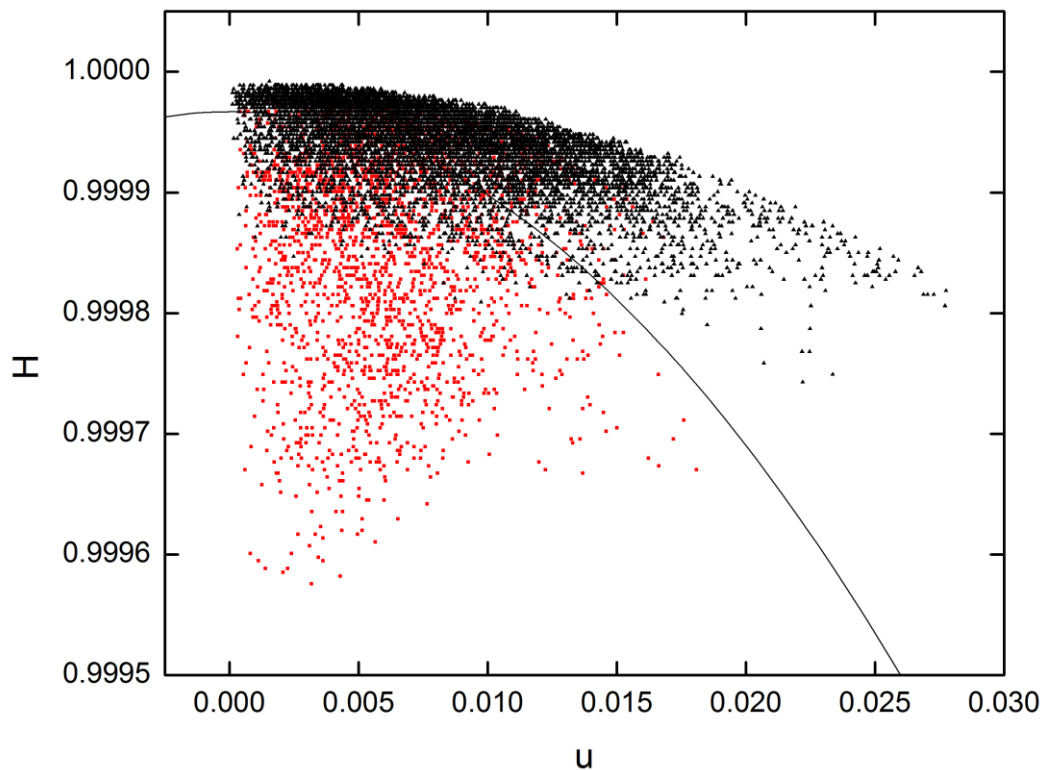


Figura 9 Mapas de entropía y velocidad correspondiente a la imagen de la [Figura 8](#) para promedios de 17x17 píxeles.



Figura 10 Segmentación según la parábola de la [Figura 9](#).

4 CONCLUSIONES

Se presentó una técnica para la detección y caracterización de texturas en imágenes digitales aplicando el método de lattice-Boltzmann caracterizado con matrices de transición construidas a partir de las imágenes originales. Se realizaron experimentos numéricos explorando diferentes alternativas del método, encontrándose que los campos macroscópicos resultantes aportan información útil para la segmentación de las texturas. Los resultados preliminares son promisorios y se espera continuar con la exploración de variantes que incrementen la potencialidad en el reconocimiento de texturas y patrones en dominios de imágenes más extendidos.

REFERENCIAS

- Chen S., Doolen G. D.: Lattice Boltzmann Method for Fluid Flows. *Annual Reviews Fluid 1102 Mechanics* 30:329-64, 1998.
- Chopard, B., y Droz, M.: Cellular Automata Modeling of Physical Systems. *Cambridge University Press*, Cambridge, UK, 1998.
- del Fresno M, Vénere M, and Clause A.: A combined region growing and deformable model method for extraction of closed surfaces in 3D CT and MRI scans. *Computerized Medical Imaging and Graphics*, 33(5):369-376, 2009.
- Frisch, U., Hasslacher, B., Pomeau, Y.: Lattice-Gas Automata for the Navier-Stokes Equation. *Physical Review Letters* 56:1505-1508, 1986.
- García, C., Boroni, G., Vénere M., Clause A.: Real-time Interactive Animations of Liquid Surfaces with Lattice-Boltzmann Engines. *Australian Journal of Basic and Applied Sciences* 4:3730-3740, 2010.
- Geist, R., and Westall, J.: Lattice-Boltzmann Lighting Models, *GPU Computing GEMS Emerald Edition*, Wen-mei Hwu, Editor, Morgan Kaufmann, 2011.
- Giraud, L., D'Humieres, D., Lallemand, P.: A Lattice Boltzmann Model for Jeffreys Viscoelastic Fluid. *Europhysics Letters*, 42:625-630, 1998.
- Golbert, D., Blanco, P., Clause, A., Feijoó, R.: Tuning a Lattice-Boltzmann Model for Applications in Computational Hemodynamics. *Medical Engineering and Physics* 34:339-349, 2012.
- Grady, L.: Random Walks for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(11):1-17, 2006.
- Haralick, R. M., and Shanmugam, K.: Computer Classification of Reservoir Sandstones. *IEEE Transactions on Geoscience Electronics* 11:171-177, 1973.
- He, X., Chen, S., Doolen, G.: A Novel Thermal Model for the Lattice Boltzmann Method in Incompressible Limit. *Journal of Computational Physics* 146:282-300, 1998.
- Higuera, F., Succi, S., Benzi, R.: Lattice Gas Dynamics with Enhanced Collisions. *Europhysics Letters* 9: 345-349, 1989.
- Iarussi, E., Cifuentes, V. y Clause, A.: Modelo de Iluminación en Medios Difusos basado en Autómatas de Lattice-Boltzmann para la Detección de Texturas en Imágenes Digitales. *Mecánica Computacional* (ISSN:1666-6070) 30:1915-1924, 2011.
- Maso G, Bulant C, D'Amato J, Clause A, Pomarico J, Iriarte D, Di Rocco H.: Autómata de Lattice Boltzmann para Modelar la Difusión Óptica en Materiales Traslúcidos. *39º Jornadas de Informática e Investigación Operativa* (ISSN 1850-2849) 39:1860-1867, Buenos Aires, 2010.
- Olivier, C.: Rain Forest Classification based on SAR Texture. *IEEE Transactions on Geoscience and Remote Sensing* 38(2):1095-1104, 2000.
- Rinaldi, P., Dari, E., Vénere, M., Clause, A.: A Lattice-Boltzmann Solver for 3D Fluid Simulation on GPU. *Simulation Modelling Practice and Theory* 25:163-171, 2012.
- Shapiro, L. G., and Stockman, G. C.: *Computer Vision*, Upper Saddle River: Prentice-Hall, 2001.
- Wechsler, H., and Kidode, M.: A Random Walk Procedure for Texture Discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):272-280, 1979.