Autómatas Celulares

Trabajo Practico Nro. 2

Badi Leonel, Buchhalter Nicolás Demián y Meola Franco Román

7 de abril de 2016

Fundamentos
Introducción

- Autómata Off-Latice de Bandadas de Agentes Autopropulsados.
- Basado en el algoritmo del trabajo de *Novel type of phase transition in a system of self-driven particles*.
- Cada agente será representado por un vector de velocidad cuyo origen estará ubicado en la posición de la partícula para cada tiempo de la simulación t.

Fundamentos
Variables relevantes

- N : cantidad de agentes
- L : longitud del lado del área de simulación
- r_c : radio de interacción de las partículas
- v : módulo de la velocidad
- $\rho = \frac{N}{L^2}$: densidad
- η : amplitud del ruido
- v_a : parámetro de orden
- Nos interesa obtener las curvas de v_a en función de η y ρ.

Implementación
Generación de los agentes

- Posiciones (x, y) aleatorias para cada agente.
- $r = 0{,}1$
- $v_{cte} = 0{,}3$
- $\theta_0 = rand() * 2\pi$ (ángulo inicial aleatorio)
- $\delta_\theta = (rand() * \eta) - \frac{\eta}{2}$

Implementación
Modelo del Agente

Una partícula está compuesta por:

- (x, y): Posición
- $|v|$: Módulo de la velocidad
- θ : Ángulo
- Lista de agentes vecinos

Implementación
Modelo del Sistema

El sistema está compuesto por:

- N
- L
- r_c
- Longitud del lado de un elemento de la grilla
- Lista de partículas
- Arreglo de listas de partículas

Implementación
Simulación

```
for(Frame frame : frames) {
    List<Particle> particles = calculateNeighbours();
    for(Particle particle : particles) {
        particle.updateAngleAndPosition();
        if(particle.isInBorderPosition()) {
            particle.makePeriodicTraslation();
        }
    }
    saveFrameData();
}
```

Código 1: Algoritmo de simulación de partículas.

Implementación
Visualización

- La simulación y la visualización son independientes
- El algoritmo de simulación escribe un archivo `.tsv` con los siguientes datos:
  - (x, y)
  - r
  - Color RGB para indicar las velocidades, donde R es la componente en el eje Y y G es la componente en eje X
- Por último, se carga en `Ovito` el archivo de salida `.tsv` para hacer la visualización

Resultados

$N = 40$, $L = 3{,}1$ y $r_c = 1$

η	v_a
0.0	1.00000
0.5	0.98799
1.0	0.95777
1.5	0.89056
2.0	0.82847
2.5	0.71321
3.0	0.55152
3.5	0.53482
4.0	0.42995
4.5	0.38458
5.0	0.11822

Tabla: Datos de la curva de v_a para $N = 40$, $L = 3{,}1$ y $r_c = 1$.

Resultados

$N = 400$, $L = 10$ y $r_c = 1$

η	v_a
0.0	0.99999
0.5	0.97909
1.0	0.87444
1.5	0.78458
2.0	0.70671
2.5	0.25537
3.0	0.46229
3.5	0.35985
4.0	0.13354
4.5	0.06530
5.0	0.04682

Tabla: Datos de la curva de v_a para $N = 400$, $L = 10$ y $r_c = 1$.

Resultados

$N = 10000$, $L = 50$ y $r_c = 1$

η	v_a
0.0	0.92861
0.5	0.07030
1.0	0.44647
1.5	0.46268
2.0	0.26689
2.5	0.20727
3.0	0.03786
3.5	0.05376
4.0	0.00901
4.5	0.00760
5.0	0.00412

Tabla: Datos de la curva de v_a para $N = 10000$, $L = 50$ y $r_c = 1$.

Resultados
Gráfico

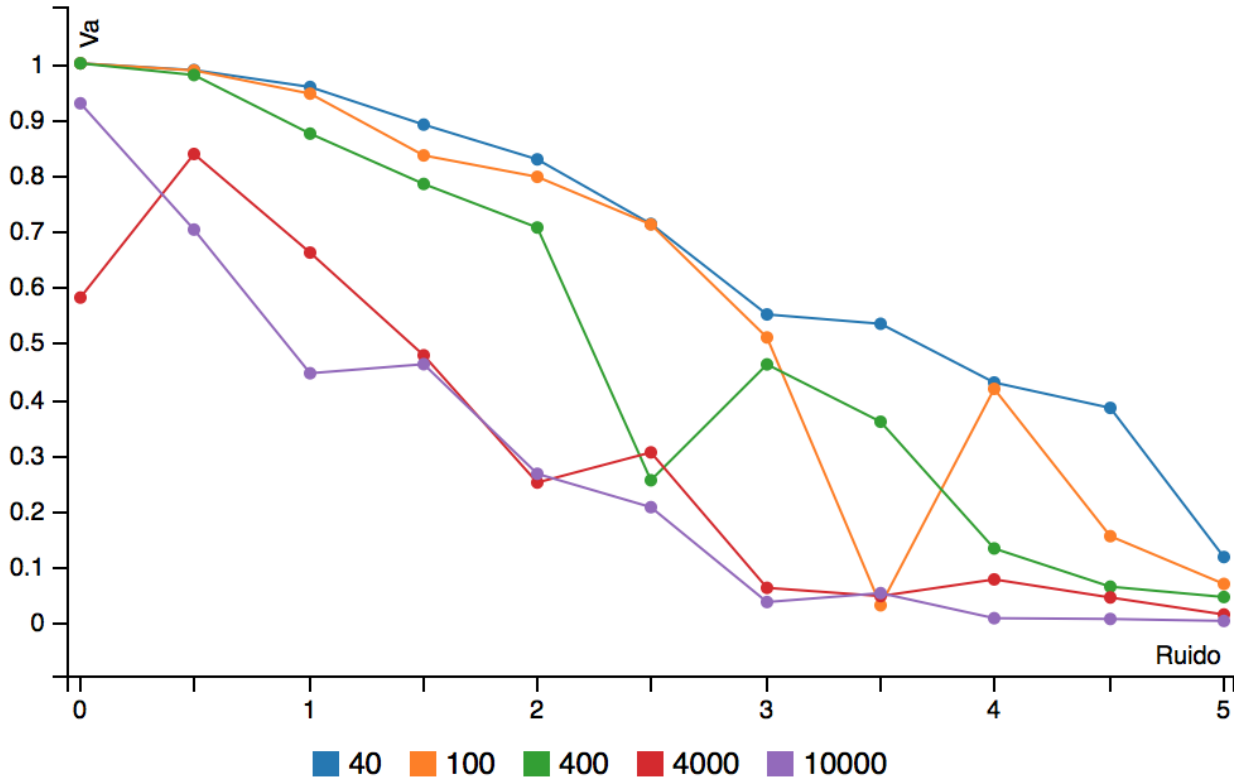

Grafico: Gráfico de la curva de v_a para distintos valores de N, L y η.
http://bl.ocks.org/lbadi/05bfacbb96e14df499a7ef18b37ca2cc

Resultados
Animaciones

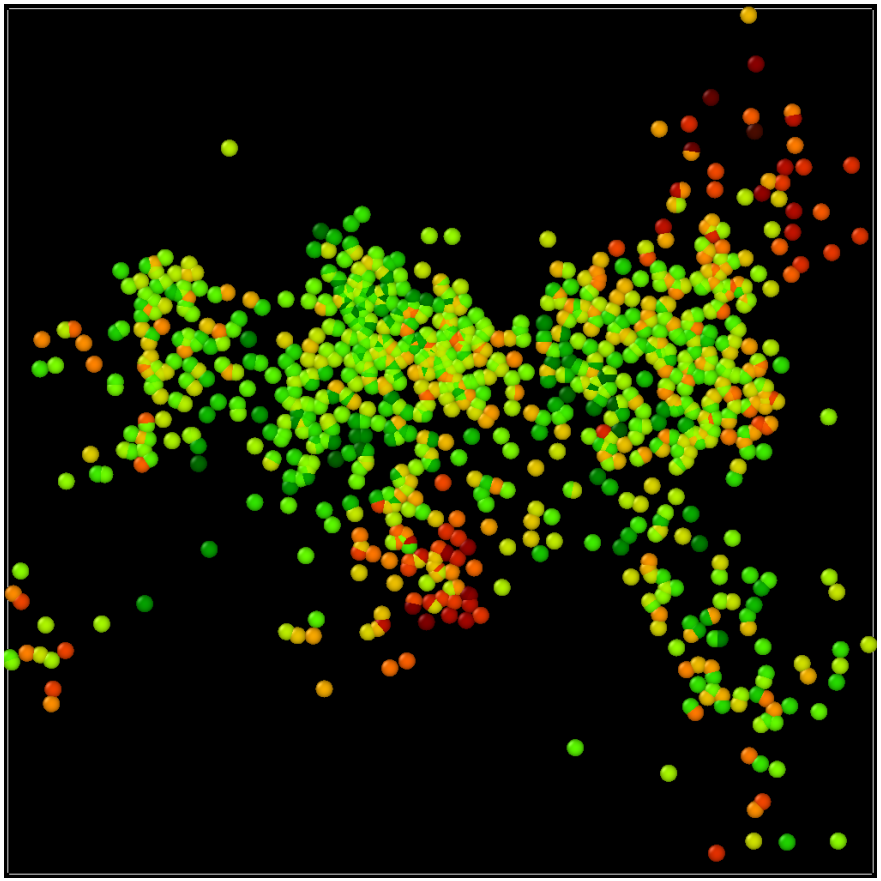

Grafico: Ver archivos `.avi` en la demostración en vivo.