

# Práctica 2

# Mapas

# métricos

Jordi Blasco Lozano

Razonamiento y representación del conocimiento

Grado en Inteligencia Artificial

---

## Indice:

<b>Indice:</b>	<b>2</b>
<b>1. Objetivo de la práctica</b>	<b>3</b>
<b>2. Consideraciones y datos de entrada</b>	<b>3</b>
<b>3. Métodos implementados</b>	<b>4</b>
<b>Rejilla de Ocupación</b>	<b>4</b>
<b>Oc-Tree</b>	<b>4</b>
<b>4. Lógica común y detalles de implementación</b>	<b>5</b>
<b>Lectura de ficheros PCD</b>	<b>5</b>
<b>Estructura de Almacenamiento de Datos</b>	<b>5</b>
<b>5. Comparación y cálculo de estadísticas</b>	<b>9</b>
<b>6. Parte Optativa: Visualización 3D con PyVista</b>	<b>13</b>

---

## 1. Objetivo de la práctica

El objetivo de esta práctica es almacenar y analizar datos métricos en 3D, provenientes de ficheros .pcd (Point Cloud Data), mediante dos estructuras de datos ampliamente usadas en Robótica:

- Rejilla de Ocupación
- Oc-Tree

Ambas estructuras deben:

- Almacenar el número de puntos que caen en cada celda o nodo.
- Calcular la media de las coordenadas (x,y,z) de los puntos que caen en dichas regiones.

El fin último es comparar cuántas celdas o nodos se generan, cuántos quedan vacíos u ocupados y la media de puntos en las celdas/nodos ocupados, para entender mejor las ventajas y desventajas de cada método.

---

## 2. Consideraciones y datos de entrada

Para la realización de la práctica, contamos con varios **ficheros .pcd** que contienen nubes de puntos 3D en un formato ASCII. Cada uno de estos ficheros incluye:

- Una cabecera con información:
  - FIELDS (x y z r g b)
  - TYPE (F F F I I I)
  - WIDTH, HEIGHT y POINTS (indican la cantidad total de puntos)
- Una sección DATA ascii donde se listan los puntos en formato:  
“x y z r g b”

en esta practica solo usaremos las tres primeras columnas (x,y,z)

---

## 3. Métodos implementados

### Rejilla de Ocupación

La Rejilla de Ocupación realiza una discretización regular del espacio.

- Se define un tamaño de celda, por ejemplo `cell_size = 1.0`
- Se calcula un offset con los mínimos ( $x_{\min}$ ,  $y_{\min}$ ,  $z_{\min}$ ) de la nube de puntos.
- Para cada punto  $(x,y,z)$ , se mapea a índices de celda  $(ix,iy,iz)$ .
- Se almacena la suma de coordenadas y el conteo total de puntos en cada índice.

De esta forma, cada celda tendrá:

- **count**: Número de puntos en la celda.
- **mean**: Media de las coordenadas  $(x,y,z)$ .

---

### Oc-Tree

La estructura Oc-Tree subdivide recursivamente el espacio en 8 octantes siempre que:

- El número de puntos en un nodo supere un umbral `max_points`, y
- El tamaño del nodo (cubo) sea mayor que un tamaño mínimo de celda (`min_cell_size`).

Cada nodo de tipo `OctreeNode` contiene:

- `min_coord` y `max_coord`, que definen los límites espaciales de su subcubo.
- Un listado de puntos (en nodos no subdivididos o durante la construcción).
- `count` y `mean` (definidos para nodos hoja).
- Una lista de 8 hijos (para los octantes), si se subdivide.

El criterio para detener la subdivisión (marcar nodo como hoja) es cuando el número de puntos no supera `max_points` o el tamaño de la región es menor o igual a `min_cell_size`.

---

## 4. Lógica común y detalles de implementación

### Lectura de ficheros PCD

En ambos métodos, el primer paso es leer las coordenadas 3D de los ficheros .pcd de la siguiente forma:

1. Abrir el fichero.
  2. Avanzar línea a línea hasta encontrar DATA ascii.
  3. Leer línea a línea cada punto; parsear únicamente (x,y,z).
  4. Almacenar estos datos en una lista.
- 

### Estructura de Almacenamiento de Datos

- **Rejilla de Ocupación**

#### Paso 1: Identificar los rangos

Antes de comenzar a asignar puntos a las celdas de la rejilla, es necesario definir bounding box que contiene todos los puntos de la nube. Esto se hace buscando los valores mínimos y máximos de cada eje (x, y, z):

- ( $x_{\min}$ ,  $x_{\max}$ ) Los límites inferior y superior del eje (x).
- ( $y_{\min}$ ,  $y_{\max}$ ) Los límites inferior y superior del eje (y).
- ( $z_{\min}$ ,  $z_{\max}$ ) Los límites inferior y superior del eje (z).

Esto define el volumen tridimensional en el que trabajará la rejilla.

Si por ejemplo tenemos los puntos:

(-5, -7, -3); (4, 2, 6); (-2, -3, 1)

Los límites serían:

$x_{\min} = -5$ ;  $x_{\max} = 4$ ;  $y_{\min} = -7$ ;  $y_{\max} = 2$ ;  $z_{\min} = -3$ ;  $z_{\max} = 6$

## Paso 2: Discretizar cada punto

Para asignar cada punto a una celda de la rejilla, necesitamos discretizar sus coordenadas continuas ( $x, y, z$ ) en índices enteros ( $i_x, i_y, i_z$ ) que identifican una celda específica y esto se hace con la fórmula:

$$i_x = \left\lfloor \frac{x - x_{\min}}{\text{cell\_size}} \right\rfloor, \quad i_y = \left\lfloor \frac{y - y_{\min}}{\text{cell\_size}} \right\rfloor, \quad i_z = \left\lfloor \frac{z - z_{\min}}{\text{cell\_size}} \right\rfloor$$

- Resta de desplazamiento: ( $x - x_{\min}$ ): Se traslada la coordenada al origen relativo de la rejilla.
- División por el tamaño de celda: Divide la distancia al origen relativo entre el tamaño de la celda ( $\text{cell\_size}$ ), obteniendo en qué parte de la celda cae el punto.
- Truncamiento: Redondea hacia abajo para asignar un índice entero a la celda correspondiente.

Ejemplo:

Con un ( $\text{cell\_size} = 2.0$ ), ( $x_{\min} = -5$ ), y el punto  $(-3, -6, 0)$ :

$$\begin{aligned} i_x &= \left\lfloor \frac{-3 - (-5)}{2.0} \right\rfloor = \left\lfloor \frac{2}{2.0} \right\rfloor = \lfloor 1 \rfloor = 1 \\ i_y &= \left\lfloor \frac{-6 - (-7)}{2.0} \right\rfloor = \left\lfloor \frac{1}{2.0} \right\rfloor = \lfloor 0.5 \rfloor = 0 \\ i_z &= \left\lfloor \frac{0 - (-3)}{2.0} \right\rfloor = \left\lfloor \frac{3}{2.0} \right\rfloor = \lfloor 1.5 \rfloor = 1 \end{aligned}$$

El punto se asigna a la celda con índices ( $i_x, i_y, i_z$ ) = (1, 0, 1)

## Paso 3: Acumular datos en la celda

Para cada celda ( $i_x, i_y, i_z$ ), mantenemos dos valores:

count: Número de puntos que caen en la celda.

sum: La suma acumulada de las coordenadas ( $x, y, z$ ) de los puntos.

## Paso 4: Calcular la media final

Para cada celda, calculamos la media de las coordenadas de los puntos almacenados en ella:

$$\text{mean}_{\text{celda}} = \frac{\text{sum}}{\text{count}}$$

Esto nos da la posición promedio de los puntos dentro de esa celda.

## • Oc-Tree

### Paso 1: Nodo raíz

El Oc-Tree comienza con un nodo raíz que abarca todo el espacio tridimensional definido por:

$\text{min\_coord} = [x_{\min}, y_{\min}, z_{\min}]$

$\text{max\_coord} = [x_{\max}, y_{\max}, z_{\max}]$

Este nodo contiene inicialmente todos los puntos de la nube.

### Paso 2: Subdivisión del nodo

Si el nodo contiene más puntos de los permitidos ( $\text{max\_points}$ ) y su tamaño es mayor al mínimo tamaño de celda permitido ( $\text{min\_cell\_size}$ ), se subdivide en 8 subnodos (octantes).

Cada subnodo corresponde a una subdivisión del espacio en 8 regiones iguales (octantes), definidas por dividir los límites en la mitad:

- El punto medio del nodo para cada uno de los ejes se calcula usando la siguiente fórmula:

- $\text{mid\_x} = (x_{\min} + x_{\max}) / 2$

- $\text{mid\_y} = (y_{\min} + y_{\max}) / 2$

- $\text{mid\_z} = (z_{\min} + z_{\max}) / 2$

- Los octantes tienen los límites:

- **Octante 1:**  $[x_{\min}, \text{mid\_x}] \times [y_{\min}, \text{mid\_y}] \times [z_{\min}, \text{mid\_z}]$

- **Octante 2:** Partiendo del Octante 1, vamos cambiando los productos de las multiplicaciones para completar las 8 regiones. Por ejemplo, otro octante podría ser:

$[\text{mid\_x}, x_{\max}] \times [y_{\min}, \text{mid\_y}] \times [z_{\min}, \text{mid\_z}]$

En este caso, hemos tomado la parte del espacio que va de la división del medio hasta el mínimo en los ejes 'y' y 'z', pero del medio al máximo en el eje 'x'. Esto se hace para completar todas las regiones dividiendo cada eje en dos partes. Finalmente, combinando las divisiones de cada eje, se obtienen las  $2^3$  regiones tridimensionales es decir los 8 octantes.

### Paso 3: Asignar puntos a los octantes

Para cada punto del nodo original, verificamos en qué octante cae. Esto se hace comparando las coordenadas del punto con los límites de los octantes.

Ejemplo:

Supongamos que tenemos un punto  $p = (x_p, y_p, z_p)$ . Lo asignamos a un octante verificando:

- Si  $x_{\min} \leq x_p < \text{mid}_x$ : Pertenece a la parte izquierda del eje x.
- Si  $y_{\min} \leq y_p < \text{mid}_y$ : Pertenece a la parte inferior del eje y.
- Si  $z_{\min} \leq z_p < \text{mid}_z$ : Pertenece a la parte frontal del eje z.

Repitiendo esta lógica, encontramos exactamente en qué octante debe estar el punto.

Este proceso es recursivo: cada subnodo puede subdividirse nuevamente en 8 octantes nuevos si cumple las condiciones de subdivisión anteriormente mencionadas, si estos pasara el subnodo entraría al paso 2 nuevamente, en caso contrario pasaría al paso 4

### Paso 4: Nodos hoja

Cuando un nodo ya no se puede subdividir por tener menos de `max_points` o alcanzar `min_cell_size`, se convierte en un nodo hoja.

En los nodos hoja:

- Se almacena el número de puntos (count).
- Se calcula la media de las coordenadas de los puntos almacenados (mean).

### • Diferencias clave

Característica	Rejilla de Ocupación	Oc-Tree
Resolución	Fija	Adaptativa
Estructura de datos	Matriz o diccionario	Árbol jerárquico
Eficiencia espacial	Consume más espacio para zonas vacías	Más eficiente (menos nodos en zonas vacías)
Complejidad	Simplicidad de implementación	Más compleja (recursión)



## 5. Comparación y cálculo de estadísticas

En ambos métodos se implementa una función que recorre la estructura y obtiene las siguientes estadísticas

### Rejilla:

- `total_celdas`: número de claves del diccionario
- `ocupadas`: cuántas celdas tienen `count > 0`.
- `vacias`: celdas que no llegaron a usarse (si se instancia todo el volumen) o 0 si solo se crea celda cuando hay puntos.
- `media_puntos_ocupadas`: la media del `count` en las celdas ocupadas.

### Oc-tree:

- `total_nodos`: número total de nodos en el árbol.
- `hojas`: cantidad de nodos sin hijos (`is_leaf=True`).
- `internas`: cantidad de nodos con hijos (`is_leaf=False`).
- `ocupadas`: hojas con `count > 0`.
- `vacias`: hojas con `count = 0`.
- `media_puntos_ocupadas`: media del `count` en hojas ocupadas.

Debemos de evaluar estos datos usando diferentes valores para los parametros de ***cell\_size*** para la rejilla de ocupación y de ***min\_cell\_octree*** y ***max\_points*** para el Oc-Tree. Una vez obtenidos los resultados de todos los archivos .pcd con distintos valores para los parametros podemos obtener las siguientes observaciones y conclusiones.

### Resultados

- 1) Con (`cell_size=1.0`, `min_cell_octree=1.0`, `max_points=100`)
- 2) Con (`cell_size=0.5`, `min_cell_octree=0.5`, `max_points=100`)
- 3) Con (`cell_size=2.0`, `min_cell_octree=1.0`, `max_points=200`)

1)

Analizando: ciencias000 Análisis Comparativo
-----
Archivo: ./Datos/ciencias000.pcd Rejilla de Ocupacion: Total de Celdas: 79120 Celdas Ocupadas: 2477 Celdas Vacías: 76643 Media de puntos en celdas ocupadas: 23.75 Oc-Tree: Total de Nodos: 2521 Hojas: 2206 Nodos Internos: 315 Celdas (Hojas) Ocupadas: 1223 Celdas (Hojas) Vacías: 983 Media de puntos en hojas ocupadas: 48.10
-----
Analizando: ciencias001 Análisis Comparativo
-----
Archivo: ./Datos/ciencias001.pcd Rejilla de Ocupacion: Total de Celdas: 97524 Celdas Ocupadas: 2361 Celdas Vacías: 95163 Media de puntos en celdas ocupadas: 24.52 Oc-Tree: Total de Nodos: 2433 Hojas: 2129 Nodos Internos: 304 Celdas (Hojas) Ocupadas: 1128 Celdas (Hojas) Vacías: 1001 Media de puntos en hojas ocupadas: 51.31
-----
Analizando: museo000 Análisis Comparativo
-----
Archivo: ./Datos/museo000.pcd Rejilla de Ocupacion: Total de Celdas: 3360 Celdas Ocupadas: 338 Celdas Vacías: 3022 Media de puntos en celdas ocupadas: 234.17 Oc-Tree: Total de Nodos: 2793 Hojas: 2444 Nodos Internos: 349 Celdas (Hojas) Ocupadas: 1212 Celdas (Hojas) Vacías: 1232 Media de puntos en hojas ocupadas: 65.29
-----
Analizando: poli000 Análisis Comparativo
-----
Archivo: ./Datos/poli000.pcd Rejilla de Ocupacion: Total de Celdas: 45630 Celdas Ocupadas: 1625 Celdas Vacías: 44005 Media de puntos en celdas ocupadas: 38.66 Oc-Tree: Total de Nodos: 2777 Hojas: 2430 Nodos Internos: 347 Celdas (Hojas) Ocupadas: 1437 Celdas (Hojas) Vacías: 993 Media de puntos en hojas ocupadas: 43.71
-----
Analizando: poli001 Análisis Comparativo
-----
Archivo: ./Datos/poli001.pcd Rejilla de Ocupacion: Total de Celdas: 48960 Celdas Ocupadas: 1654 Celdas Vacías: 47306 Media de puntos en celdas ocupadas: 37.14 Oc-Tree: Total de Nodos: 2473 Hojas: 2164 Nodos Internos: 309 Celdas (Hojas) Ocupadas: 1373 Celdas (Hojas) Vacías: 791 Media de puntos en hojas ocupadas: 44.73
-----

2)

Analizando: ciencias000 Análisis Comparativo
-----
Archivo: ./Datos/ciencias000.pcd Rejilla de Ocupacion: Total de Celdas: 617136 Celdas Ocupadas: 6563 Celdas Vacías: 610573 Media de puntos en celdas ocupadas: 8.96 Oc-Tree: Total de Nodos: 2945 Hojas: 2577 Nodos Internos: 368 Celdas (Hojas) Ocupadas: 1413 Celdas (Hojas) Vacías: 1164 Media de puntos en hojas ocupadas: 41.63
-----
Analizando: ciencias001 Análisis Comparativo
-----
Archivo: ./Datos/ciencias001.pcd Rejilla de Ocupacion: Total de Celdas: 768060 Celdas Ocupadas: 6436 Celdas Vacías: 761624 Media de puntos en celdas ocupadas: 9.00 Oc-Tree: Total de Nodos: 2889 Hojas: 2528 Nodos Internos: 361 Celdas (Hojas) Ocupadas: 1365 Celdas (Hojas) Vacías: 1163 Media de puntos en hojas ocupadas: 42.40
-----
Analizando: museo000 Análisis Comparativo
-----
Archivo: ./Datos/museo000.pcd Rejilla de Ocupacion: Total de Celdas: 23465 Celdas Ocupadas: 1192 Celdas Vacías: 22273 Media de puntos en celdas ocupadas: 66.40 Oc-Tree: Total de Nodos: 4033 Hojas: 3529 Nodos Internos: 504 Celdas (Hojas) Ocupadas: 1735 Celdas (Hojas) Vacías: 1794 Media de puntos en hojas ocupadas: 45.61
-----
Analizando: poli000 Análisis Comparativo
-----
Archivo: ./Datos/poli000.pcd Rejilla de Ocupacion: Total de Celdas: 351000 Celdas Ocupadas: 4898 Celdas Vacías: 346102 Media de puntos en celdas ocupadas: 12.83 Oc-Tree: Total de Nodos: 3249 Hojas: 2843 Nodos Internos: 406 Celdas (Hojas) Ocupadas: 1684 Celdas (Hojas) Vacías: 1159 Media de puntos en hojas ocupadas: 37.30
-----
Analizando: poli001 Análisis Comparativo
-----
Archivo: ./Datos/poli001.pcd Rejilla de Ocupacion: Total de Celdas: 371680 Celdas Ocupadas: 4775 Celdas Vacías: 366905 Media de puntos en celdas ocupadas: 12.86 Oc-Tree: Total de Nodos: 2937 Hojas: 2570 Nodos Internos: 367 Celdas (Hojas) Ocupadas: 1560 Celdas (Hojas) Vacías: 1010 Media de puntos en hojas ocupadas: 39.37
-----

3)

Analizando: ciencias000 Análisis Comparativo
-----
Archivo: ./Datos/ciencias000.pcd Rejilla de Ocupacion: Total de Celdas: 10120 Celdas Ocupadas: 851 Celdas Vacías: 9269 Media de puntos en celdas ocupadas: 69.13 Oc-Tree: Total de Nodos: 1345 Hojas: 1177 Nodos Internos: 168 Celdas (Hojas) Ocupadas: 668 Celdas (Hojas) Vacías: 509 Media de puntos en hojas ocupadas: 88.06
-----
Analizando: ciencias001 Análisis Comparativo
-----
Archivo: ./Datos/ciencias001.pcd Rejilla de Ocupacion: Total de Celdas: 12474 Celdas Ocupadas: 836 Celdas Vacías: 11638 Media de puntos en celdas ocupadas: 69.25 Oc-Tree: Total de Nodos: 1393 Hojas: 1219 Nodos Internos: 174 Celdas (Hojas) Ocupadas: 649 Celdas (Hojas) Vacías: 570 Media de puntos en hojas ocupadas: 89.19
-----
Analizando: museo000 Análisis Comparativo
-----
Archivo: ./Datos/museo000.pcd Rejilla de Ocupacion: Total de Celdas: 480 Celdas Ocupadas: 87 Celdas Vacías: 393 Media de puntos en celdas ocupadas: 909.76 Oc-Tree: Total de Nodos: 1841 Hojas: 1611 Nodos Internos: 230 Celdas (Hojas) Ocupadas: 809 Celdas (Hojas) Vacías: 802 Media de puntos en hojas ocupadas: 97.81
-----
Analizando: poli000 Análisis Comparativo
-----
Archivo: ./Datos/poli000.pcd Rejilla de Ocupacion: Total de Celdas: 6237 Celdas Ocupadas: 503 Celdas Vacías: 5734 Media de puntos en celdas ocupadas: 124.89 Oc-Tree: Total de Nodos: 1553 Hojas: 1359 Nodos Internos: 194 Celdas (Hojas) Ocupadas: 800 Celdas (Hojas) Vacías: 559 Media de puntos en hojas ocupadas: 78.52
-----
Analizando: poli001 Análisis Comparativo
-----
Archivo: ./Datos/poli001.pcd Rejilla de Ocupacion: Total de Celdas: 6240 Celdas Ocupadas: 522 Celdas Vacías: 5718 Media de puntos en celdas ocupadas: 117.68 Oc-Tree: Total de Nodos: 1257 Hojas: 1100 Nodos Internos: 157 Celdas (Hojas) Ocupadas: 738 Celdas (Hojas) Vacías: 362 Media de puntos en hojas ocupadas: 83.22
-----

## Observaciones

A partir de los resultados obtenidos con las configuraciones de parámetros:

### 1) (cell\_size=1.0, min\_cell\_octree=1.0, max\_points=100):

- La rejilla de ocupación genera una cantidad considerable de celdas vacías (entre 76,000 y 95,000) esto es debido a que el espacio tridimensional es dividido en celdas uniformes y solo una pequeña proporción de estas celdas contiene puntos.
- En el Oc-Tree, la cantidad de hojas ocupadas es menor que las celdas ocupadas en la rejilla. Esto refleja la ventaja del Oc-Tree para adaptarse a regiones más densas y evitar subdivisiones innecesarias en áreas vacías.
- La media de puntos en las hojas ocupadas del Oc-Tree (entre 41 y 65 puntos) es mayor que en la rejilla, lo que indica una mejor agrupación de puntos en las celdas del árbol.

### 2) (cell\_size=0.5, min\_cell\_octree=0.5, max\_points=100):

- Reducir el tamaño de las celdas de la rejilla incrementa drásticamente el número total de celdas (hasta 768,000). Aunque esto mejora la resolución de la rejilla, también aumenta el número de celdas vacías.
- En el Oc-Tree, se observa un aumento en el número total de nodos y hojas debido a la menor tolerancia de tamaño de celda. Sin embargo, el Oc-Tree sigue siendo más eficiente, manteniendo menos hojas vacías comparadas con la rejilla.
- La media de puntos en hojas ocupadas en el Oc-Tree (entre 37 y 45 puntos) muestra una agrupación más compacta en comparación con la rejilla.

### 3) (cell\_size=2.0, min\_cell\_octree=1.0, max\_points=200):

- Aumentar el tamaño de celda de la rejilla reduce significativamente el número total de celdas (alrededor de 10,000 a 12,000), lo que disminuye la cantidad de celdas vacías. Esto demuestra que un tamaño de celda mayor es útil en entornos menos densos.
- En el Oc-Tree, la cantidad de nodos y hojas disminuye en comparación con configuraciones de menor tamaño de celda, reflejando un menor número de subdivisiones.
- La media de puntos por celda ocupada en el Oc-Tree (entre 78 y 97 puntos) es mayor que en las otras configuraciones, lo que indica una mejor eficiencia en el agrupamiento de puntos en áreas densas.

## Comparación General

### Rejilla de Ocupación

- Es simple de implementar y adecuada para entornos donde se requiere una división uniforme del espacio.
- Sufre de ineficiencia en regiones con muchos puntos vacíos, especialmente con tamaños de celda pequeños.
- Aumentar el tamaño de celda mejora la eficiencia, pero reduce la resolución espacial.

### Oc-Tree

- Es más eficiente para representar datos en 3D, adaptándose a las regiones densas y dejando sin dividir las áreas vacías.
- Produce menos nodos totales en comparación con las celdas de la rejilla.
- Permite un balance entre resolución y eficiencia al ajustar los parámetros `min_cell_octree` y `max_points`.

## Conclusión Final

El Oc-Tree es claramente superior a la rejilla de ocupación en términos de eficiencia y adaptabilidad para datos tridimensionales dispersos. Mientras que la rejilla de ocupación es útil para representaciones uniformes, el Oc-Tree ofrece una solución más compacta y ajustada a la distribución de puntos, lo que lo hace más adecuado para aplicaciones que requieren almacenamiento eficiente y resolución dinámica.

## 6. Parte Optativa: Visualización 3D con PyVista

Consiste en la visualización 3D de los datos y la estructura del Oc-Tree , utilizando la librería PyVista. El visualizador permite observar tanto los puntos originales de la nube como las celdas generadas por el Oc-Tree.

### Implementación:

- Se utiliza la librería PyVista para renderizar la nube.
- Cada fichero .pcd se lee completamente y se construye un objeto PolyData que contiene todos los puntos.
- Se crea un plot con el objeto, permite rotar, hacer zoom y desplazarse por el espacio 3D
- En la clase Oc-tree se ha implementado un metodo que consiste en guardar una lista de todos los nodos que genera el Oc-Tree subdividiendose, para exportarlos y representarlos en formato de cubos en el plot junto a los puntos de los archivos .pcd correspondiente

PyVista  
archivo: ciencias000, puntos:58826, min\_cell\_size:1, max\_points:100

