

## Práctica 04

DOCENTE	CARRERA	CURSO
Vicente Machaca Arceda	Maestría en Ciencias de la Computación	Algoritmos y Estructura de Datos

PRÁCTICA	TEMA	DURACIÓN
04	Kd-tree	—

### 1. Integrantes

- Grupo N° 2
- Integrantes:
  - EDER ALONSO AMPUERO ATAMARI
  - HOWARD FERNANDO ARANZAMENDI MORALES
  - JOSE EDISON PEREZ MAMANI
  - HENRRY IVAN ARIAS MAMANI

### 2. Repositorio GitHub

URL Github: Repositorio Práctica 4 AyED

### 3. Marco Teórico

#### 3.1. Kd-Tree

#### 3.2. KNN

### 4. Ejercicios

#### 4.1. Ejercicio N

##### 4.1.1. Definición

Un árbol K-D (también llamado árbol K-dimensional) es un árbol de búsqueda binaria donde los datos en cada nodo son un punto K-dimensional en el espacio. En resumen, es una estructura de datos de partición de espacio (detalles a continuación) para organizar puntos en un espacio K-Dimensional.

Un nodo que no es una hoja en el árbol K-D divide el espacio en dos partes, llamadas medios espacios.

Los puntos a la izquierda de este espacio están representados por el subárbol izquierdo de ese nodo y los puntos a la derecha del espacio están representados por el subárbol derecho. Pronto estaremos explicando el concepto de cómo se divide el espacio y se forma el árbol.

En aras de la simplicidad, entendamos un árbol 2-D con un ejemplo.

La raíz tendría un plano alineado con el eje x, los hijos de la raíz tendrían ambos planos alineados con el eje y, los nietos de la raíz tendrían todos planos alineados con el eje x, y los bisnietos de la raíz tendrían todos planos alineados con el eje y, y así sucesivamente.

#### 4.1.2. Generalización

Numeremos los planos como 0, 1, 2, ... (K - 1). Del ejemplo anterior, es bastante claro que un punto (nodo) en la profundidad D tendrá un plano alineado donde A se calcula como:

$$A = D \bmod K$$

#### 4.1.3. ¿Cómo determinar si un punto estará en el subárbol izquierdo o en el subárbol derecho?

Si el nodo raíz está alineado en el plano A, el subárbol izquierdo contendrá todos los puntos cuyas coordenadas en ese plano sean más pequeñas que las del nodo raíz. De manera similar, el subárbol derecho contendrá todos los puntos cuyas coordenadas en ese plano sean mayores-iguales que las del nodo raíz.

#### 4.1.4. Creación de un árbol 2-D

Considere los siguientes puntos en un plano 2-D: (3, 6), (17, 15), (13, 15), (6, 12), (9, 1), (2, 7), (10, 19)

1. Insertar (3, 6): dado que el árbol está vacío, conviértalo en el nodo raíz.
2. Insertar (17, 15): compararlo con el punto del nodo raíz. Dado que el nodo raíz está alineado con X, el valor de la coordenada X se comparará para determinar si se encuentra en el subárbol derecho o en el subárbol izquierdo. Este punto estará alineado con Y.
3. Insertar (13, 15): el valor X de este punto es mayor que el valor X del punto en el nodo raíz. Entonces, esto estará en el subárbol derecho de (3, 6). Nuevamente compare el valor Y de este punto con el valor Y del punto (17, 15) (¿Por qué?). Como son iguales, este punto estará en el subárbol derecho de (17, 15). Este punto estará alineado con X.
4. Insertar (6, 12): el valor X de este punto es mayor que el valor X del punto en el nodo raíz. Entonces, esto estará en el subárbol derecho de (3, 6). Nuevamente compare el valor Y de este punto con el valor Y del punto (17, 15) (¿Por qué?). Como 12 < 15, este punto estará en el subárbol izquierdo de (17, 15). Este punto estará alineado con X.
5. Insertar (9, 1): De manera similar, este punto estará a la derecha de (6, 12).
6. Insertar (2, 7): De manera similar, este punto estará a la izquierda de (3, 6).
7. Inserta (10, 19): De manera similar, este punto estará a la izquierda de (13, 15).

#### Función Insert:

Las funciones de inserción se utilizan para insertar un nodo en un Quad Tree existente. Esta función primero verifica si el nodo dado está dentro de los límites del quad actual. Si no es así, detenemos inmediatamente la inserción. Si está dentro de los límites, seleccionamos el elemento secundario apropiado para contener este nodo en función de su ubicación. Esta función es  $O(\log N)$  donde N es el tamaño de la distancia.

#### Función Search:

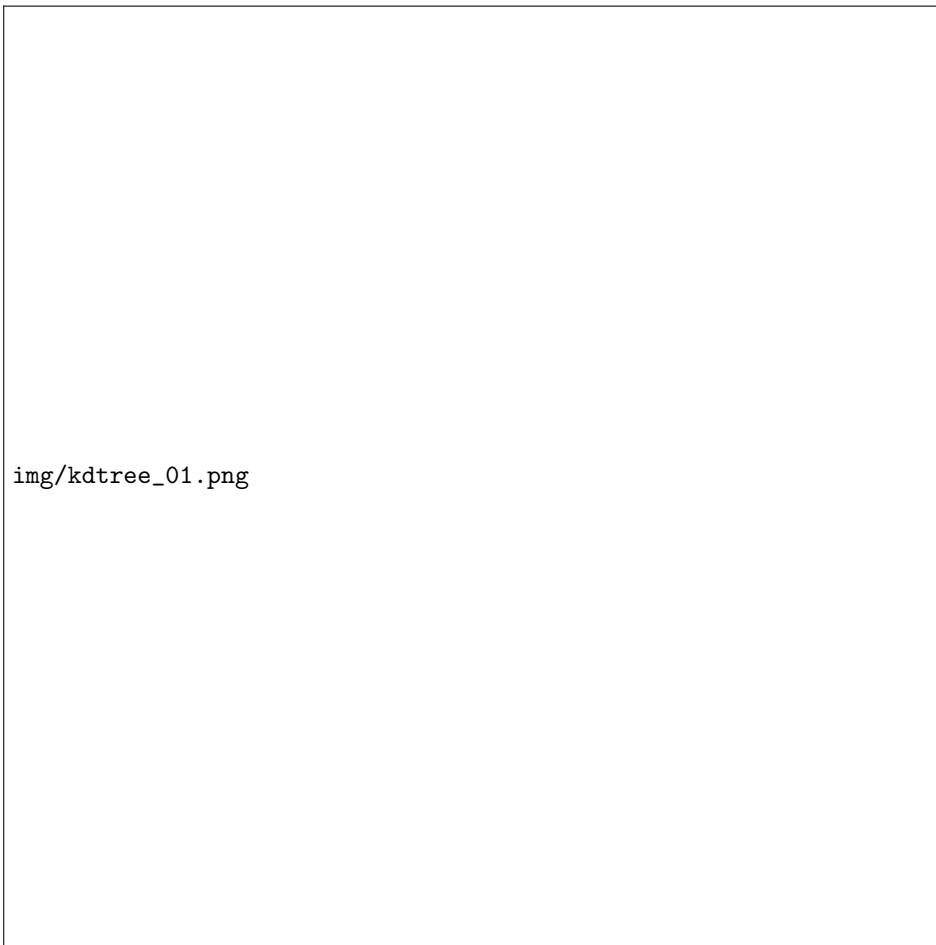


Figura 1: Nodos en KD-TREE

La función de búsqueda se utiliza para localizar un nodo en el cuadrante dado. También se puede modificar para devolver el nodo más cercano al punto dado. Esta función se implementa tomando el punto dado, comparándolo con los límites de los quads secundarios y recursivamente. Esta función es  $O(\log N)$  donde  $N$  es el tamaño de la distancia.

#### 4.1.5. Cuadrantes en Quadtree:

- Se observa en la figura ?? como estan ubicados los cuadrantes y cuales son los nombres asignados a cada uno de ellos.

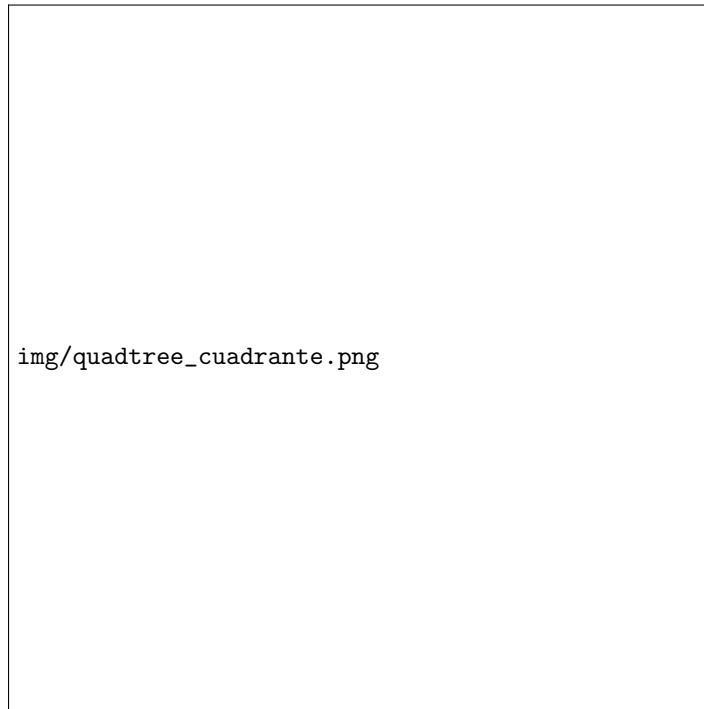


Figura 2: Disposición de los cuadrantes en el Quadtree

#### 4.1.6. Resultados de la practica:

- En la figura ?? se puede observar que a medida que se incrementan puntos  $n > 4$  se va subdividiendo el cuadrante en 4 subcuadrantes.
- En la figura ?? se puede observar la subdivision de los cuadrantes y los puntos contenidos.
- En la figura ?? se puede observar que se van resaltando los puntos dentro del area del cuadro verde, el cual se va intersectando.



Figura 3: Inserción de puntos

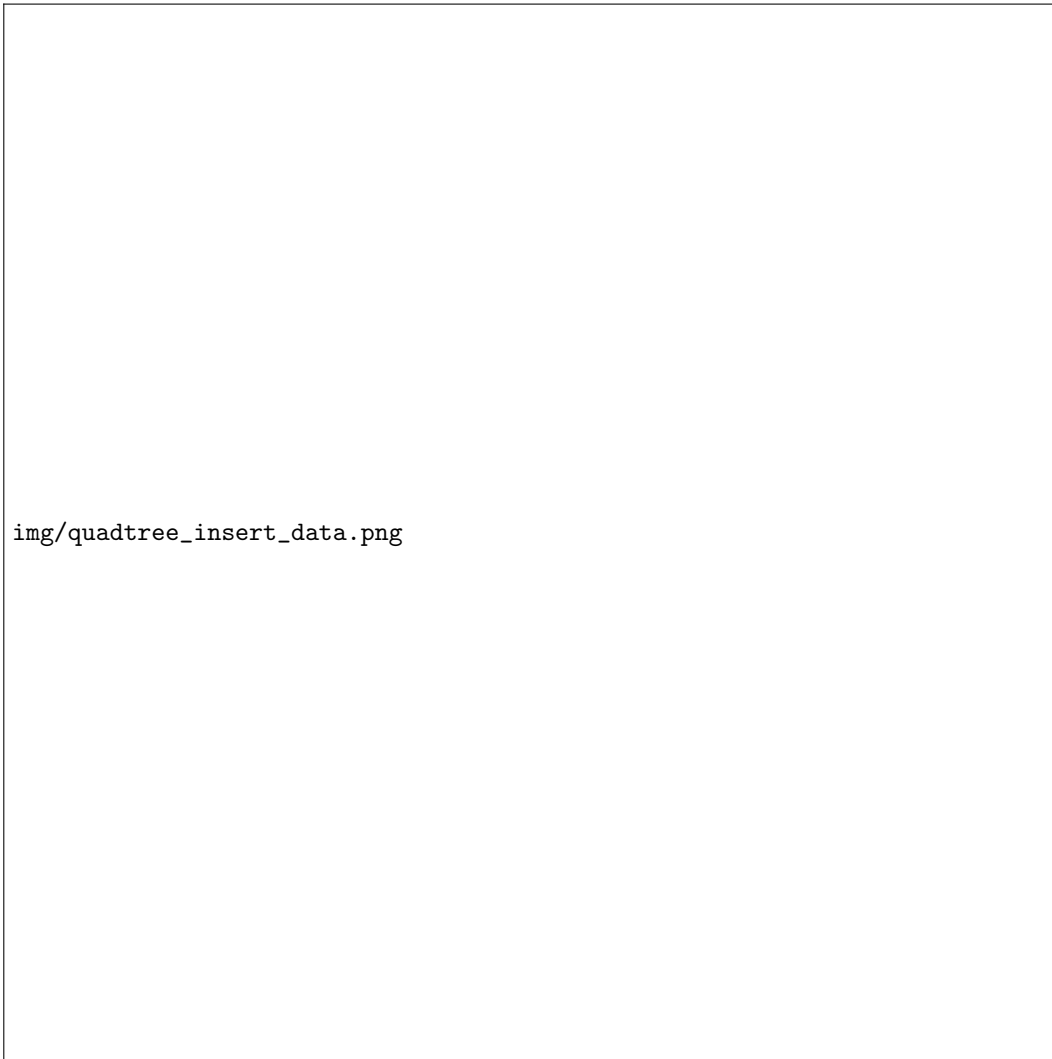


Figura 4: Puntos insertados



Figura 5: Búsqueda de puntos

## 4.2. Octree

El Octree es una estructura de datos de árbol en la que cada nodo interno puede tener como máximo 8 hijos. Al igual que el árbol binario que divide el espacio en dos segmentos, Octtree divide el espacio en ocho partes como máximo, lo que se denomina octanos. Se utiliza para almacenar el punto 3-D que ocupa una gran cantidad de espacio. Si todo el nodo interno del Octárbol contiene exactamente 8 hijos, entonces se llama Octárbol completo. También es útil para gráficos de alta resolución como gráficos de computadora en 3D.

El Octree se puede formar a partir de un volumen 3D siguiendo los siguientes pasos de acuerdo a la figura ??:

1. Divide el espacio 3D actual en ocho cubos.
2. Si algún cubo tiene más de un punto, divídalo más en 8 cubos.
3. No divida el cubo que tiene uno o cero puntos
4. Realice este proceso repetidamente hasta que todo el cuadro contenga uno o cero puntos.

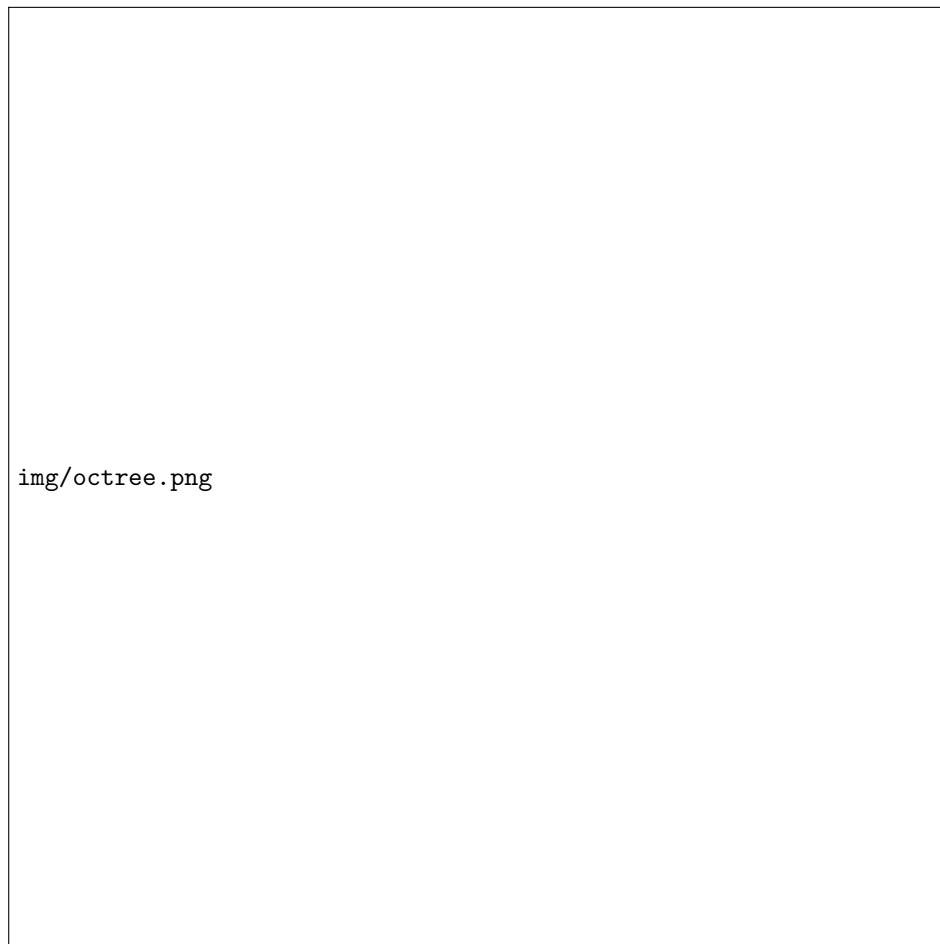


Figura 6: Representación gráfica del Octree

Si  $S$  es el número de puntos en cada dimensión, entonces el número de nodos que se forman en Octtree viene dado por esta fórmula  $(S^3 - 1)/7$ .



**Función Insert:**

- Para insertar un nodo en Octree, en primer lugar, verificamos si existe un nodo o no, si existe un nodo, luego regresamos, de lo contrario, vamos recursivamente.
- Primero, comenzamos con el nodo raíz y lo marcamos como actual, luego encontramos el nodo secundario en el que podemos almacenar el punto.
- Si el nodo está vacío, se reemplaza con el nodo que queremos insertar y convertirlo en un nodo hoja
- Si el nodo es el nodo hoja, conviértalo en un nodo interno y si es un nodo interno, vaya al nodo secundario. Este proceso se realiza de forma recursiva hasta que no se encuentra un nodo vacío.
- La complejidad temporal de esta función es  $O(\log(N))$  donde N es el número de nodos.

**Función Search:**

- Esta función se utiliza para buscar el punto existe es el árbol o no.
- Comience con el nodo raíz y busque recursivamente si se encuentra el nodo con el punto dado y luego devuelva verdadero, si se encuentra un nodo vacío o un punto límite o un punto vacío, devuelva falso.
- Si se encuentra un nodo interno, vaya a ese nodo. La complejidad temporal de esta función también es  $O(\log N)$  donde N es el número de nodos

**4.2.1. Cuadrantes en Octree:**

- Se observa en la figura ?? como se distribuyen los cuadrantes en 3D y cuales son los nombres asignados a cada uno de ellos.

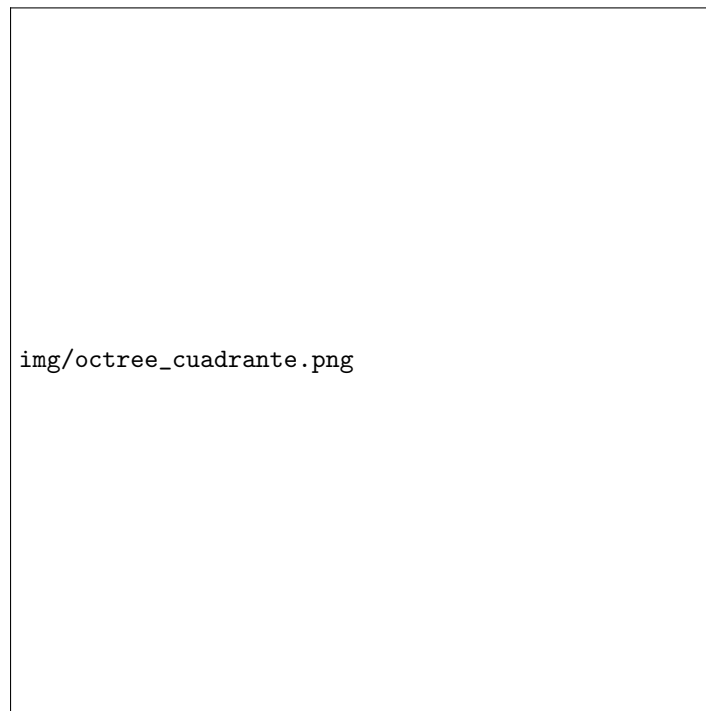


Figura 7: Disposición de los cuadrantes en el Octree

#### 4.2.2. Resultados de la practica:

- En la figura ?? se puede observar que a medida que se incrementan puntos  $n > 2$  se va subdividiendo el cuadrante en 8 subcuadrantes.



Figura 8: Inserción de puntos en octree

- En la figura ?? se puede observar la subdivisión de los cuadrantes y los puntos contenidos en cada cuadrante.



Figura 9: Puntos insertados en octree

## 5. Conclusiones

- El quadtree se codifican en un árbol de cuatro hijos no balanceado, y el octree se codifica en un árbol octal no balanceado.
- De la bibliografía revisada, se concluye que las aplicaciones más comunes para los QuadTree y OcTree son el Procesamiento de imágenes, generación de mallas e indexado espacial.
- Las estructuras octree son usadas mayormente para partir un espacio tridimensional, dividiéndolo recursivamente en ocho octantes, siendo análogas tridimensionales de los quadtree bidimensionales.

- El coste medio de insertar un nuevo nodo en un Octree es de  $O(\log 8n)$ , siendo  $n$  su número de nodos. Asimismo, el coste por buscar una hoja es  $O(\log 8n)$ .

## 6. Referencias

1. Tobler, W., Chen, Z. T. (1986). A quadtree for global information storage. *Geographical Analysis*, 18(4), 360-371.
2. Samet, H. (1984). The quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2), 187-260.
3. Rojas Delgado, J. (2015). Estructura de datos espacial jerárquica para la indexación de agrupaciones de objetos geométricos (Bachelor's thesis, Universidad de las Ciencias Informáticas. Facultad 6).
4. Castillo Oliva, A. D. (2018). Visualización de grandes volúmenes de datos empleando Octree (Bachelor's thesis, Universidad de las Ciencias Informáticas. Centro Vertex, Entornos Interactivos 3D. Facultad 4).
5. Rivero, J. P. S., de la Hoz, Á. P., Medina, M. Á. P. GENERACIÓN DE MALLAS APLICACIONES A LA INGENIERÍA.
6. Brunet Crosa, P., Santistevé i Puyuelo, F., Vilanova, A., Chiarabini, L., Patow, G. A., Staffetti, E., Surinyac, J. (1999). Estructuras geométricas jerárquicas para la modelización de escenas 3D.
7. P5.js
8. Geeksforgeeks