

PRACTICA 4 DE LABORATORIO

DOCENTE	CARRERA	CURSO
Vicente Machaca Arceda	Escuela Profesional de Ciencias de la Computación	Estructuras de Datos Avanzadas

PRÁCTICA N°	TEMA	DURACIÓN (HORAS)
04	Octree - Aplicaciones	10

1. OBJETIVOS

- Comprender las aplicaciones del Octree e *Image quantization*

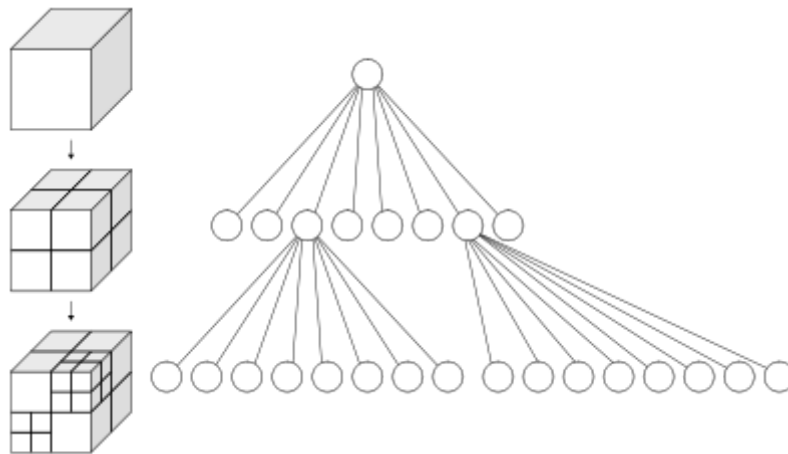
2. TEMAS A TRATAR

- Octree
- *Image Quantization*

3. MARCO TEÓRICO

OCTREE

(Extraído de: <https://esacademic.com/dic.nsf/eswiki/1418673>)



En una grilla octree, cada nodo subdivide el espacio que representa en ocho octantes. En una región punto (PR) octree, el nodo almacena un punto tridimensional explícito, el cual es el "centro" de la subdivisión para ese nodo; el punto que define una de las esquinas para cada uno de los ocho hijos. En una octree MX, el punto de subdivisión es implícitamente el centro del espacio que el nodo representa. El nodo raíz de una PR octree puede representar un espacio infinito; el nodo raíz de una octree MX debe representar un espacio con límite finito para que los centros implícitos estén bien definidos. Las grillas octree nunca se consideran árbol kd, ya que los árboles kd dividen en una dimensión mientras que las grillas octree dividen alrededor de un punto. Los árboles kd además son siempre binarios, lo cual no se cumple para las grillas octree.

COLOR QUANTIZATION

La cuantificación del color reduce el número de colores usados en una imagen; esto es importante para visualizar imágenes en dispositivos que soportan un número limitado de colores y para eficiencia de compresión de ciertos tipos de imágenes. La mayoría de los editores de imagen y muchos sistemas operativos soportan de forma nativa la cuantificación del color. Entre los algoritmos de cuantificación de color modernos más populares se encuentran el algoritmo del color más cercano (para paletas fijas), el algoritmo Median cut, y un algoritmo basado en octrees.

Por ejemplo en la imagen de la izquierda tiene menos colores, mientras que en la imagen de la derecha hay mas colores.

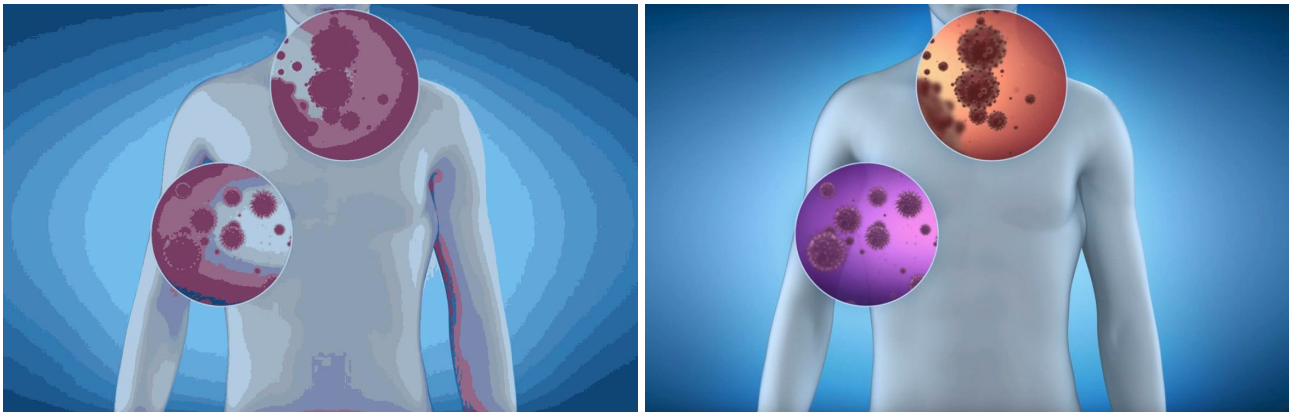


Figura 1: Izquierda: Imagen con menos colores, Derecha: imagen con una mayor cantidad de colores.

Por ejemplo si fuera en escala de grises:



1 bit
2 niveles



2 bit
4 niveles



3 bit
8 niveles



4 bit
16 niveles



5 bit
32 niveles



6 bit
64 niveles

Figura 2: Diferentes niveles de colores en una imagen.

4. EJERCICIOS

Implemente el algoritmo de **Color Quantization** utilizando un Octree en **C++**. Es decir, dada como entrada una imagen:

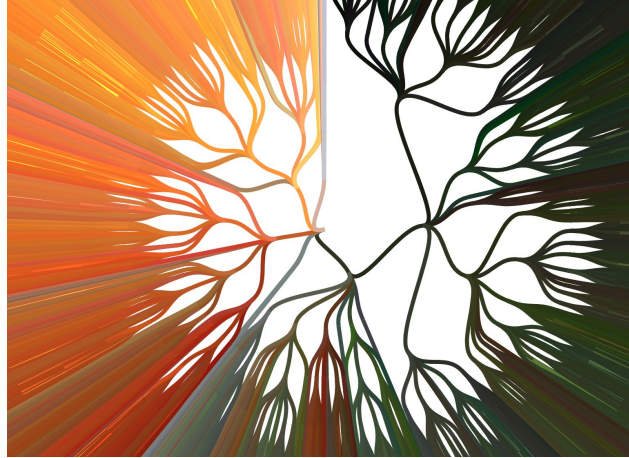


Figura 3: Imagen original.

Usted debe implementar un algoritmo utilizando un octree para reducir la cantidad de colores de dicha imagen, por ejemplo en las Figuras 4 y 5, la cantidad de colores se ha reducido a 256 y 64 respectivamente, además también podemos ver la paleta de colores utilizada. Su algoritmo debe retornar tanto la imagen reducida como la paleta de colores.

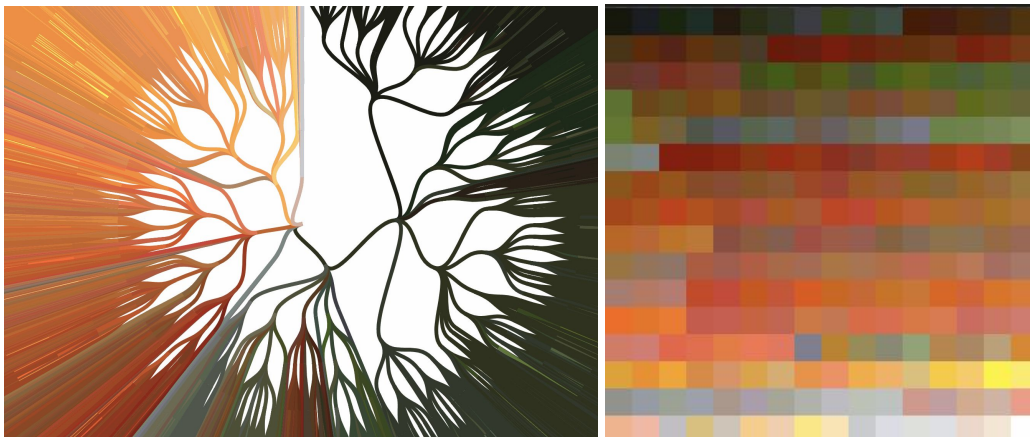


Figura 4: Izquierda: Imagen con solo 256 colores, Derecha: Paleta de colores.



Figura 5: Izquierda: Imagen con solo 64 colores, Derecha: Paleta de colores.

Para revisar el algoritmo puede guiarse de estas implementaciones:

- [Descripción del algoritmo en C.](#)
- [Implementación en Python.](#)
- [Descripción del algoritmo.](#)
- [Implementación en JavaScript](#)

REFERENCIAS

Orchard, M. T., & Bouman, C. A. (1991). Color quantization of images. *IEEE transactions on signal processing*, 39(12), 2677-2690.

Gervautz, M., & Purgathofer, W. (1988). A simple method for color quantization: Octree quantization. In *New trends in computer graphics* (pp. 219-231). Springer, Berlin, Heidelberg.

Bloomberg, D. S. (2008). Color quantization using octrees. *Leptonica*, ss, 1-10.