

# Taller AGM

Algoritmos y Estructuras de Datos III



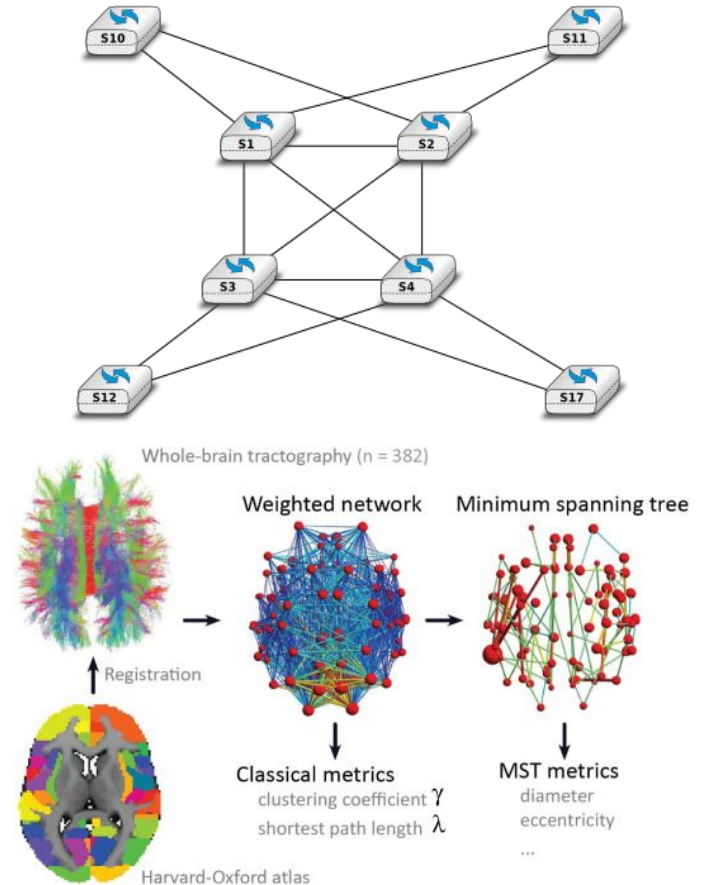
# Menú del día

- Aplicación de AGM
- Código útil para TP

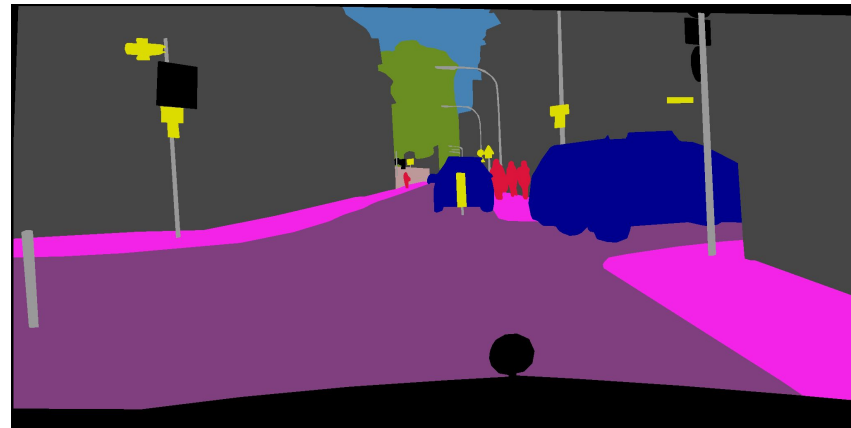
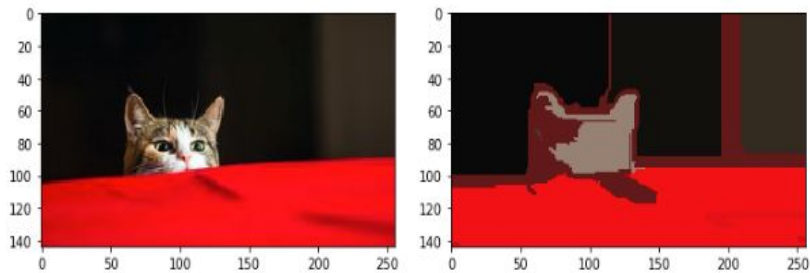
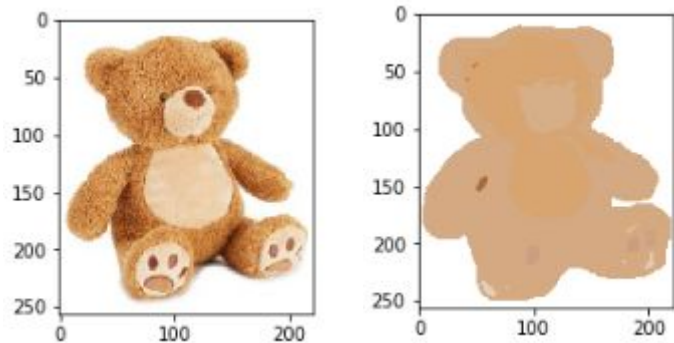


# Aplicaciones

- Network design
- Spanning tree protocol
- Neurociencia
- Clustering
- Segmentación de imágenes
- ...



# Segmentación de imágenes



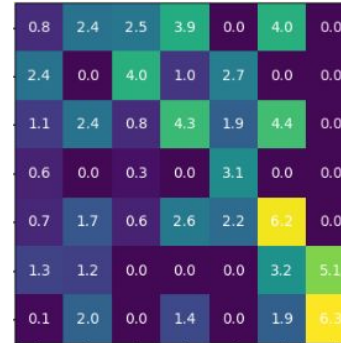
# Algoritmo: Contexto

- Vamos a usar escala de grises
- Las imágenes son matrices de números
- Vamos a ver a la imagen como un grafo
- Cada píxel es un nodo
- Cada nodo tiene un eje a sus 4 vecinos
- Peso es diferencia entre ambos píxeles
- Componentes conexas = Segmentos

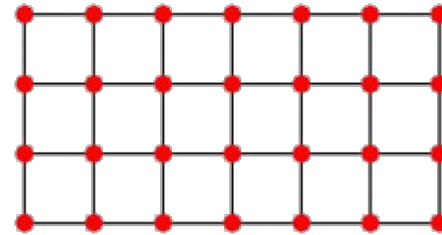
Observación: Cuando dos píxeles son muy distintos es “costoso” pasar de un nodo a otro.



Usamo escala de grises



Las imágenes son matrices de números



Grafo grilla

# Algoritmo

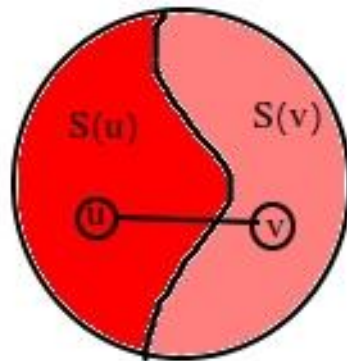
- Ordenar  $E$  de menor a mayor (parecido a distinto)
- Cada vértice está en su propio segmento
- Para  $(u, v, w)$  en  $E$ :
  - Si  $\text{Segmento}(u) == \text{Segmento}(v) \Rightarrow$  ignorar
  - Si  $\text{Similares}(\text{Segmento}(u), \text{Segmento}(v), w) \Rightarrow$  unir

Segmentos “Similares” visto de 3 formas:

1. Diferencia( $u, v$ ) chica vs las diferencias que hay dentro de sus segmentos
2.  $W$  actual sea chico con respecto a los  $w$ 's dentro de cada segmento
3.  $w \leq \min(\max(w_1 \text{ en } \text{Segmento}(u)), \max(w_2 \text{ en } \text{Segmento}(u)))$

En realidad:

4.  $w \leq \min(\max(w_1 \text{ en } S(u)) + k / \text{Size}(S(u)), \max(w_2 \text{ en } S(v)) + k / \text{Size}(S(v)))$



# ¿A qué algoritmo se parece?



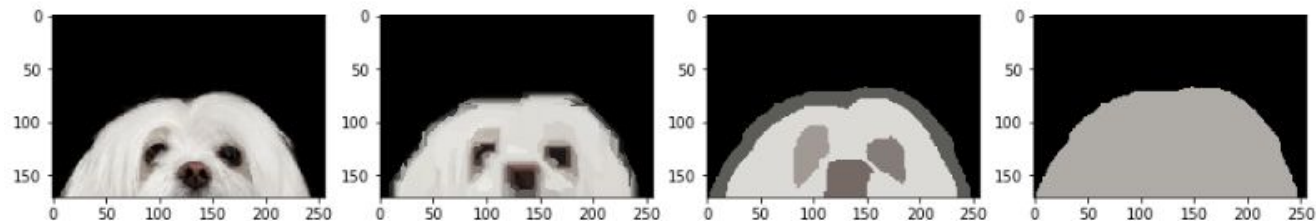
# Taller

Objetivo:

- Codear el algoritmo
- Experimentación y análisis cualitativo
- ¿Qué pasa con  $k$ ? ¿Es igual para todas las imágenes?

Repaso de:

- Observo
- Hipótesis
- Experimento
- Concluyo



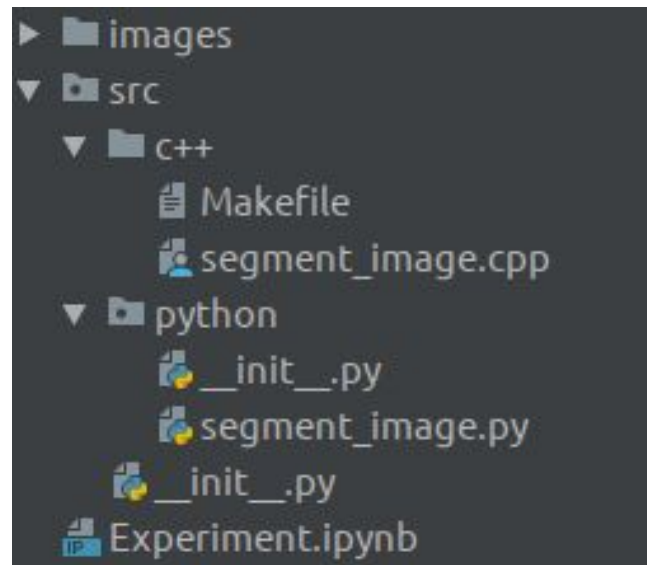


# Taller

Descargar **TallerAGM.zip** del campus

Estructura:

- **images/**: imágenes de prueba 📁
- **src/**:
  - **c++/**: código con algoritmo 🔨
  - **python/**: leer y mostrar imágenes 📁
- **Experiment.ipynb**: Experimentación 🔨



Completar **segment\_image.cpp** y jugar con **Experiment.ipynb**