

Geometría Computacional

Proyecto Final

Problema:

¿Cómo puedo hacer para guardar en poquito espacio y que cuando la abra/expandas no se pierdan sus generalidades de color, teniendo líneas finas, etc.?



Problema:

¿Cómo puedo hacer para guardar en poquito espacio y que cuando la abra/expandi no se pierdan sus generalidades de color, teniendo líneas finas, etc.?



SLIC + Voronoi = Voronoi Superpixels

Segmentación de imágenes en regiones de Voronoi

Introducción

Según el Wiktionary:

Etymology

super- + *pixel*

Noun

superpixel (*plural superpixels*)

1.(computer graphics) A polygonal part of a digital image, larger than a normal pixel, that is rendered in the same colour and brightness.

Descripción y uso

Los *superpixels* determinan regiones de ‘generalización’ del color de una imagen, reduciendo la complejidad del trabajo con las imágenes.

Descripción y uso

Los *superpixels* determinan regiones de ‘generalización’ del color de una imagen, reduciendo la complejidad del trabajo con las imágenes.

Tienen muchas aplicaciones, ya que se pueden usar para estimar profundidades en imágenes, estimaciones de cuerpos, localización de objetos, compresión de imágenes, entre otras.

Idea general

Utilizamos un método de segmentación conocido como SLIC (Simple Linear Iterative Clustering). [Más detalles adelante...]

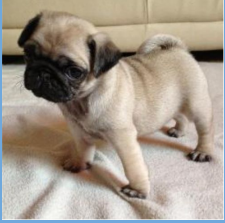
Idea general

Utilizamos un método de segmentación conocido como SLIC (Simple Linear Iterative Clustering). [Más detalles adelante...]

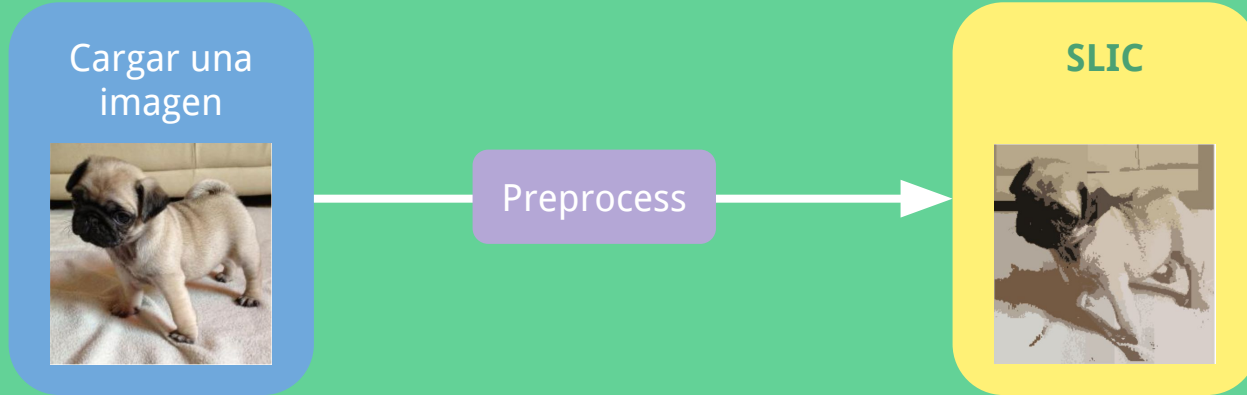
Posteriormente, utilizamos el resultado para calcular regiones de Voronoi con los centros de los clusters obtenidos con el método anterior.

Procedimiento general

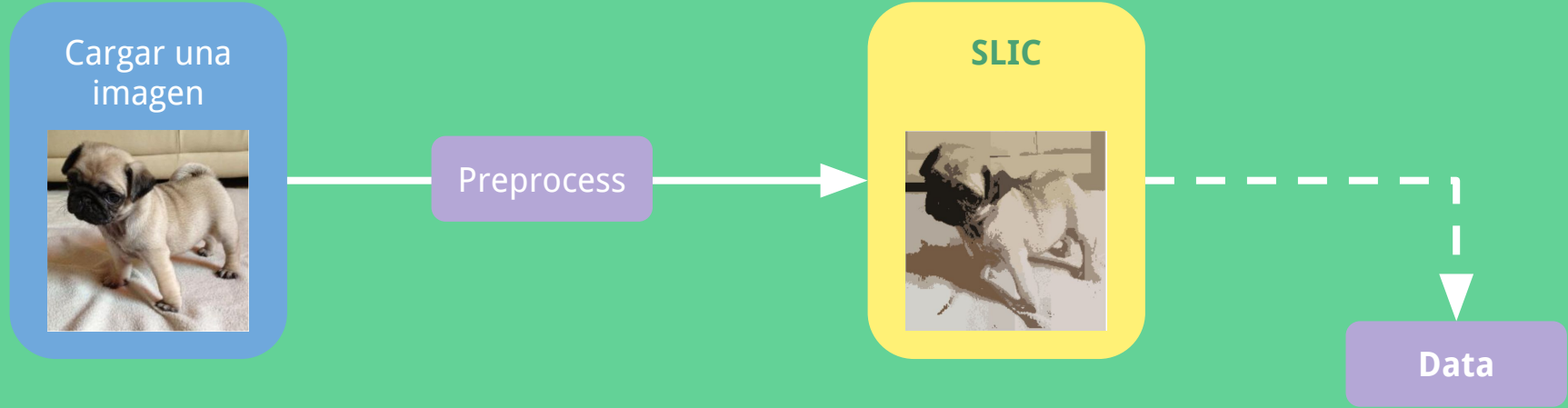
Cargar una
imagen



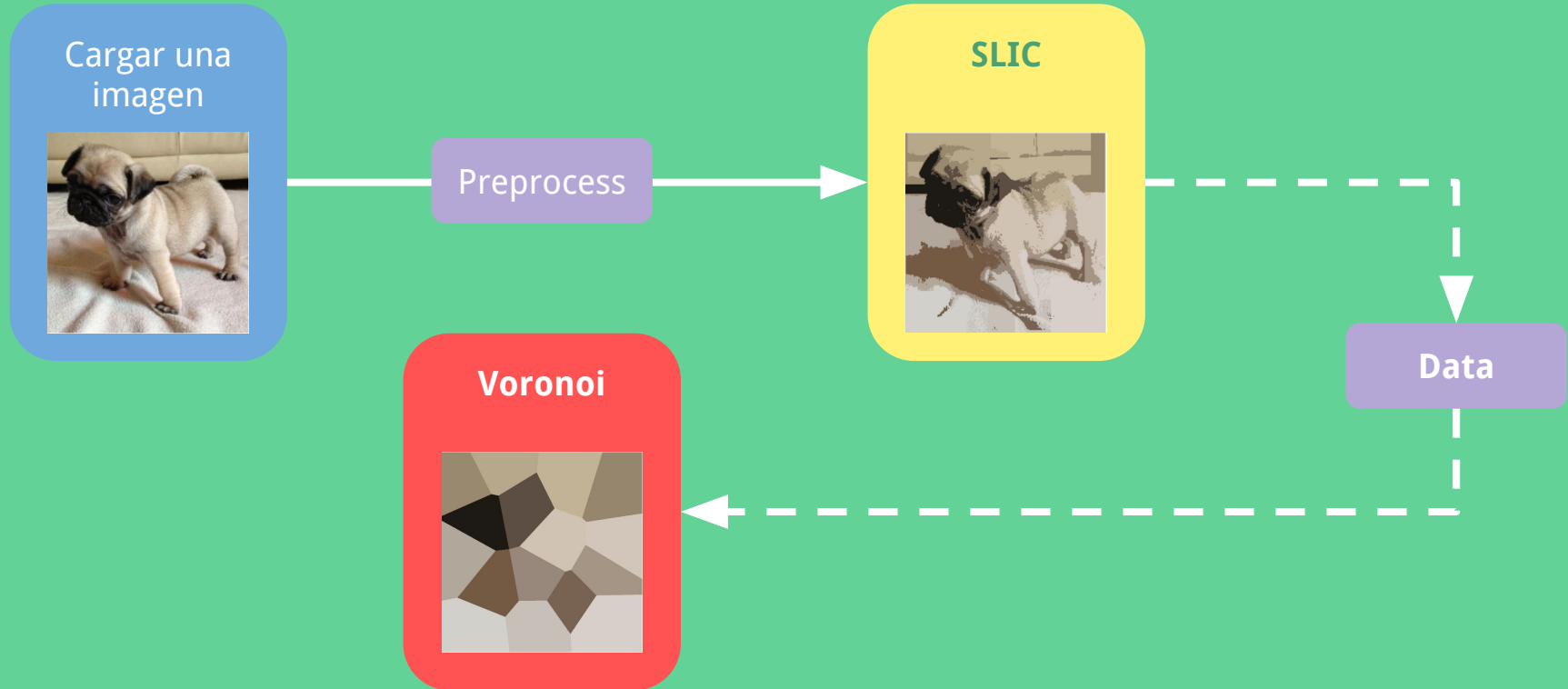
Procedimiento general



Procedimiento general



Procedimiento general



SLIC (Simple Linear Iterative Clustering)

SLIC

Genera superpixels mediante clustering de píxeles basados en la similitud de color y su proximidad en el plano. Esto se hace en el espacio 5-dimensional [\mathbf{RGBxy}], donde [\mathbf{RGB}] es el vector de color y [\mathbf{xy}] el vector de posición.

SLIC

Genera superpixels mediante clustering de píxeles basados en la similitud de color y su proximidad en el plano. Esto se hace en el espacio 5-dimensional [\mathbf{RGBxy}], donde [\mathbf{RGB}] es el vector de color y [\mathbf{xy}] el vector de posición.

La distancia espacial depende del tamaño de la imagen. Definimos una nueva distancia que considera el tamaño del superpixel. Así, forzamos la similitud del color, así como la proximidad de píxeles, de tal forma que los tamaños de cluster esperados y su extensión son aproximadamente iguales.

Distancia

- ❖ El algoritmo tiene como entrada un entero K como el número deseado de superpixels de aproximadamente el mismo tamaño.

Distancia

- ❖ El algoritmo tiene como entrada un entero K como el número deseado de superpixels de aproximadamente el mismo tamaño.
- ❖ Así, para una imagen de N píxeles, el tamaño aproximado de cada superpixel es N/K .

Distancia

- ❖ El algoritmo tiene como entrada un entero K como el número deseado de superpixels de aproximadamente el mismo tamaño.
- ❖ Así, para una imagen de N píxeles, el tamaño aproximado de cada superpixel es N/K .
- ❖ Luego, tendríamos un centro de superpixel $S = \sqrt{N/K}$ en cada intervalo de rejilla.

Distancia

- ❖ El algoritmo tiene como entrada un entero K como el número deseado de superpixels de aproximadamente el mismo tamaño.
- ❖ Así, para una imagen de N píxeles, el tamaño aproximado de cada superpixel es N/K .
- ❖ Luego, tendríamos un centro de superpixel $S = \sqrt{N/K}$ en cada intervalo de rejilla.
- ❖ Inicializamos nuestro algoritmo escogiendo K centros iniciales:
 $C_k = [r_k, g_k, b_k, x_k, y_k]^T$ con $k = [1, K]$ en intervalos de rejilla regulares.

Distancia

- ❖ El algoritmo tiene como entrada un entero K como el número deseado de superpixels de aproximadamente el mismo tamaño.
- ❖ Así, para una imagen de N píxeles, el tamaño aproximado de cada superpixel es N/K .
- ❖ Luego, tendríamos un centro de superpixel $S = \sqrt{N/K}$ en cada intervalo de rejilla.
- ❖ Inicializamos nuestro algoritmo escogiendo K centros iniciales:
 $C_k = [r_k, g_k, b_k, x_k, y_k]^T$ con $k = [1, K]$ en intervalos de rejilla regulares.
- ❖ Dado que la extensión espacial de cada superpixel es aproximadamente S^2 , podemos asegurar que los píxeles asociados a cada centro yacen en una región de tamaño aproximado $2S \times 2S$.

Distancia

Finalmente, definimos nuestra distancia D_s como:

Distancia

Finalmente, definimos nuestra distancia D_s como:

$$d_{rgb} = \sqrt{(r_k - r_i)^2 + (g_k - g_i)^2 + (b_k - b_i)^2},$$

$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2},$$

$$D_s = d_{rgb} + (m/S)d_{xy},$$

Distancia

Finalmente, definimos nuestra distancia D_s como:

$$d_{rgb} = \sqrt{(r_k - r_i)^2 + (g_k - g_i)^2 + (b_k - b_i)^2},$$

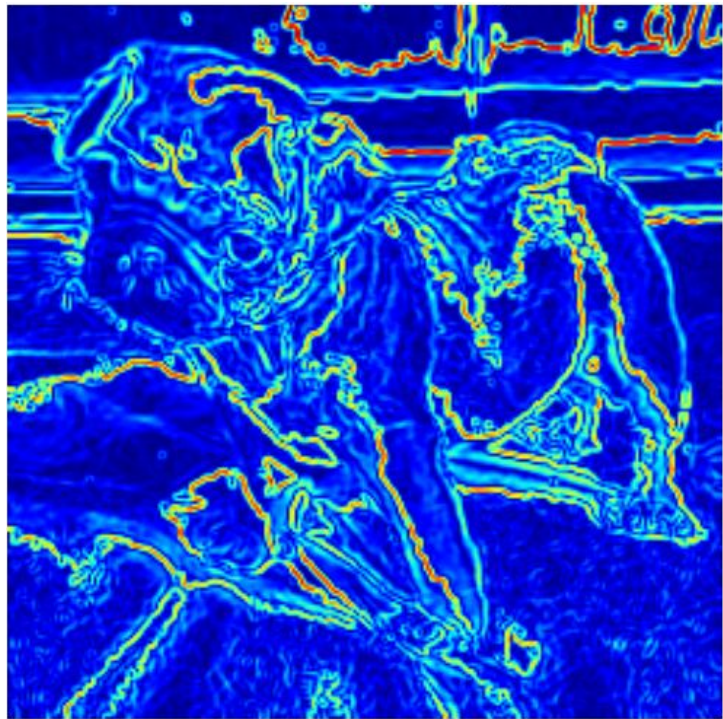
$$d_{xy} = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2},$$

$$D_s = d_{rgb} + (m/S)d_{xy},$$

donde D_s es la suma de la distancia [**RGB**] y [**xy**] normalizada de acuerdo al tamaño de la región. Mientras más alto el valor de m , más énfasis a la proximidad espacial y por ende, tendríamos un cluster más compacto. Este valor puede estar en el rango [1,20]; para el proyecto utilizamos $m = 10$.

Pre-procesamiento

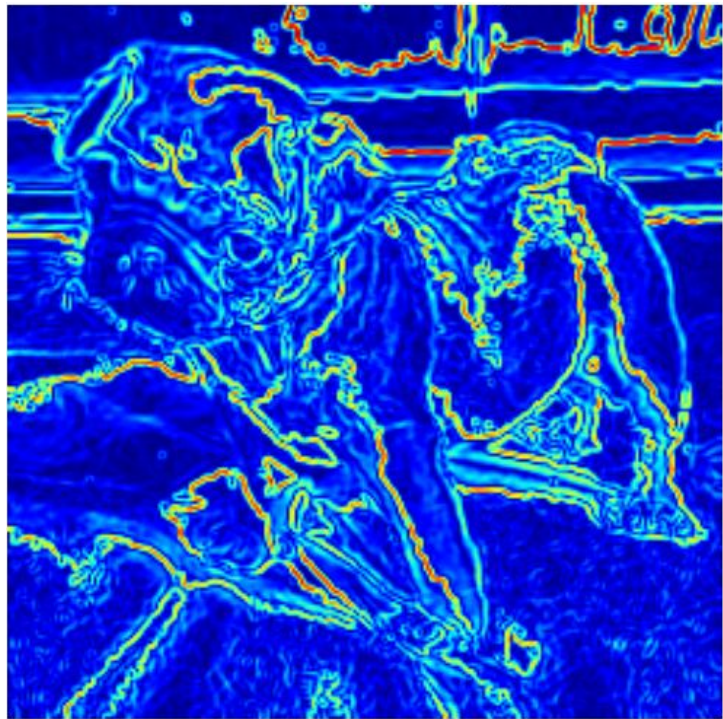
Para esta parte, consideramos las siguientes acciones:



Pre-procesamiento

Para esta parte, consideramos las siguientes acciones:

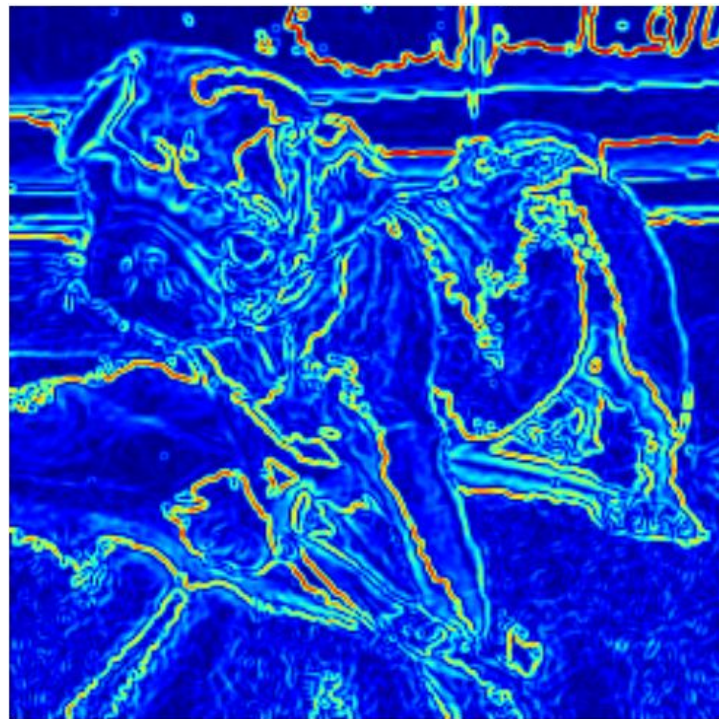
1. Las imágenes fueron reescaladas a imágenes cuadradas cuyos lados son potencia de 2, para trabajar con matrices más pequeñas y tratar de minimizar tiempos.



Pre-procesamiento

Para esta parte, consideramos las siguientes acciones:

1. Las imágenes fueron reescaladas a imágenes cuadradas cuyos lados son potencia de 2, para trabajar con matrices más pequeñas y tratar de minimizar tiempos.
2. En la inicialización, dadas las imprecisiones del clustering en caso de iniciar con un centro en un borde, calculamos la magnitud del gradiente y obtenemos el valor mínimo para colocar un centro inicial ahí.

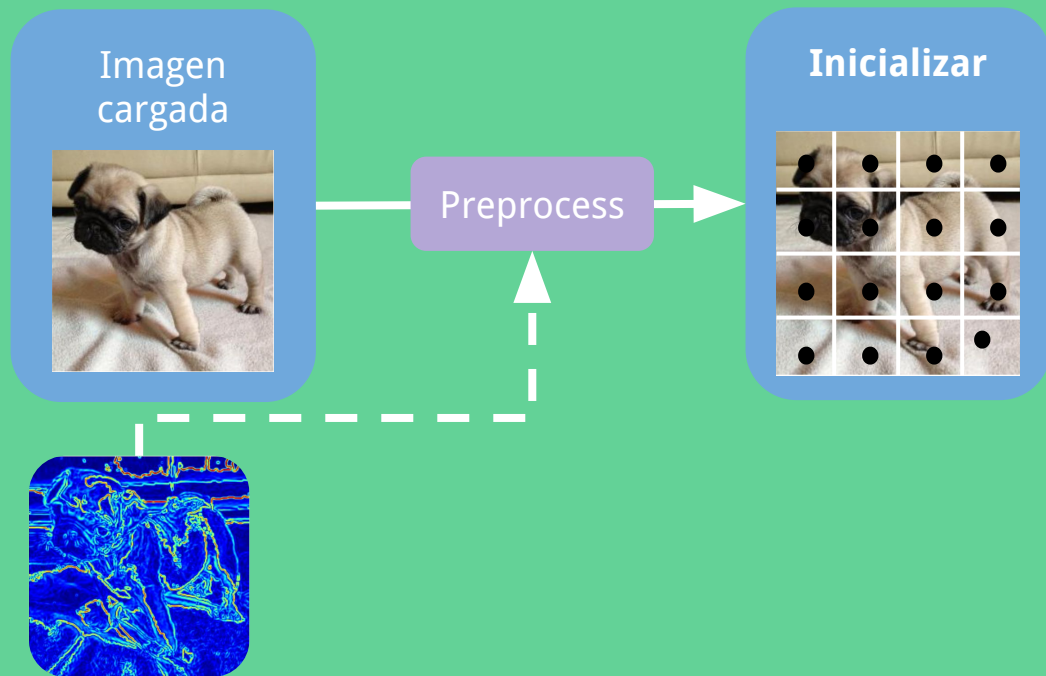


SLIC Procedure

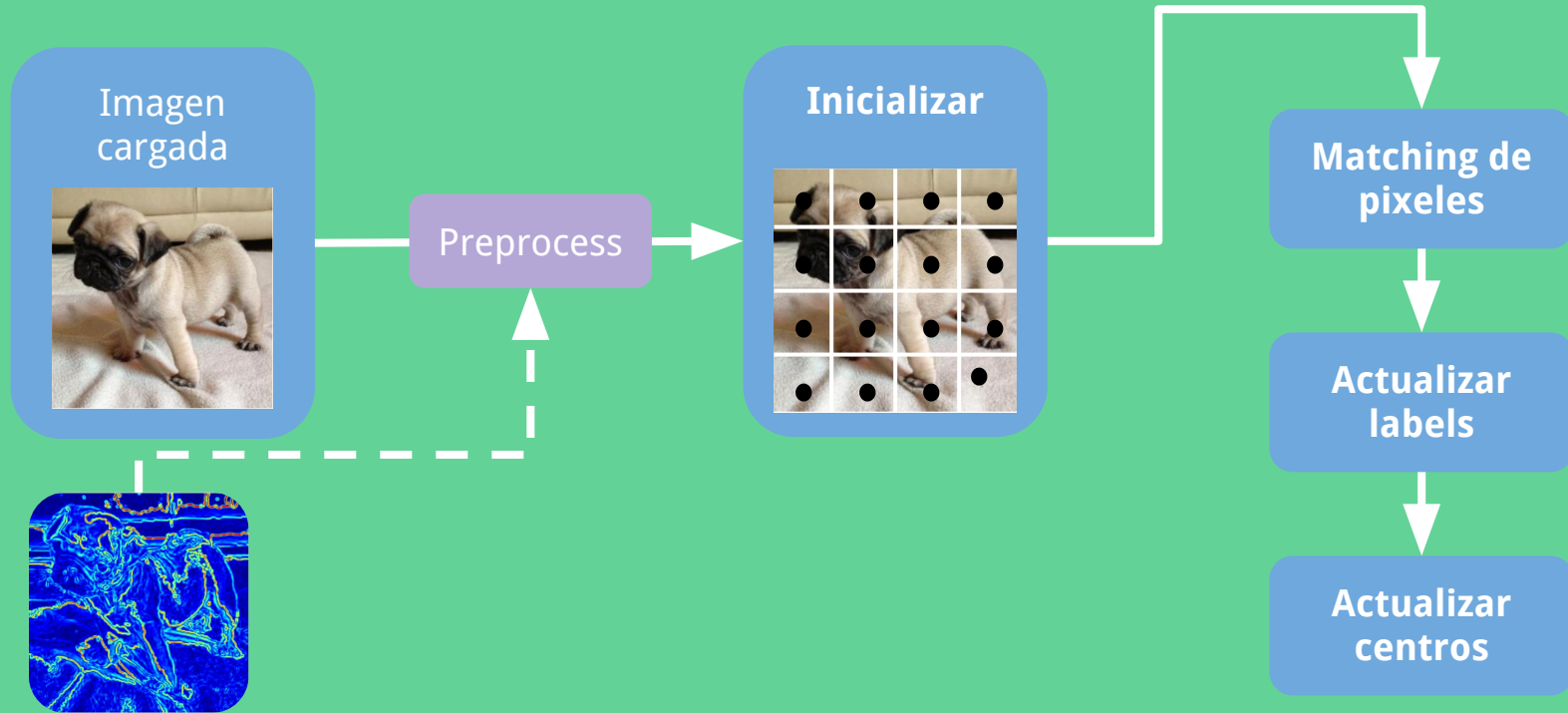
Algoritmo. Segmentación eficiente de superpixel.

1. Inicializar los centros en $C_k = [r_k, g_k, b_k, x_k, y_k]^T$ dentro de rejillas de tamaño S .
 2. Perturbar los centros de cada cluster en un rectángulo de $n \times n$ a la posición mínima del gradiente.
 3. REPEAT
 4. FOR cada centro C_k DO:
 5. Asignar el mejor matching de pixeles en la vecindad $2S \times 2S$.
 6. ENDFOR
 7. Calcula los nuevos centros y el error residual E .
 8. UNTIL $E \leq \text{umbral}$.
 9. Forzar conectividad.
-

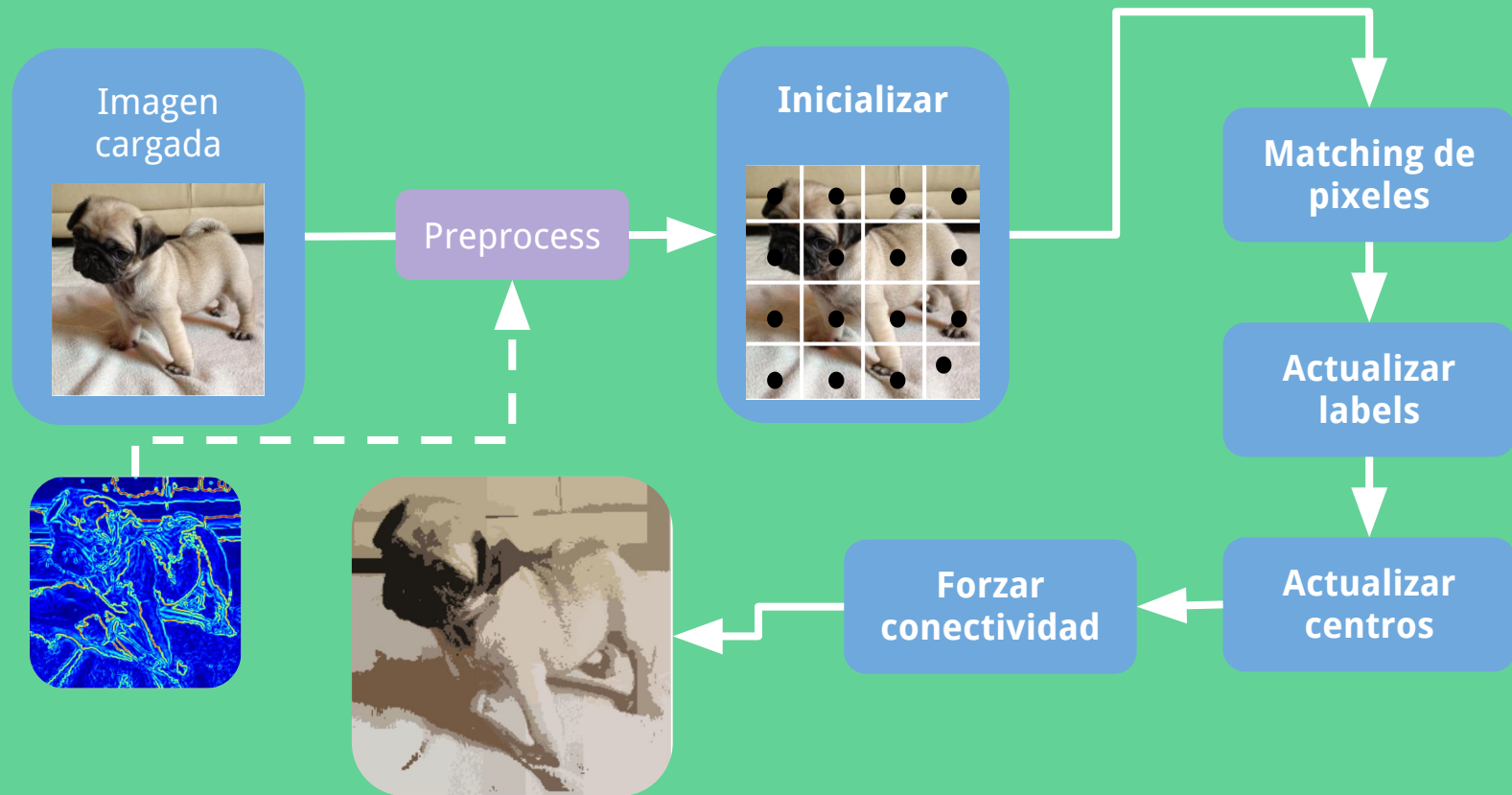
SLIC



SLIC



SLIC

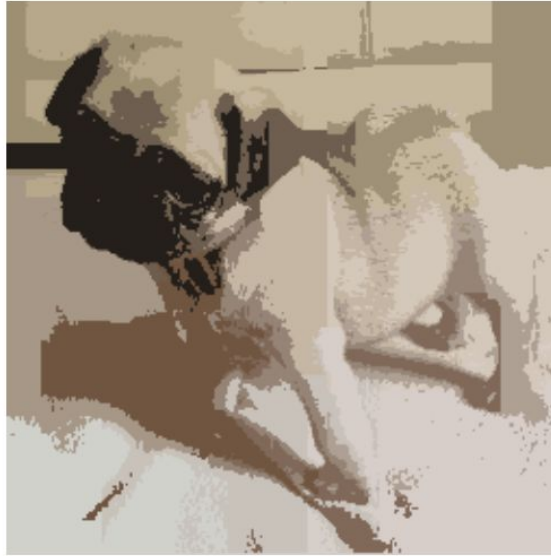


Algunos resultados

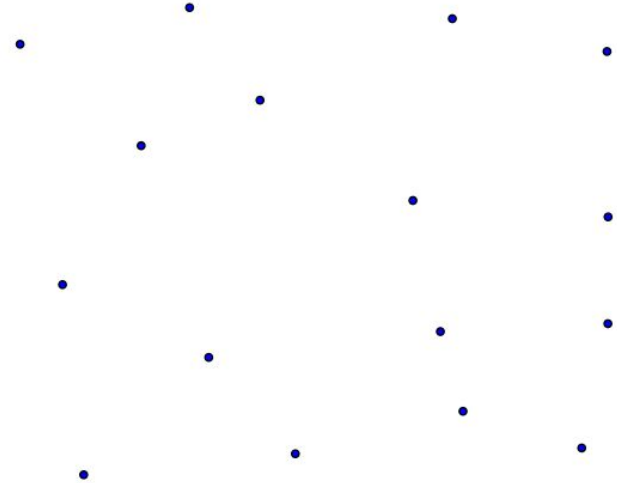
Original Image



SLIC with 4^2 clusters and 2 iterations



Scatter of new centers



Algunos resultados

Original Image



SLIC with 8^2 clusters and 5 iterations



Scatter of new centers

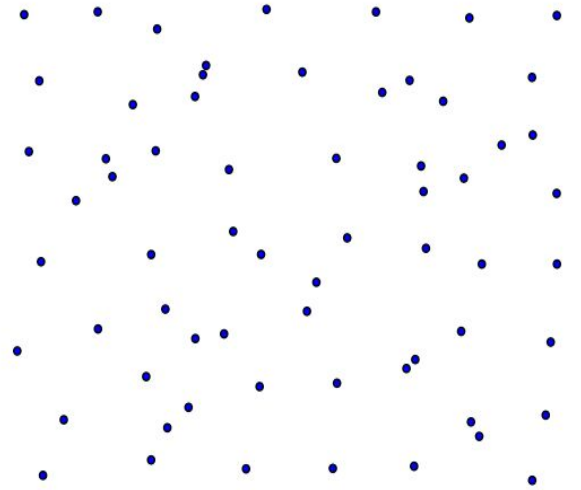


Diagrama de Voronoi

¿Por qué un diagrama de Voronoi?

- ❖ Por las regiones que SLIC regresa.

¿Por qué un diagrama de Voronoi?

- ❖ Por las regiones que SLIC regresa.
- ❖ Podríamos aplicar más procesos al SLIC, pero una buena opción es usar un diagrama de Voronoi para aproximar regiones convexas.

¿Por qué un diagrama de Voronoi?

- ❖ Por las regiones que SLIC regresa.
- ❖ Podríamos aplicar más procesos al SLIC, pero una buena opción es usar un diagrama de Voronoi para aproximar regiones convexas.
- ❖ SLIC nos proporciona los centros de cada cluster y el color del superpixel en dicha región, gracias a la estructura del espacio 5-dimensional [*RGBxy*].

¿Por qué un diagrama de Voronoi?

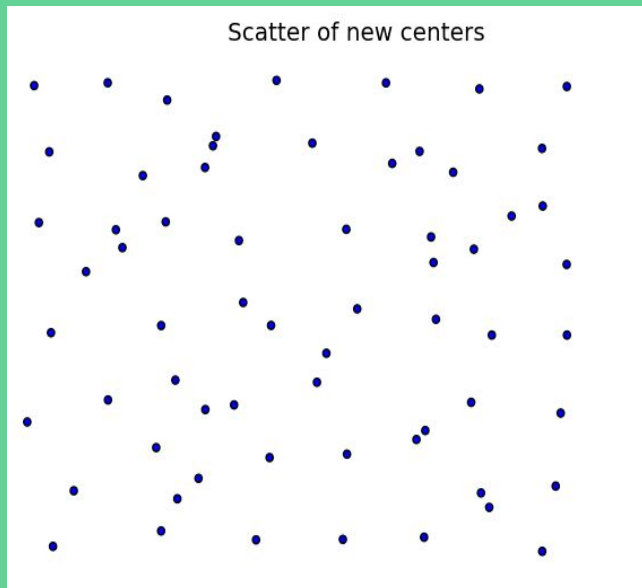
- ❖ Por las regiones que SLIC regresa.
- ❖ Podríamos aplicar más procesos al SLIC, pero una buena opción es usar un diagrama de Voronoi para aproximar regiones convexas.
- ❖ SLIC nos proporciona los centros de cada cluster y el color del superpixel en dicha región, gracias a la estructura del espacio 5-dimensional [*RGBxy*].
- ❖ Con esto podemos obtener regiones convexas.

¿Por qué un diagrama de Voronoi?

- ❖ Por las regiones que SLIC regresa.
- ❖ Podríamos aplicar más procesos al SLIC, pero una buena opción es usar un diagrama de Voronoi para aproximar regiones convexas.
- ❖ SLIC nos proporciona los centros de cada cluster y el color del superpixel en dicha región, gracias a la estructura del espacio 5-dimensional [*RGBxy*].
- ❖ Con esto podemos obtener regiones convexas.
- ❖ Lo que ayuda a resolver nuestro problema es guardar los centros y el color. Al final sólo guardamos nuestro vector C_k .

Dados los centros y los colores, sólo obtenemos el diagrama de Voronoi.

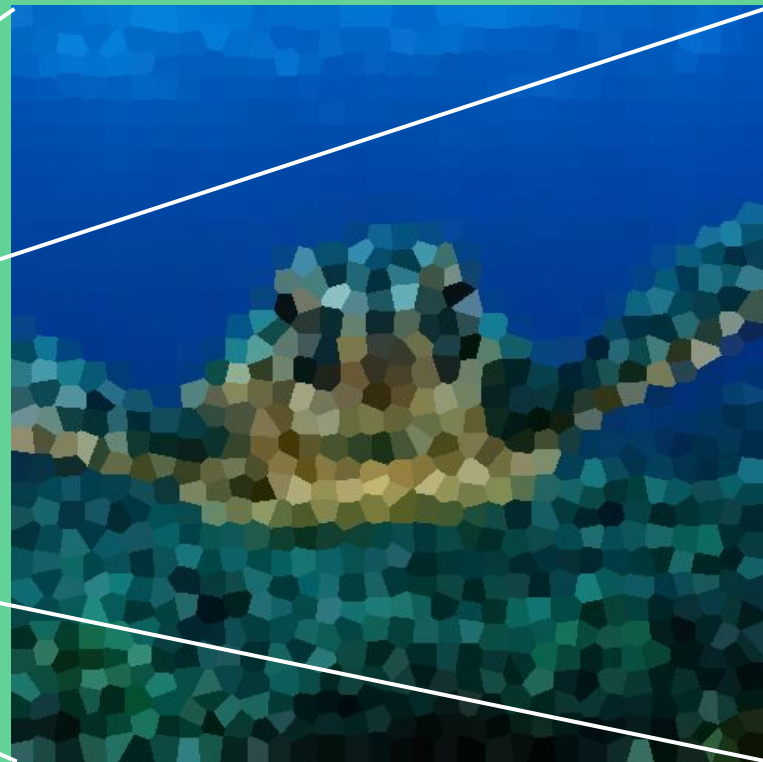
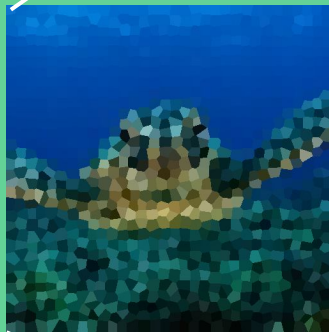
SLIC + Voronoi =



Para una buena cantidad de puntos, obtenemos mejores aproximaciones a la imagen original.

Multiplicando los centros
por un factor, podemos
reescalar nuestra imagen
reconstruida.

SLIC + Voronoi =



Conclusiones

SLIC + Voronoi

¿Cumple con lo necesitado?

- ❖ Usando un diagrama de Voronoi podemos reconstruir una imagen, con información importante como coordenada de un superpixel en el plano y su color asociado.
 - ❖ La ventaja de las coordenadas es que al multiplicar por un escalar, podemos reconstruir nuestra imagen tan grande/pequeña como queramos.
-

Trabajo a Futuro

SLIC + Voronoi

¿Qué podría ser mejorado?

- ❖ Quizá un triangulación de Delaunay ayude a una buena reconstrucción de la imagen original, sin embargo habría que hacer un análisis en la estructura y almacenamiento.
 - ❖ Limpiar el código: Usar algoritmos de buena complejidad y que estén bien implementados.
-



RODOLFO FERRO

(ferro@cimat.mx)

Estudiante de Lic. en Matemáticas

Universidad de Guanajuato

¡Por su atención, gracias!

