
Implementación de Algoritmo de Superpíxeles para la Segmentación de Imágenes a Color

¹Ing. Alan Hernández Girón
alan.hernandez@polimentes.com
²Dra. Yesenia Eleonor González Navarro
ygonzalez@ipn.mx
²M. en C. Adrián Morales Blas
amorales@ipn.mx

¹BECPOLIMENTES SAPI de CV. Ciudad de México.

²Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas-IPN. Ciudad de México

Resumen

En el presente artículo se implementa el algoritmo SLIC (Simple Linear Iterative Clustering, por sus siglas en inglés) de agrupamiento de datos, catalogado dentro de la técnica de superpíxeles para la segmentación de imágenes a color. Para las pruebas se utilizaron imágenes de 640×480 píxeles y se implementó en el lenguaje de programación Matlab®.

Palabras clave: Algoritmo SLIC, superpíxeles, segmentación de imágenes a color.

Introducción

La segmentación de imágenes digitales mediante superpíxeles se ha convertido en una técnica popular para aplicaciones en visión artificial [1, 2]. El término de superpíxel fue introducido en 2003 [3] y se refiere al agrupamiento de píxeles con rasgos similares dentro de una imagen. Los superpíxeles proporcionan una manera conveniente de realizar una caracterización focal de parámetros en una imagen digital.

A la fecha se han desarrollado distintos algoritmos de superpíxeles y se han propuesto categorizarlos de acuerdo al tipo de algoritmos o técnicas que toman como base [1].

Este trabajo se centra en la segmentación de imágenes a color a partir de la implementación del algoritmo SLIC (Simple Linear Iterative Clustering, por sus siglas en inglés) que está catalogado como un algoritmo basado en técnicas de agrupamiento de datos difusos y que reporta un costo computacional bajo comparado con otras arquitecturas [4].

I. Desarrollo

Algunas de las características de los algoritmos de superpíxeles son [3]:

- Son computacionalmente eficientes: Reducen la complejidad de una imagen de cientos de miles de píxeles a solo unos cuantos cientos de superpíxeles.
- Realizan una representación eficiente de la imagen: Los superpíxeles pueden relacionarse entre sí dentro de una sección más grande en la imagen.
- Los superpíxeles son significativos perceptualmente: Todos los píxeles dentro de un superpíxel son uniformes y similares en características, por ejemplo, en color, textura y posición espacial.
- Debido a que los superpíxeles son resultado de una sobre-segmentación, muchas de las estructuras de la imagen se conservan.

Para que los superpíxeles sean útiles, estos deben ser rápidos, fáciles de usar y deben producir segmentaciones de alta calidad. La mayoría de los algoritmos que implementan superpíxeles no cumplen con estos requisitos, por el contrario, estos padecen de un alto costo computacional, pobre calidad de segmentación, forma y tamaño inconsistente, o contienen múltiples parámetros de ajuste [5]. La aproximación que se utiliza en este artículo, a pesar de simple, aborda estos problemas y genera alta calidad de segmentación, superpíxeles cuasi-uniformes, así como una forma y tamaño compacto de cada uno de ellos.

Algoritmo SLIC

El algoritmo SLIC genera superpíxeles mediante un agrupamiento de píxeles basados en su similitud de color y su proximidad en el plano de la imagen. Se realiza en un espacio de 5 dimensiones 5-D $[labxy]$, donde $[lab]$ es el vector-píxel de color en el espacio de color CIELAB [6], el cual es extensamente considerado como un espacio de color perceptivamente lineal (uniforme), es decir, que un cambio en el valor de color debe producir un cambio casi de la misma importancia visual; $[xy]$ es la posición del píxel en los ejes horizontal y vertical de la imagen. Mientras que el valor máximo entre dos colores en el espacio de color CIELAB está limitado, la distancia espacial en el plano xy depende del tamaño de la imagen, por lo que no es posible simplemente usar distancia Euclidiana en un espacio 5-D sin antes normalizar las distancias espaciales. Con el fin de agrupar píxeles en un espacio 5-D, el algoritmo SLIC introduce el cálculo de una nueva distancia que considera el tamaño de los superpíxeles, con ella, se hace cumplir la similitud de color así como la proximidad entre cada píxel en este espacio 5-D, de manera que el tamaño del agrupamiento esperado y su extensión espacial sean aproximadamente iguales [4].

En (3) se describe la ecuación que define el cálculo de la nueva distancia d_s a partir de la obtención de distancia en el espacio de color CIELAB y de la distancia espacial entre píxeles.

$$d_{lab} = \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2}, \quad (1)$$

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

$$d_s = d_{lab} + \frac{m}{s} d_{xy}. \quad (3)$$

Donde d_s es la suma de la distancia lab y la distancia xy normalizada mediante el intervalo de cuadrícula S . La variable m introducida en d_s permite el control de la compacidad de un superpíxel. Entre más grande sea el valor de m , más se enfatiza la proximidad entre píxeles y más compacto es el superpíxel.

Implementación del algoritmo SLIC

El algoritmo SLIC toma como base la teoría de agrupamiento de datos difusos [7, 8], por lo que es necesario iniciar el entrenamiento calculando los centros de cúmulos iniciales. Realizando un muestreo de K centros de cúmulos igualmente espaciados dentro de la imagen y colocando dichos centros a las posiciones que corresponden a la posición de menor gradiente dentro de una vecindad de 3×3 píxeles vecinos. El gradiente de la imagen puede obtenerse a partir de:

$$G(x, y) = \|I(x + 1, y) - I(x - 1, y)\|^2 + \|I(x, y + 1) - I(x, y - 1)\|^2, \quad (4)$$

donde $I(x, y)$ es el vector lab que corresponde a un píxel en la posición (x, y) y $\| \cdot \|$ es la norma L^2 . Cada píxel de la imagen se asocia con el centro de cúmulo más cercano de acuerdo a si el área de búsqueda incluye la posición de dicho píxel. Una vez que todos los píxeles son asociados a un cúmulo, se calcula un nuevo centro de acuerdo a un vector promedio $labxy$ formado por todos los píxeles que pertenecen a ese cúmulo. Este proceso se repite de forma iterativa hasta cumplirse una condición de paro, que puede ser el valor de distancia entre la posición actual y anterior de los centros de cúmulos (cuando el valor de distancia varíe por debajo de cierto umbral, que el algoritmo se detenga).

El diagrama de flujo de la Figura 1 muestra la secuencia del algoritmo SLIC para la obtención de superpíxeles aplicado a la segmentación de imágenes.

II. Resultados

La implementación del algoritmo se realizó en un procesador core i5 a 2.5 GHz. La Figura 2 muestra el procesamiento de una imagen utilizando distintos valores de K (número de superpíxeles). También se muestra el tiempo de procesamiento de cada imagen.

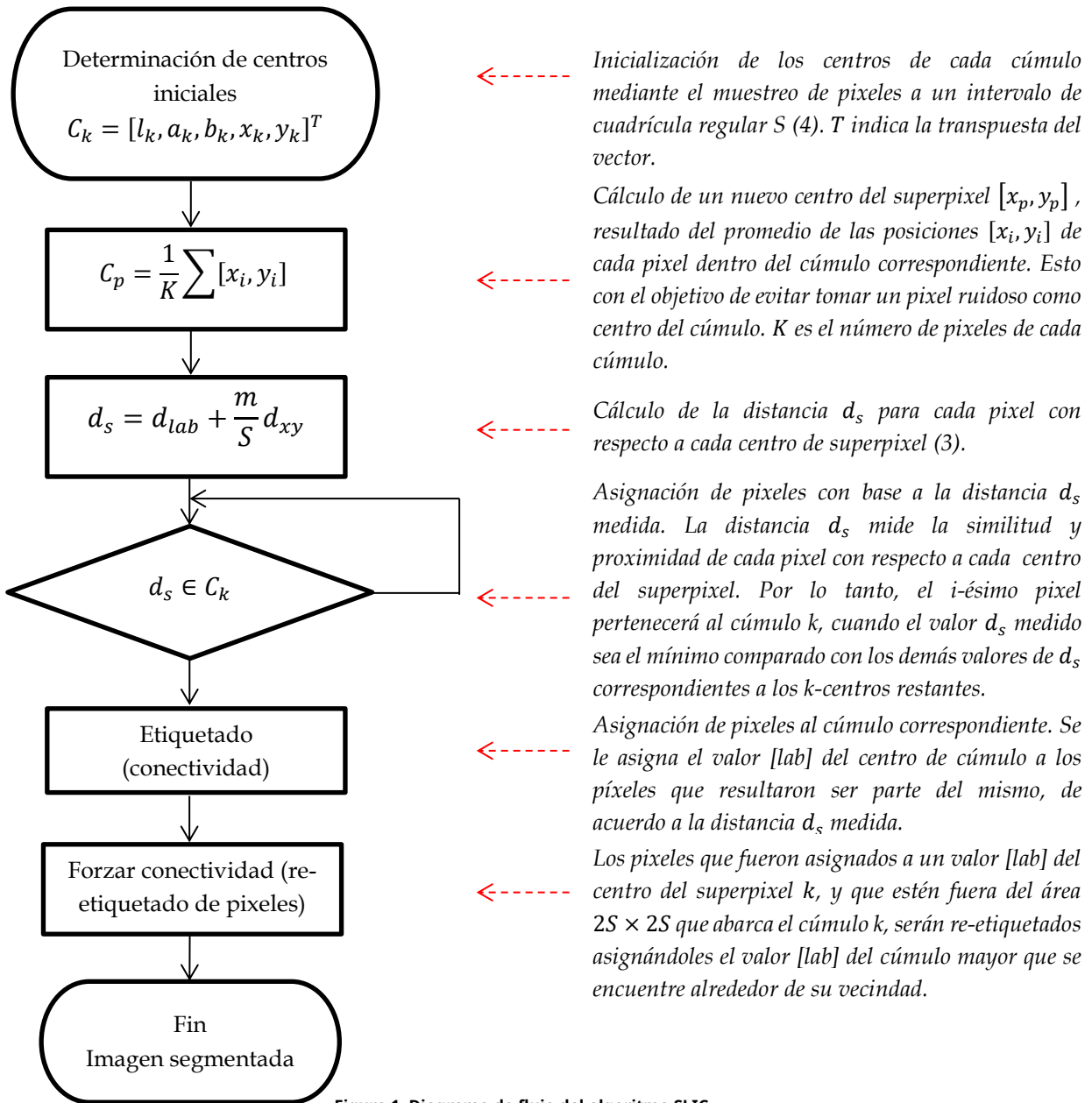


Figura 1. Diagrama de flujo del algoritmo SLIC.



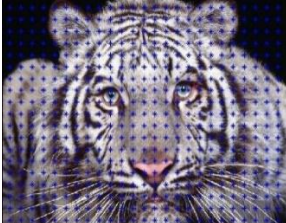

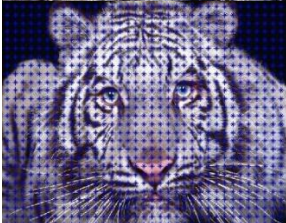

Imagen de entrada	Imagen de salida	Número de superpíxeles (K)	Tiempo de procesamiento [s]
		100	18.33
		500	82.93
		1000	106.67

Figura 2. Aplicación del algoritmo SLIC de superpíxeles a una imagen utilizando distintos valores de K .

Aplicación

La técnica de superpíxeles puede utilizarse como una etapa que permite simplificar los elementos de una imagen para su extracción posterior. La Figura 3 ofrece una comparación de una imagen original con la misma imagen después del algoritmo con un número de 9 superpíxeles ($K = 9$). Puede observarse que el objeto queda definido en sus contornos y proporciona menos información, lo cual hace más sencilla su extracción de la imagen.



Figura 3. Aplicación del algoritmo SLIC de superpíxeles a una imagen con 9 superpíxeles ($K = 9$).

III. Conclusiones

El algoritmo implementado resulta una alternativa novedosa para la segmentación de imágenes. En la sección de resultados se muestra la diferencia de una segmentación con $K = 100$ y $K = 1000$, donde la última puede destacar mayor visualización de detalles, y por ende, mayor número de clases para una imagen que usa un valor de K alto. Entonces, el valor de K

dependerá del tipo de imágenes que se estén trabajando en una determinada aplicación, considerando el número de objetos que existen en la imagen, es decir, para una imagen que presenta mayor número de patrones u objetos a identificar se utiliza un valor de K alto, por el contrario para una imagen donde existen pocos objetos el valor de K disminuye notablemente resultando en una segmentación efectiva en término de tiempo de procesado. Los tiempos de procesamiento dependen totalmente de la velocidad del procesador de la computadora que ejecuta el algoritmo, y que además, éste se encuentre libre de tareas mientras realiza exclusivamente la ejecución del programa. La ventaja de utilizar esta técnica es que solo necesita de un parámetro para ajustar la salida, el número K de superpíxeles. Algo importante de resaltar es que el algoritmo no realiza un etiquetado o conexión de píxeles.

El algoritmo nos permite compactar información relevante para después extraerla de manera sencilla, dependiendo de las características del superpíxel. Puede ser desde tomando sus valores característicos del elemento o estableciendo un umbral para extraer los elementos de características similares.

Esta técnica muestra preferencia en vecindades cuadradas, debido a los rangos establecidos para corroborar vecindades.

Referencias

- [1] Stutz D., Hermans A., Leibe B. (2017). *Superpixels: An Evaluation of the State-of-the-Art*. Computer Vision and Image Understanding. doi: 10.1016/j.cviu.2017.03.007.
- [2] Achanta R., Shaji A., Smith K., Lucchi, A., Fua, P., Susstrunk. S. (2012). *SLIC superpixels compared to state-of-the-art superpixel methods*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (11): 2274–2281.
- [3] Ren X., Malik J. (2003). *Learning a classification model for segmentation*. International Conference on Computer Vision. Vol. 1: 10–17.
- [4] Achanta R., Shaji A., Smith K., Lucchi A., Fua P., Susstrunk S. (2010). *SLIC Superpixels*. EPFL Technical Report 149300.
- [5] Mori G., Ren X., Efros A., Malik J. (2004). *Recovering Human Body Configurations: Combining Segmentation and Recognition*. CVPR '04, Vol. 2: 326-333.
- [6] Fairchild M. (2005). *Color and Image Appearance Models*". [Color Appearance Models](#). John Wiley and Sons.
- [7] Jang J. S., Sun C. T., Mizutani E. (1997). *Neuro Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. USA: Prentice Hall.
- [8] Guevara Gómez E., Sánchez Tello O., González Navarro Y. (2015). *Segmentación de Imágenes Mediante Técnicas de Agrupamiento de Datos Empleando los Algoritmos K-Means y C-Means*. Bo