

Métodos de Segmentación

GRAPH CUTS

Visión Computacional para imágenes y Video

Profesores

Dr. Gilberto Ochoa Ruiz
MIP Ma. del Refugio Melendez Alfaro
M. en C. Jose Angel Martinez Navarro

THE TEAM

Somos cuatro personas con un **objetivo común**: Comprender y aplicar métodos de **visión computacional** para fortalecer nuestra formación en machine learning e inteligencia artificial. Trabajando juntos desde cursos exigentes como procesamiento de lenguaje natural y métodos avanzados de aprendizaje máquina, **aprendimos a responder a retos** y avanzar como equipo trabajando con rigor, buena comunicación y una colaboración que surgió de manera natural. Esta base nos permitió abordar el tema de **segmentación de imágenes** con claridad y disciplina.



Javier Rebull Saucedo
A01795838
“Bendiciones para todos”



Juan Carlos Pérez
A01795941
“Mejorar es una rutina”



Oscar Enrique García
A01016093
“Constancia antes que talento”



Luis Sánchez Salazar
A01232963
“Divide y vencerás”

Métodos de Segmentación

- Son técnicas que **dividen una imagen en regiones** con significado y sirven para **separar objetos, fondos o estructuras** para facilitar tareas como detección, conteo o medición. Se aplican usando criterios como color, textura, bordes o modelos entrenados que determinan dónde comienzan y terminan dichas regiones.
- En visión computacional funcionan como un paso esencial porque **organizan la imagen** y permiten que los sistemas interpreten su contenido de forma más clara y útil.

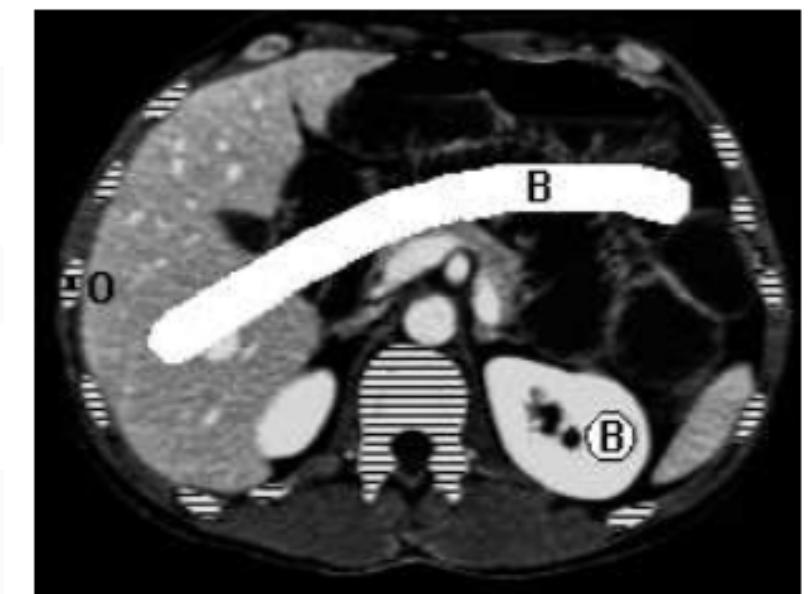
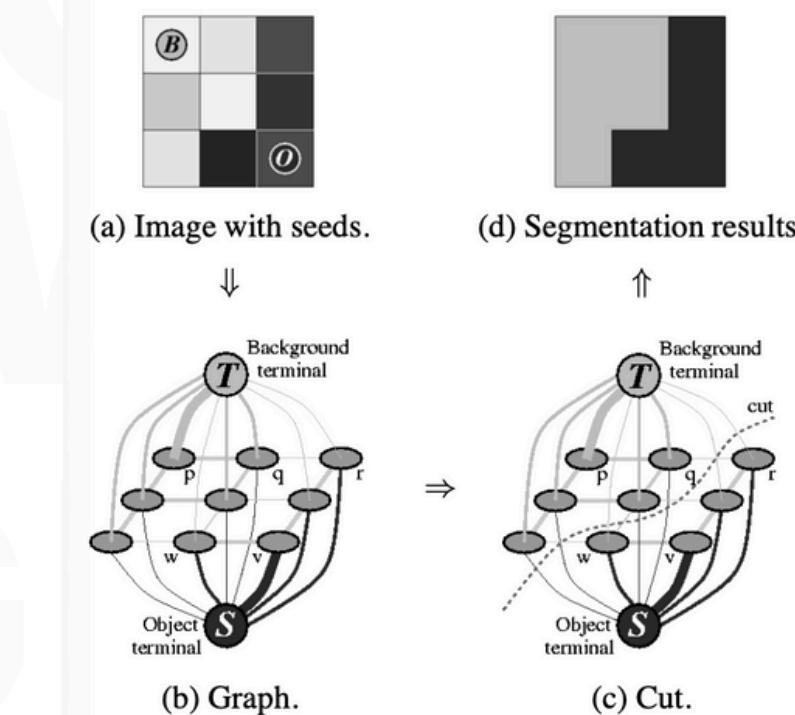
La segmentación divide la imagen en partes útiles que permiten entender y analizar mejor su contenido visual.



GRAPH CUTS: Artículo de Referencia

Boykov, Y. Y., & Jolly, M.-P. (2001). *Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images.* Proceedings of the IEEE International Conference on Computer Vision, 1, 105–112.

- **Yuri Y. Boykov**, profesor de Ciencias de la Computación en la *University of Waterloo* y originario de Rusia, aporta al artículo la base teórica y algorítmica de optimización y graph cuts.
- **Marie-Pierre Jolly**, investigadora en imagen médica en *Siemens Healthineers* y originaria de Francia, aporta la experiencia aplicada que demuestra el método en fotos, video y datos clínicos.
- En este artículo se presenta un método interactivo de **segmentación basado en graph cuts** que combina restricciones del usuario, términos de región y de borde, permite recalcular óptimos globales en 2D y 3D, y demuestra su utilidad en fotos, video y aplicaciones médicas.



GRAPH CUTS: Otros artículos de soporte

Rother, C., Kolmogorov, V., & Blake, A. (2004). GrabCut: Interactive foreground extraction using iterated graph cuts. ACM Transactions on Graphics, 23(3), 309–314.

Una versión más práctica y simplificada de los graph cuts, usando un rectángulo inicial y modelos GMM, mientras que el artículo de Boykov y Jolly presenta la base teórica general para segmentación óptima con restricciones duras.

Shen, Z., Liu, Y., Li, Y., Jiang, B., Chen, S., & Bai, X. (2017). Deep GrabCut for object selection. arXiv:1707.00243.
Extiende la idea de GrabCut con

Extiende la idea de GrabCut con redes neuronales profundas para mejorar la segmentación a partir de una caja inicial, mientras que Boykov y Jolly se enfocan en la optimización clásica sin aprendizaje profundo.

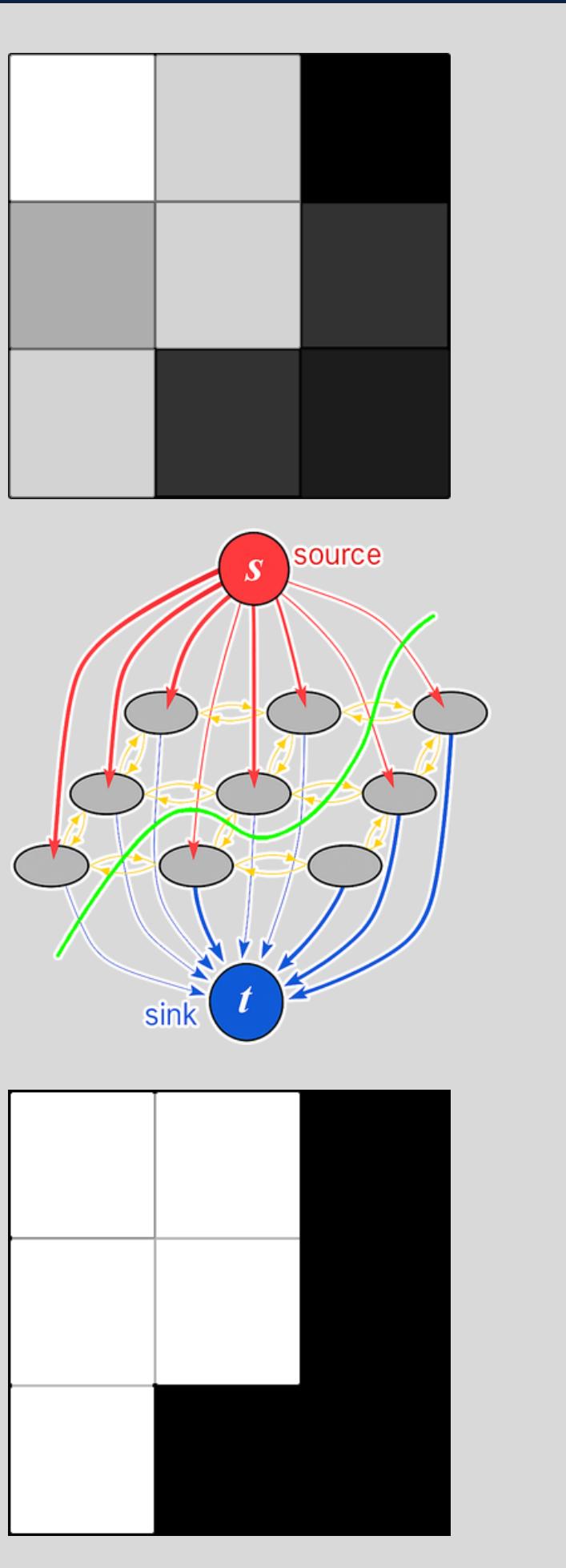
Kumar, A., Verma, R., & Singh, P. (2024). Graph cuts in image segmentation: A review.

Revisan la evolución de los graph cuts, mostrando cómo el trabajo de Boykov establece la base teórica, cómo GrabCut la vuelve práctica e interactiva y cómo Deep GrabCut la moderniza con aprendizaje profundo.

GRAPH CUTS: Introducción

- Es un método de segmentación que modela la imagen a analizar como un **grafo**.
- En esta representación, cada pixel se interpreta como un **nodo**, el cual conecta con **nodos vecinos**.
- Cada **conexión** conlleva un peso o **weight**, el cual depende de la intensidad, similaridad y proximidad.
- La **intensidad** refleja el valor fotométrico del pixel.
- La **similaridad** indica qué tan parecidos son dos pixeles según su color o intensidad.
- La **proximidad** mide qué tan cerca están dentro de la estructura de la imagen.
- A partir de estos weights, Graph Cuts encuentra el corte de energía mínima que **separa el objeto y el fondo de forma estable y coherente**.

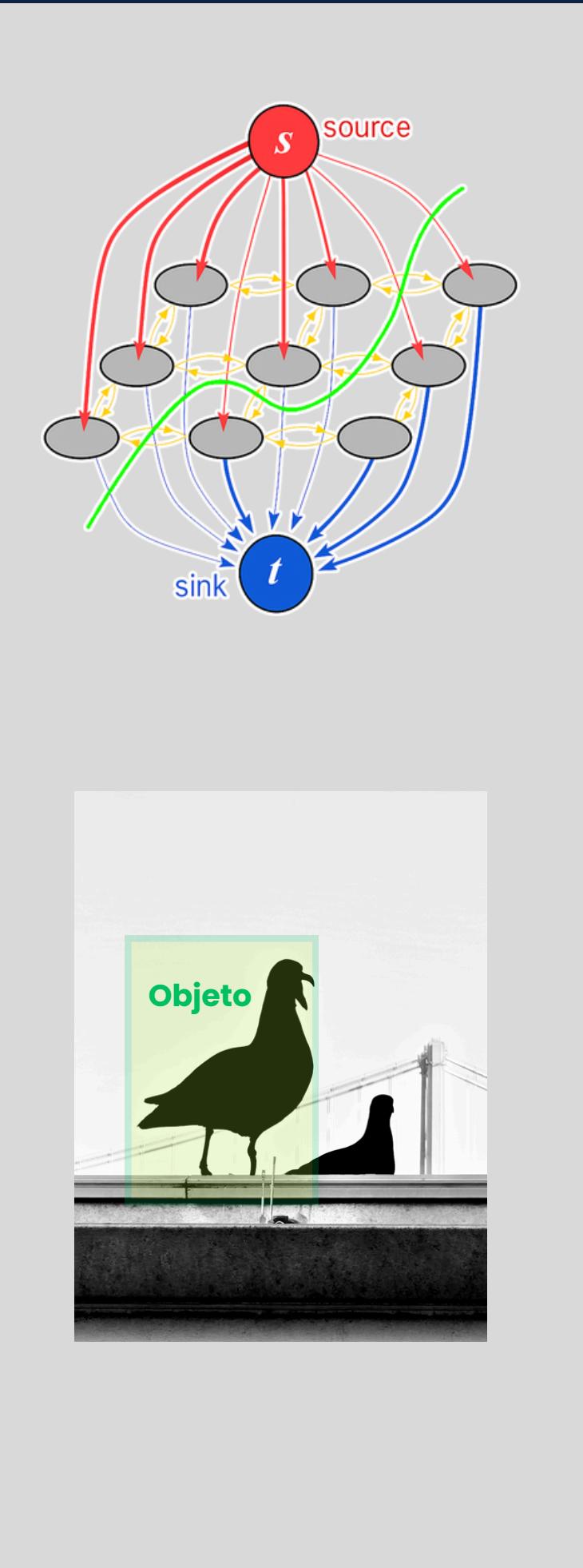
Graph Cuts convierte el problema de segmentación de imagen en un problema de grafos y usa los weights entre pixels para encontrar la separación más estable entre objeto y fondo.



GRAPH CUTS: Definiciones

- Crea dos nodos especiales (**source** y **sink**) que representan objeto y fondo.
- Conecta cada pixel a estos nodos con costos que indican su **afinidad hacia cada clase**.
- Usa **Min-Cut/Max-Flow** para encontrar el corte con el costo total más bajo.
- Requiere **intervención del usuario**, quien marca algunos pixeles como objeto o fondo, que sirve como evidencia inicial para guiar la segmentación.
- Los pixeles marcados forman un **mask** y quedan fijados, por otro lado los weights se ajustan sobre los pixeles restantes.
- Con esta información, el algoritmo construye la energía y obtiene la frontera óptima entre objeto y fondo.

El algoritmo toma marcas iniciales fija esos pixeles y ajusta los costos del resto para encontrar la frontera más coherente entre objeto y fondo.

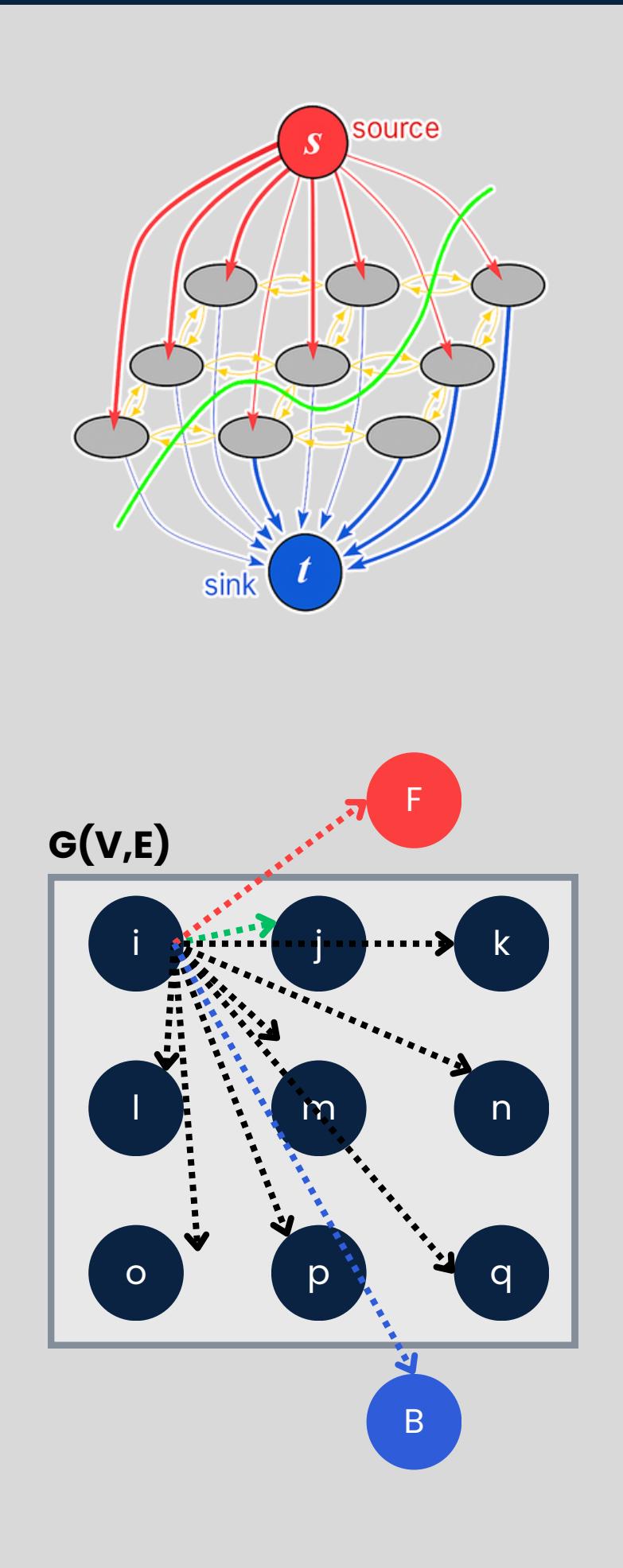


GRAPH CUTS: Sustento matemático

- Se construye un grafo $G(V,E)$ donde cada pixel de la imagen es un nodo en V y las conexiones entre pixels forman las aristas en E.
- Cada arista tiene dirección y un peso que refleja la relación entre dos píxeles vecinos.
- El peso w entre un nodo i en posición (x_i, y_i) y un nodo j en posición (x_j, y_j) se calcula usando una función que combina la **distancia espacial** entre ambos y la **diferencia en intensidad** según la fórmula mostrada:

$$w_{ij} = \frac{1}{e^{-\frac{\|f(x_i, y_i) - f(x_j, y_j)\|^2}{2\sigma^2}}}$$

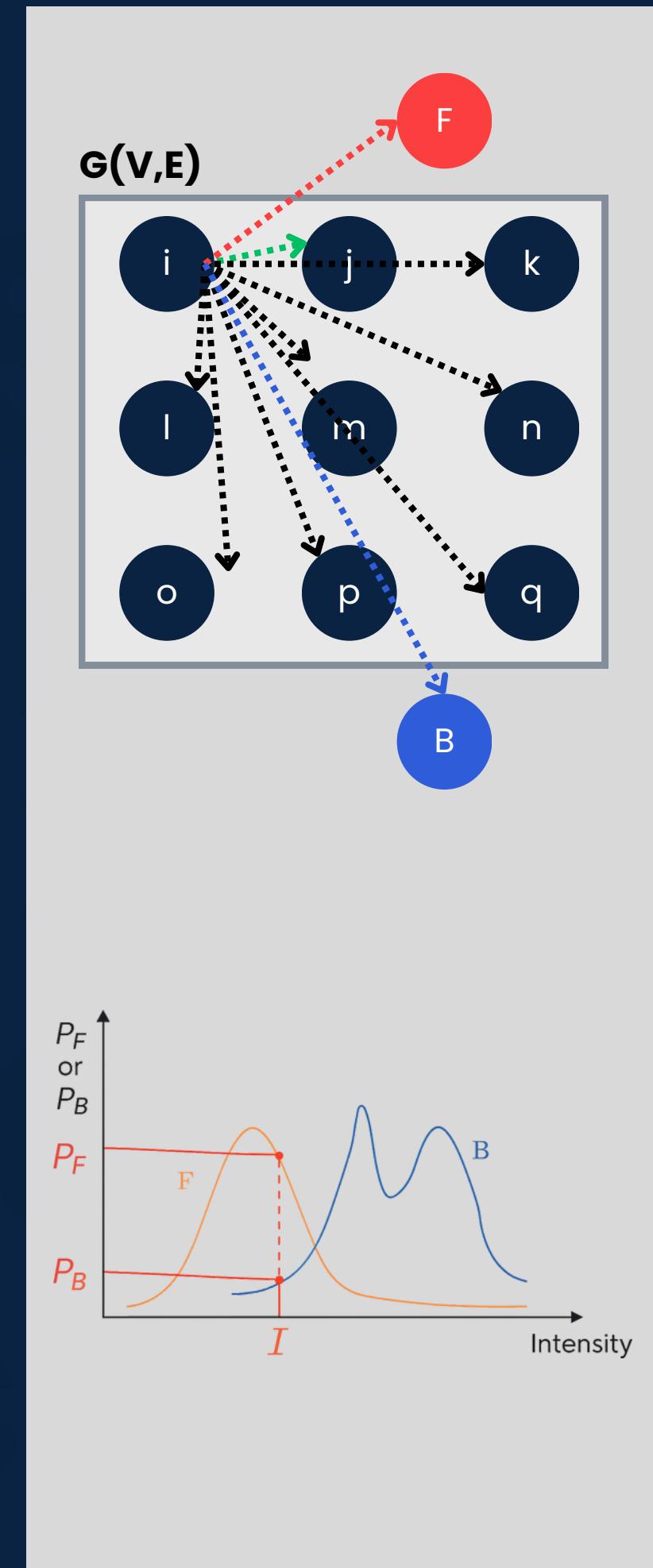
La fórmula ajusta el peso entre dos píxeles según su distancia y su diferencia de intensidad, y eso permite que el grafo refleje de forma natural cómo debería separarse la imagen.



GRAPH CUTS: Sustento matemático

- De manera similar hay pesos respecto a los super nodos de fondo y objeto conocidos como w_{iB} y w_{iF} , respectivamente.
 - Para píxeles marcados como **objeto** $w_{iF} = \infty$ y $w_{iB} = 0$
 - Para píxeles marcados como **fondo** $w_{iF} = 0$ y $w_{iB} = \infty$
 - Para los píxeles **restantes** se estima su probabilidad de ser objeto o fondo
- Con los píxeles marcados se construyen los modelos **PF** y **PB**, los cuales pueden ser representados usualmente como histogramas de color o GMM
- Para los píxeles restantes, se estima su probabilidad de ser objeto o fondo usando los modelos anteriores
- Para cada pixel calcula $PF(i)$ y $PB(i)$ según su color
- Estas probabilidades se convierten en costos **t-links** hacia source y sink usando
$$w_{iF} = -\lambda \log P_B(i)$$
$$w_{iB} = -\lambda \log P_F(i)$$
- Un pixel típico de objeto tendrá bajo costo hacia source y alto hacia sink

Los píxeles no marcados se asignan según qué tanto su color encaja en los modelos de foreground y background, convirtiendo esas probabilidades en costos



GRAPH CUTS: Teoría de grafos

- Con los pesos calculados, el problema de segmentación de imágenes se convierte en uno de grafos.
- Se aplica el algoritmo **Max Flow/Min Cut** propuesto por Lester R. Ford y Ralph Fulkerson, que se viene utilizando desde 1956.
- No se explicará a detalle, pero un buen resumen es el siguiente:
 - Cada nodo del grafo está conectado por enlaces con **capacidad** (los pesos).
 - El algoritmo intenta empujar **flujo** desde el source (objeto) hasta el sink (fondo).
 - Se intenta mover el mayor flujo **sin superar** esas capacidades.
 - **El flujo aumenta** mientras existan caminos con capacidad disponible.
 - Cuando todas las rutas posibles están saturadas se alcanza el **max-flow**.
 - Los enlaces que impiden aumentar más el flujo forman el **min-cut**

El algoritmo busca la máxima cantidad de flujo posible y, al hacerlo, identifica automáticamente el corte más barato que separa objeto y fondo.



GRAPH CUTS: Ventajas y desventajas

Fortalezas

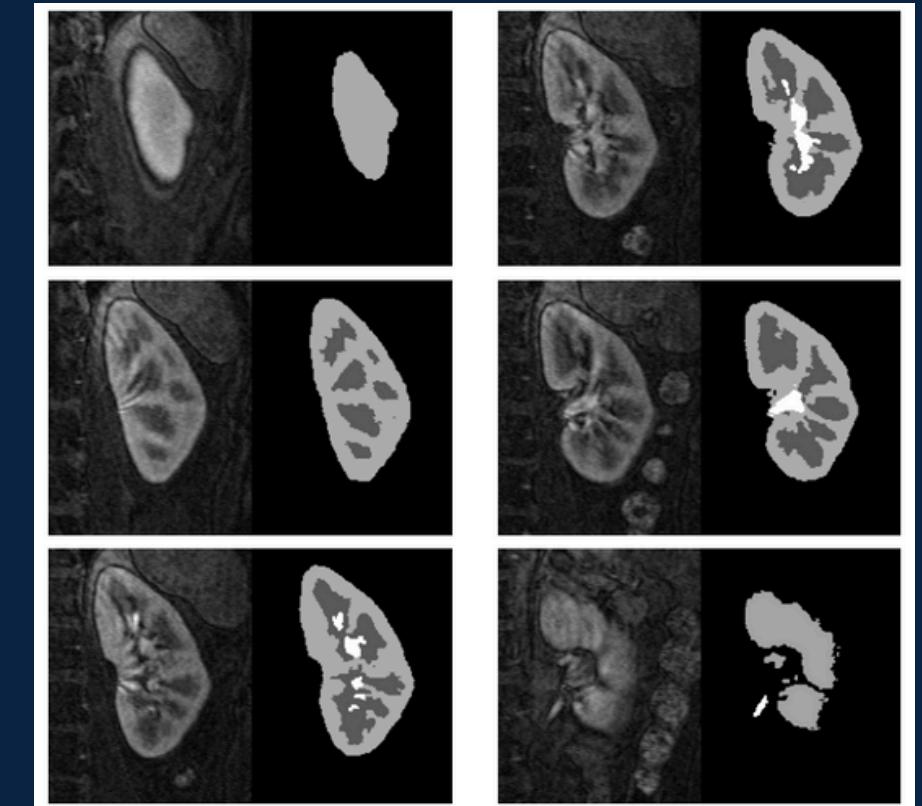
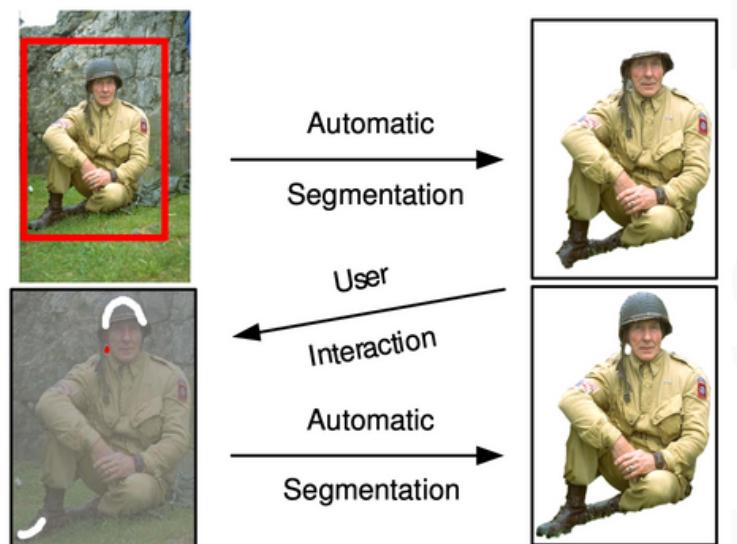
- Produce fronteras muy precisas y estables.
- Combina información local y global de la imagen.
- Permite guiar la segmentación con pocas marcas del usuario.
- Se adapta bien a variaciones de iluminación y color.
- Mantiene coherencia espacial y reduce ruido en la segmentación.
- Tolera bordes suaves gracias a la continuidad entre vecinos.
- Funciona en distintos espacios de color y tipos de imágenes.
- La solución es óptima respecto a la energía definida.
- El modelo probabilístico maneja bien la incertidumbre en los píxeles no marcados.

Debilidades

- Requiere intervención manual para iniciar el proceso.
- El desempeño depende de la calidad de la mask.
- Más costoso computacionalmente que métodos simples.
- Puede fallar si foreground y background tienen colores muy similares.
- Requiere ajustar parámetros como σ y λ .
- Puede sobresegmentar si los modelos de color están mal estimados.
- Depende mucho de la distribución de intensidades en escenas con textura compleja.
- En imágenes grandes, la etapa de max-flow puede ser lenta y demandar mucha memoria.

GRAPH CUTS: Casos de uso

- **Segmentación interactiva.** Permite separar objeto y fondo con pocos indicios del usuario. Aquí entra GrabCut, que usa un rectángulo inicial y un modelo iterativo que mejora la segmentación sin marcar foreground explícito.
- **Matting parcial.** Cuando se necesita una frontera limpia entre objeto y fondo y luego refinar el alfa en la zona de borde.
- **Edición de imágenes.** Eliminar o recortar objetos con precisión, preparar recortes para composiciones y pipelines de edición.
- **Segmentación médica.** Separar órganos, tumores o estructuras anatómicas donde se necesita una frontera estable y globalmente óptima.
- **Restauración de imágenes.** Rellenar regiones dañadas o perdidas, ya que el min-cut puede escoger bordes coherentes.
- **Detección de objetos con cajas imprecisas.** Métodos como GrabCut o Deep GrabCut permiten corregir cajas poco ajustadas y obtener máscaras precisas.



GRAPH CUTS: Conclusiones

Reflexión General

- Graph Cuts **convierte** la segmentación en un problema de energía mínima que produce fronteras precisas.
- **Combina** la información entre píxeles vecinos con la mask definida por el usuario para obtener resultados coherentes.
- Los **pesos** entre píxeles dependen de **distancia** e **intensidad**, reflejando la estructura real de la imagen.
- La **mask** inicial permite construir modelos de color que guían la clasificación de los píxeles no marcados.
- El método es fuerte y estable, pero requiere **intervención inicial** y puede ser costoso en cómputo.
- Funciona bien cuando **foreground** y **background** tienen características distinguibles.
- Es ideal cuando se busca control, precisión y **coherencia visual** en la segmentación.



Javier Rebull Saucedo

"Comprendí que asignar foreground o background no depende solo de la mask, sino también de cómo el modelo interpreta los píxeles restantes. Esa mezcla de guía inicial y razonamiento del algoritmo hace que el resultado sea coherente en diferentes escenarios."



Luis Sánchez Salazar

"Me impresionó la elegancia matemática del método. Ver cómo un problema visual se transforma en uno de optimización y termina ofreciendo una solución clara y estable me hizo entender por qué Graph Cuts sigue siendo utilizado."



Oscar Enrique García

"Me gustó cómo los pesos entre píxeles permiten que la segmentación respete la estructura natural de la imagen. Eso explica por qué funciona bien incluso cuando los bordes no son tan definidos o hay variaciones de iluminación."



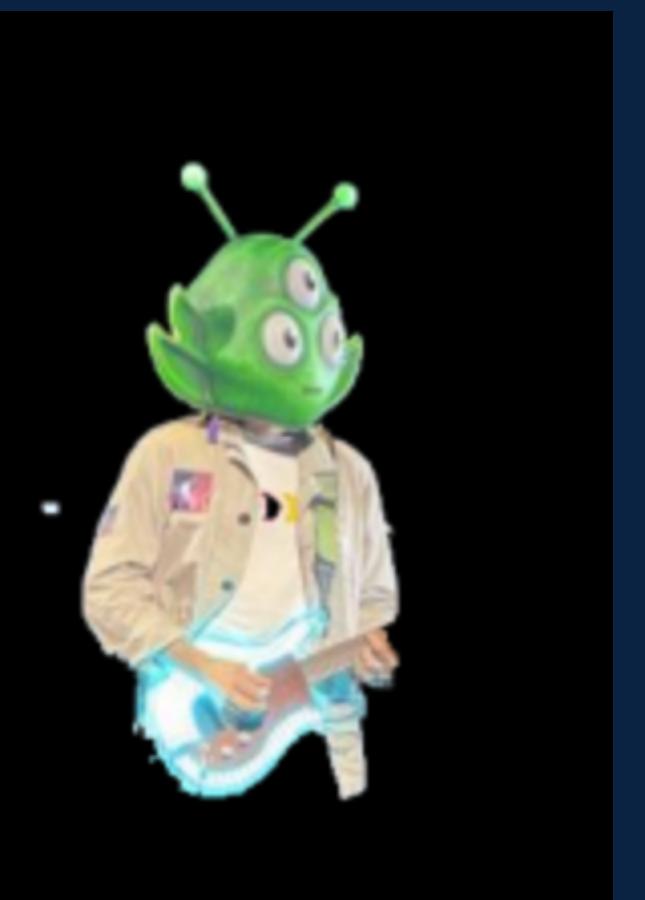
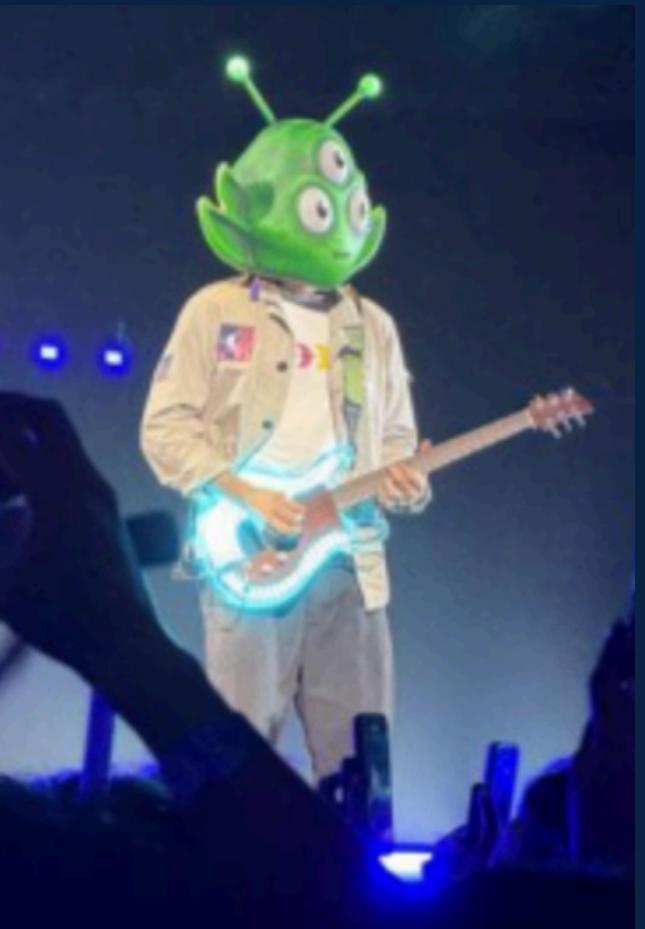
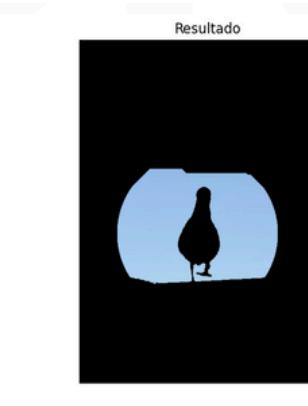
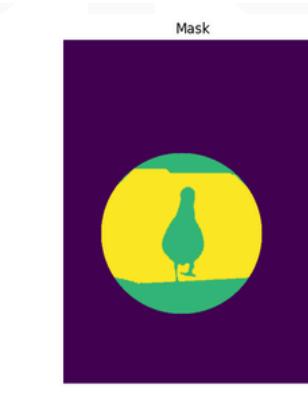
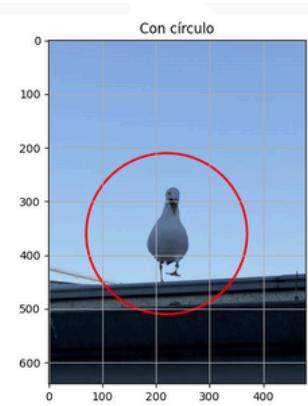
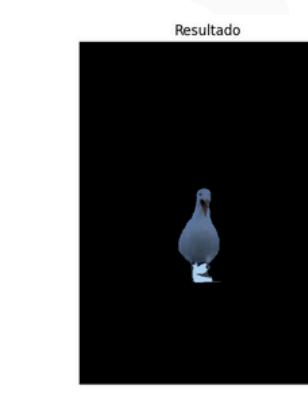
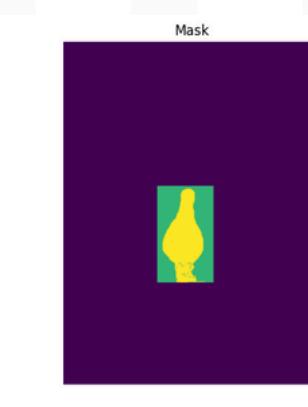
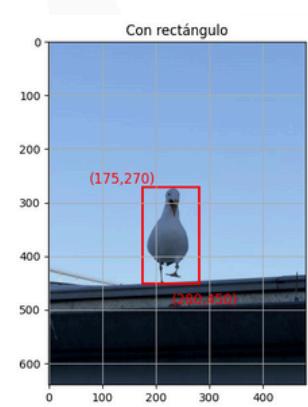
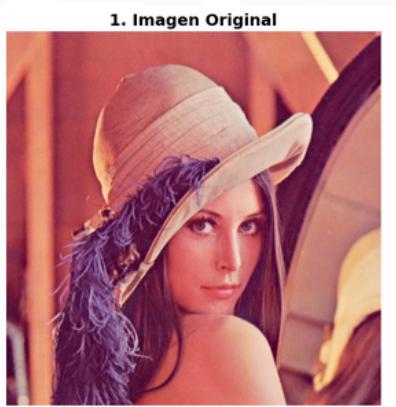
Juan Carlos Pérez

"La mask inicial no es un simple dibujo, sino una guía que orienta todo el proceso. Esa combinación entre intervención humana y cálculo automático lo hace muy útil en tareas de edición, segmentación interactiva o preparación de datos."

EJEMPLO INTERACTIVO

- Link Google Colab:

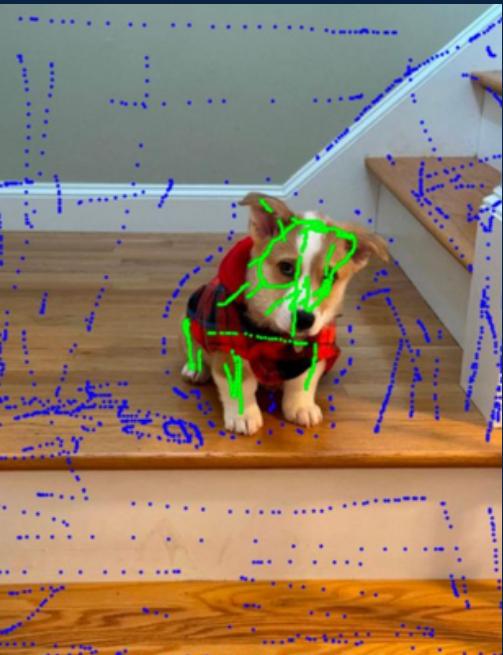
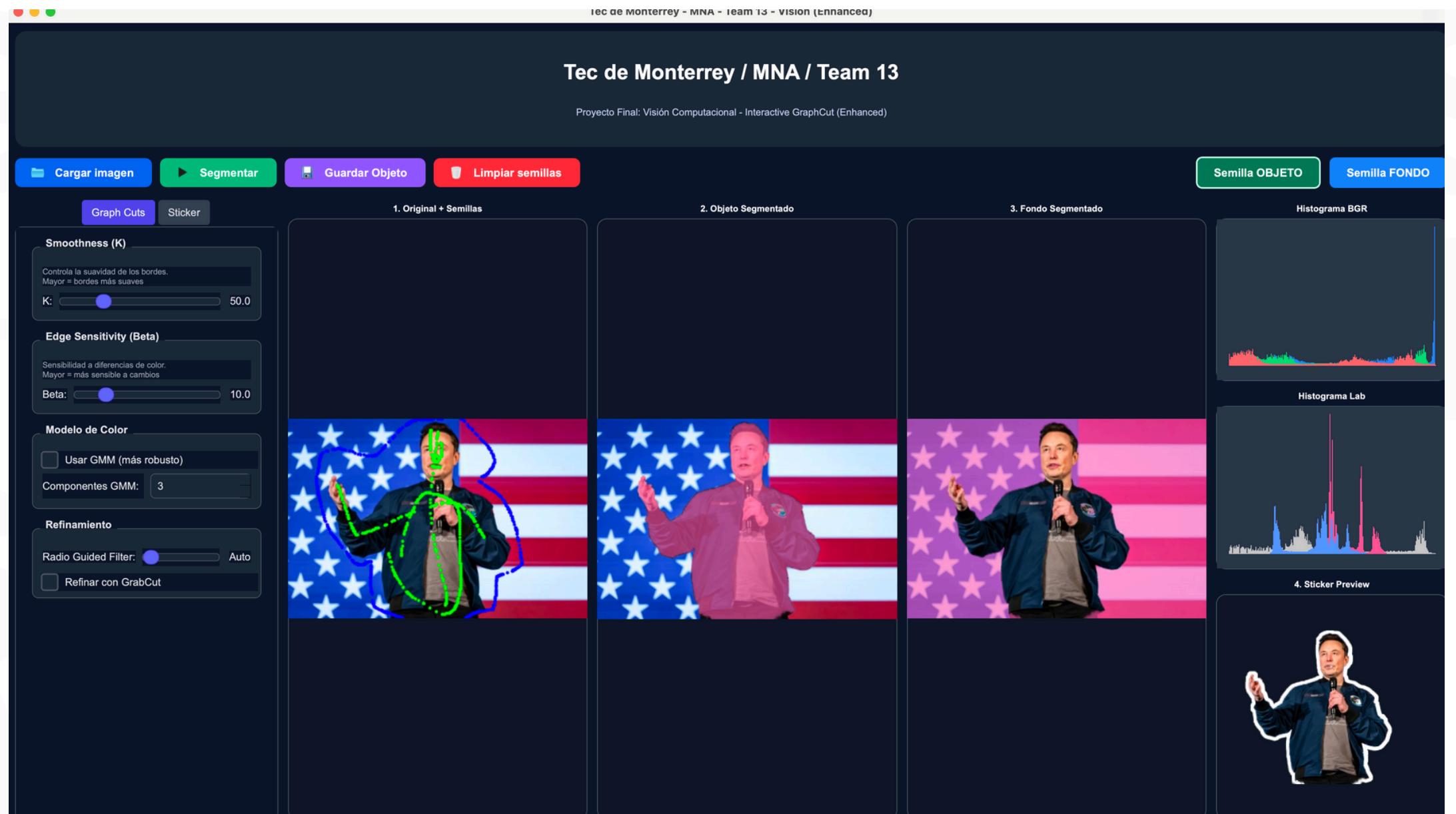
<https://colab.research.google.com/drive/1h27LiBLhfs9OkNtox-L2tZjxePLyyhRC>



Aplicación de Segmentación y Stickers

- Link Google Colab:

https://github.com/jrebull/MNA_Vision_Team13_GraphCuts



Artículo: Graph Cuts para Segmentación Óptima de Objetos: Análisis y Aplicaciones Prácticas

- Link:

https://rebull.org/visionmna/Articulo_Graph_Cuts_Segmentacion_Objetos_Analisis_Team_13.pdf

Graph Cuts para Segmentación Óptima de Objetos: Análisis y Aplicaciones Prácticas

Basado en el trabajo de Boykov & Jolly (2001)

Javier Augusto Rebull Saucedo
Maestría en Inteligencia Artificial Aplicada
Instituto Tecnológico de Monterrey
Massachusetts, USA
A01795838@tec.mx

Juan Carlos Pérez Nava
Maestría en Inteligencia Artificial Aplicada
Instituto Tecnológico de Monterrey
Mexico City, Mexico
A01795941@tec.mx

Luis Gerardo Sánchez Salazar
Maestría en Inteligencia Artificial Aplicada
Instituto Tecnológico de Monterrey
California, USA
A01232963@tec.mx

Oscar Enrique García García
Maestría en Inteligencia Artificial Aplicada
Instituto Tecnológico de Monterrey
Mexico City, Mexico
A01016093@tec.mx

Resumen—La segmentación de imágenes es un paso fundamental en visión computacional. Este artículo analiza el método *Graph Cuts*, un enfoque de optimización global que formula la segmentación como un problema de energía mínima en un grafo. Explicamos cómo la interacción del usuario mediante semillas se combina con la información de píxeles vecinos para encontrar el corte óptimo (Min-Cut) que separa el objeto del fondo. Se presentan comparaciones con métodos alternativos (K-means, Mean Shift, Region Growing, Watershed, Superpixels) y se discuten ventajas, limitaciones y aplicaciones clave en edición de imágenes, segmentación médica y creación de datasets.

Index Terms—Segmentación, Graph Cuts, Max-Flow/Min-Cut, Optimización Global, Visión Computacional

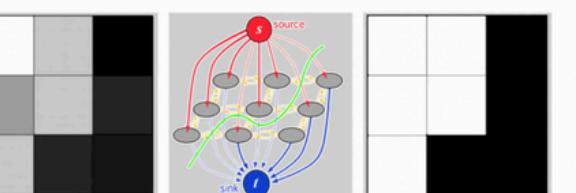


Figura 1: Transformación de imagen a grafo. Los píxeles se convierten en nodos conectados a los terminales s (objeto) y t (fondo). El corte mínimo define la segmentación.

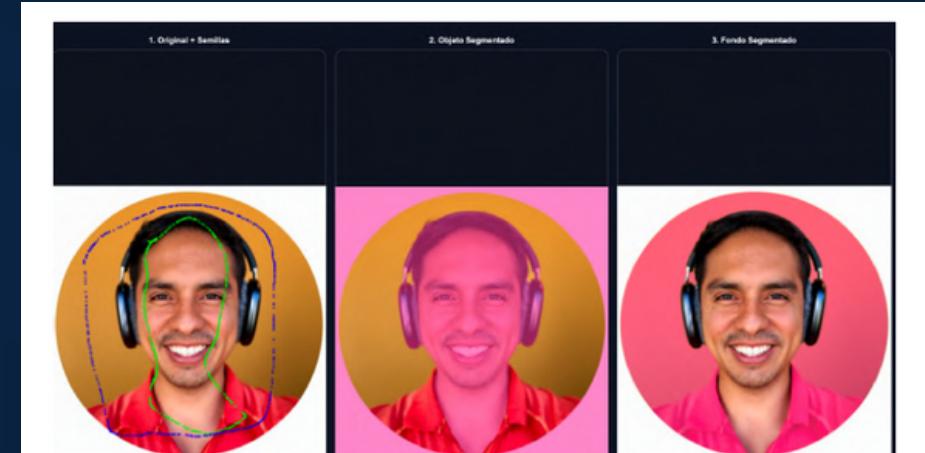


Figura 17: Pipeline retrato con fondo complejo.

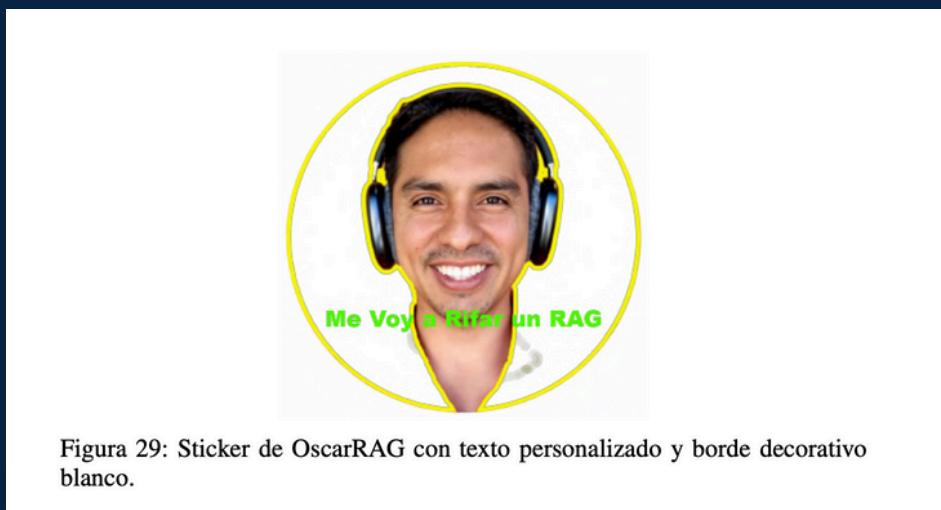


Figura 29: Sticker de OscarRAG con texto personalizado y borde decorativo blanco.

Videos de la Presentación del Artículo y de Demo Interactiva en Google Colab

Link Youtube Presentación del Algoritmo:
<https://youtu.be/6dq-iJ5XWrg>



YouTube



Link YouTube Demo Interactiva en Colab y Desktop:
<https://youtu.be/3CcJtX-EWxc>



Figura 17: Pipeline retrato con fondo complejo.



Figura 29: Sticker de OscarRAG con texto personalizado y borde decorativo blanco.

Referencias

- [1] Y. Y. Boykov and M. P. Jolly, “**Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images**,” in Proceedings Eighth IEEE International Conference on Computer Vision (ICCV), vol. 1, pp. 105–112, 2001.
- [2] Y. Boykov and V. Kolmogorov, “**An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 9, pp. 1124–1137, 2004.
- [3] C. Rother, V. Kolmogorov, and A. Blake, “**GrabCut: Interactive foreground extraction using iterated graph cuts**,” ACM Transactions on Graphics (TOG), vol. 23, no. 3, pp. 309–314, 2004.
- [4] Y. Boykov and G. Funka-Lea, “**Graph cuts and efficient N-D image segmentation**,” International Journal of Computer Vision, vol. 70, no. 2, pp. 109–131, 2006.
- [5] L. R. Ford and D. R. Fulkerson, “**Maximal flow through a network**,” Canadian Journal of Mathematics, vol. 8, pp. 399–404, 1956.
- [6] J. MacQueen, “**Some methods for classification and analysis of multivariate observations**,” in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, no. 14, pp. 281–297, 1967.
- [7] D. Comaniciu and P. Meer, “**Mean shift: A robust approach toward feature space analysis**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603–619, 2002.
- [8] R. Adams and L. Bischof, “**Seeded region growing**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, pp. 641–647, 1994.
- [9] L. Vincent and P. Soille, “**Watersheds in digital spaces: An efficient algorithm based on immersion simulations**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 6, pp. 583–598, 1991.
- [10] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “**SLIC superpixels compared to state-of-the-art superpixel methods**,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274–2282, 2012.
- [11] T.-Y. Lin et al., “**Microsoft COCO: Common objects in context**,” in European Conference on Computer Vision (ECCV), pp. 740–755, 2014.
- [12] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “**LabelMe: A database and web-based tool for image annotation**,” International Journal of Computer Vision, vol. 77, no. 1–3, pp. 157–173, 2008.

Gracias

Visión Computacional para imágenes y Video

Profesores

Dr. Gilberto Ochoa Ruiz
MIP Ma. del Refugio Melendez Alfaro
M. en C. Jose Angel Martinez Navarro