

Proyecto Procesamiento de Imágenes y Visión

Pablo Eduardo Mosquera Gutiérrez

Juan Sebastián Parrado Gutiérrez

Andrea Juliana Ruiz Gómez

4 de diciembre de 2020

1. Introducción

La segmentación de objetos sobre una imagen es una de las aplicaciones más utilizadas por usuarios de plataformas como Adobe Photoshop. Realizar este procedimiento puede tomar un tiempo proporcional al resultado deseado; no obstante, durante el desarrollo de este proyecto, se ha realizado un programa en lenguaje *Python* que buscar solventar esta problemática, ofreciendo una segmentación de objetos con resultados sobresalientes. En paralelo, se ha desarrollado una interfaz gráfica con el fin de facilitar el proceso para usuarios que desconocen este lenguaje de programación. El valor agregado de esta solución, no solo yace en la interfaz mencionada sino que dentro de esta se puede colocar la imagen segmentada sobre un *background* ingresado por el usuario.

2. Objetivo General

Realizar un algoritmo para segmentar objetos sobre una imagen mediante la aplicación del algoritmo *Grabcut*, generando una interfaz amena con el usuario.

3. Objetivos específicos

- Realizar un algoritmo para segmentar objetos sobre imágenes mediante el algoritmo de *Grabcut*, utilizando la librería OpenCV.
- Agregar un fondo a la imagen segmentada mediante la aplicación de operadores binarios.
- Generar una interfaz gráfica que facilite el uso del programa para usuarios que desconozcan el lenguaje de programación.

4. Grabcut

Consiste en la estimación de la distribución de color usando el método de mezcla Gaussiano en un área definida por el usuario con un cuadro delimitador con el fin de separar la distribución del objeto y la del fondo. Luego de la ejecución del algoritmo, el usuario tiene la opción de corregir las estimaciones realizadas y mal clasificadas señalando un punto o una región que hace parte ya sea del fondo o del objeto a segmentar y volviendo a ejecutar el algoritmo. Este proceso se puede repetir tantas veces como sea necesario para lograr la segmentación esperada por el usuario. [?]



Figura 1: GrabCut. Tomado de [?]

5. Desarrollo

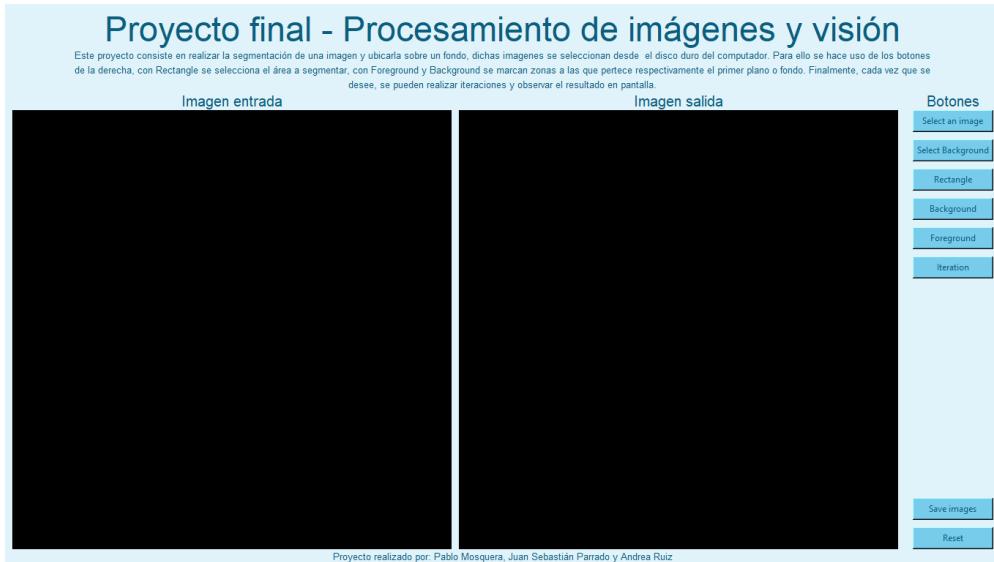


Figura 2: Interfaz Gráfica

El código desarrollado está comprendido por una interfaz gráfica (*Figura 2*), que por medio de diferentes botones se activan acciones a realizar sobre la imagen original. En la máquina de estados de la *Figura 3* se observa que los estados impares corresponden a los botones y los estados pares a las funciones consecuentes a ser presionados.

- s0: En este estado inicial se evalúa que botón de la interfaz gráfica fue presionado.

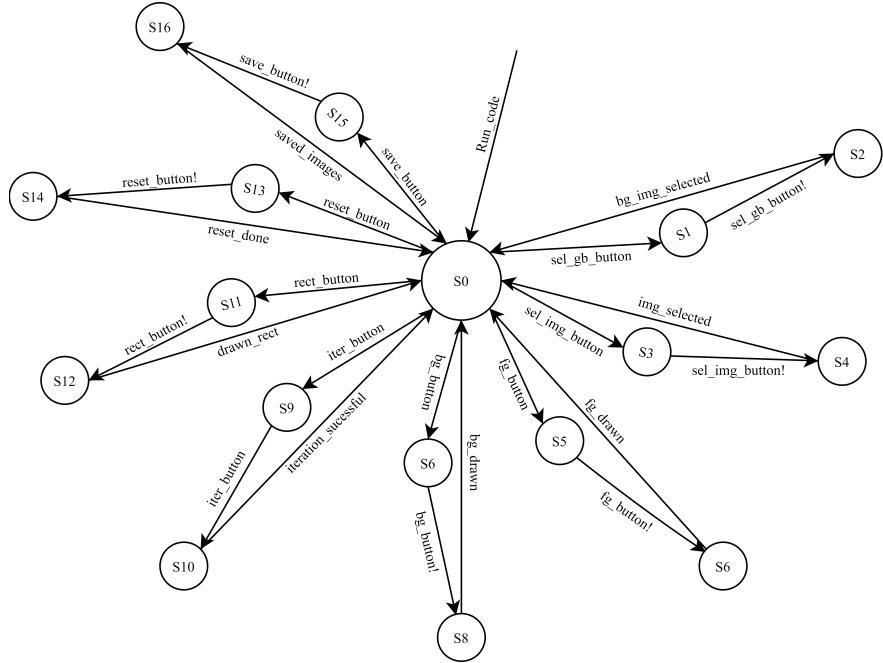


Figura 3: Máquina de estados

- s1: Cuando el botón *Select background* es presionado se dirige a la función correspondiente con el proceso de seleccionar el fondo.
- s2: Esta función abre una ventana en la cual se puede buscar desde el equipo el archivo seleccionado, y la almacena en la variable correspondiente a la imagen del fondo.

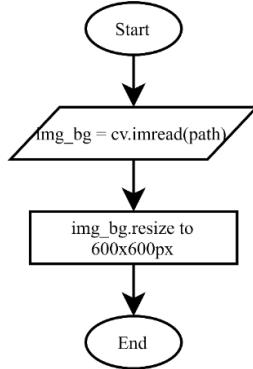


Figura 4: Diagrama de flujo del estado 2.

- s3: Cuando el botón *Select image* es presionado se dirige a la función correspondiente con el proceso de seleccionar la imagen a segmentar.
- s4: Esta función abre una ventana en la cual se puede buscar desde el equipo el archivo seleccionado, y la almacena en la variable correspondiente a la imagen original.

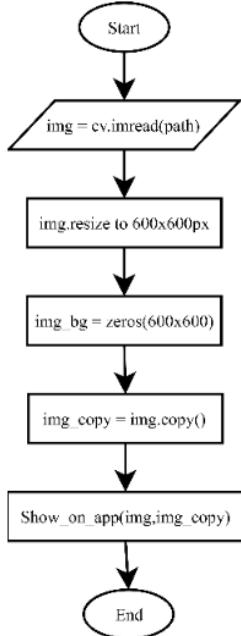


Figura 5: Diagrama de flujo del estado 4.

- s5: Cuando el botón *Foreground* es presionado se dirige a la función correspondiente con el proceso de seleccionar con el mouse el primer plano de la imagen.
- s6: Se activa una bandera, la cual habilita ciertos eventos realizados con el mouse. Apenas se presiona el click izquierdo, se empiezan a ver sobre la imagen círculos de color blanco y en la máscara se agregan en dichos posiciones el número 1. Apenas se levanta el click, la bandera de inicio se desactiva, impidiendo que se siga dibujando sobre la imagen o agregando puntos a la máscara.

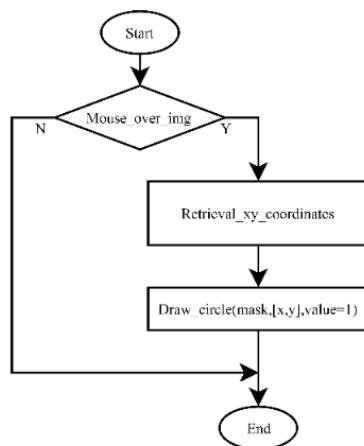


Figura 6: Diagrama de flujo del estado 6.

- s7: Cuando el botón *Background* es presionado se dirige a la función correspondiente con el proceso de seleccionar con el mouse el fondo.
- s8: Se activa una bandera, la cual habilita ciertos eventos realizados con el mouse. Apenas se presiona el click izquierdo, se empiezan a ver sobre la imagen círculos de color negro y en la máscara se agregan en dichas posiciones el número 0. Apenas se levanta el click, la bandera de inicio se desactiva, impidiendo que se siga dibujando sobre la imagen o agregando puntos a la máscara.

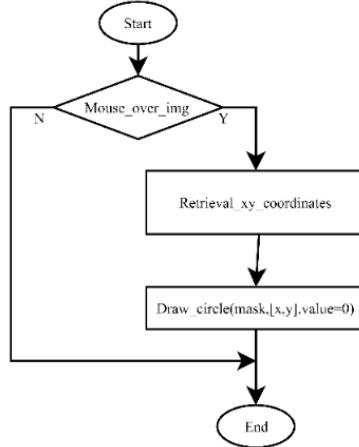


Figura 7: Diagrama de flujo del estado 8.

- s9: Cuando el botón *Iteration* es presionado se dirige a la función correspondiente con el proceso de iteración.

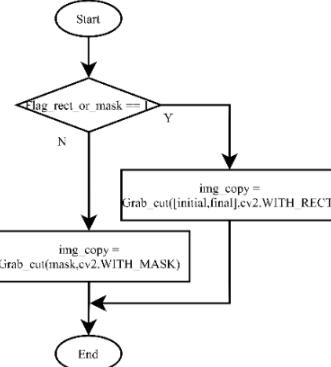


Figura 8: Diagrama de flujo del estado 10.

- s10: En esta función se implementa Grabcut, teniendo como entrada la imagen original, la máscara realizada con el rectángulo y marcas de fondo y primer plano, las variables del modelo de fondo y primer plano (array con dimensiones definidas, de uso interno de la función), el número de iteraciones (siempre 1) y el tipo de segmentación que se

desea, en este caso de las 4, escogimos la segmentación por máscara, también hay una segmentación por rectángulo, sin embargo no se implementó, puesto que el rectángulo se está agregando a la máscara como un primer plano probable. La máscara que se obtuvo como resultado de Grabcut cuenta con 4 valores: 0 fondo seguro, 1 primer plano seguro, 2 fondo posible y 3 primer plano posible. Por lo tanto, se hace una conversión, los valores 0 y 2, pasan a ser 0 y los valores 1 y 3, a ser 1. Se hace lo mismo con una máscara para el fondo, sin embargo, los valores 0 y 2, pasan a ser 1 y los valores 1 y 3, a ser 0. Finalmente, se realizan operaciones bitwise con la imagen original, con el fondo y finalmente con fondo e imagen original, en las primeras 2 se tiene en cuenta la máscara.

- s11: Cuando el botón *Rectangle* es presionado se dirige a la función correspondiente con el proceso de dibujar un rectángulo sobre la imagen con el mouse.
- s12: Se activa una bandera, la cual habilita ciertos eventos realizados con el mouse. Apenas se presiona el click izquierdo, se determina el punto inicial del rectángulo, y mientras se mueve se va observando como se dibuja. Apenas se levanta el click, se determina el punto final del rectángulo, quedando dibujado sobre la imagen en la interfaz y agregando el número 3 (posible primer plano) en la máscara en todos los puntos del rectángulo, teniendo en cuenta que es un rectángulo relleno.

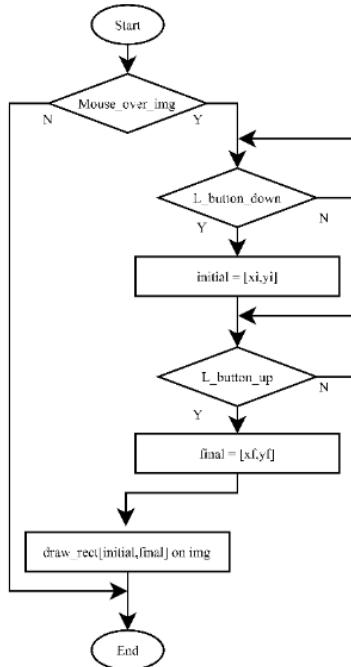


Figura 9: Diagrama de flujo del estado 12.

- s13: Cuando el botón *Reset* es presionado se dirige a la función correspondiente con el proceso de hacerle reset a las variables.
- s14: Se reinician todas las variables, para poder comenzar de nuevo con otras imágenes.

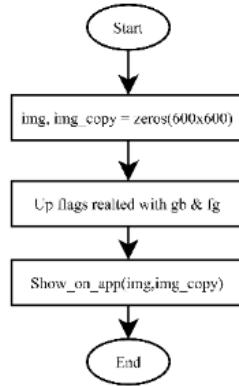


Figura 10: Diagrama de flujo del estado 14.

- s15: Cuando el botón *save images* es presionado se dirige a la función correspondiente con el proceso de guardar imágenes.
- s16: Esta función tiene como objetivo guardar una imagen con el nombre *output.png*, la cual está compuesta de izquierda a derecha por la imagen original, una barra, la imagen original con el rectángulo seleccionado y las máscaras de fondo y primer plano marcadas, una barra, la imagen segmentada, una barra, una imagen segmentada con fondo, una barra y la imagen del fondo. La barra es una imagen negra con ancho de 5 píxeles, la cual está con el fin de separar las imágenes.

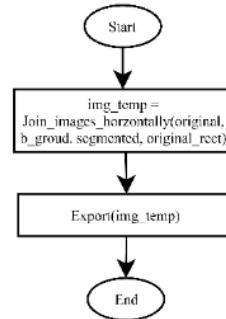


Figura 11: Diagrama de flujo del estado 16.

6. Resultados

A continuación, en las siguientes figuras se podrá observar un proceso realizado en la interfaz gráfica, en la *Figura 12* se observa la ventana que se abre al presionar los botones *select image* o *select background*, con el fin de seleccionar una imagen desde el equipo usado, en la *Figura 13* se observa el resultado de la imagen segmentada después de haber realizado el rectángulo y una iteración; en la *Figura 14* se realiza la máscara de primer plano y fondo. Finalmente, en la *Figura 15* se seleccionó un fondo, el cual se le agregó a la imagen segmentada.



Figura 12: Procedimiento realizado en la interfaz gráfica - 1.

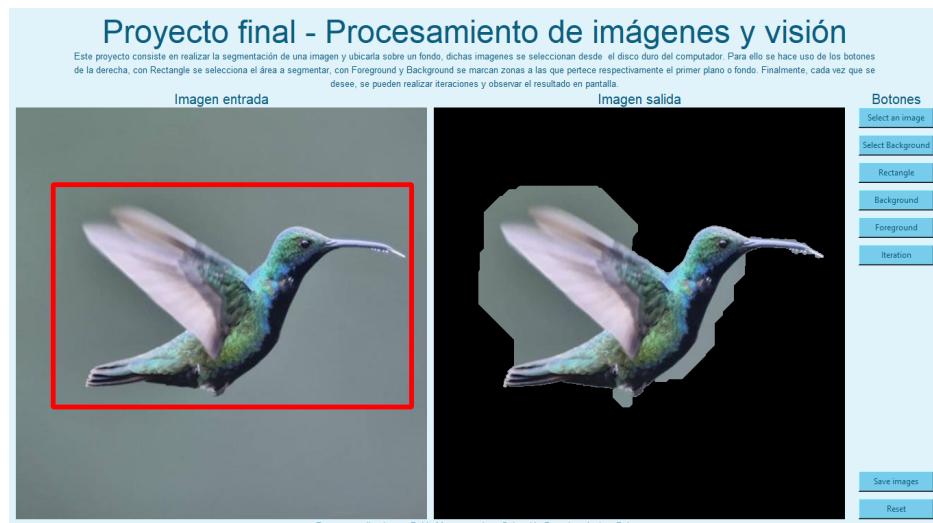


Figura 13: Procedimiento realizado en la interfaz gráfica - 2.

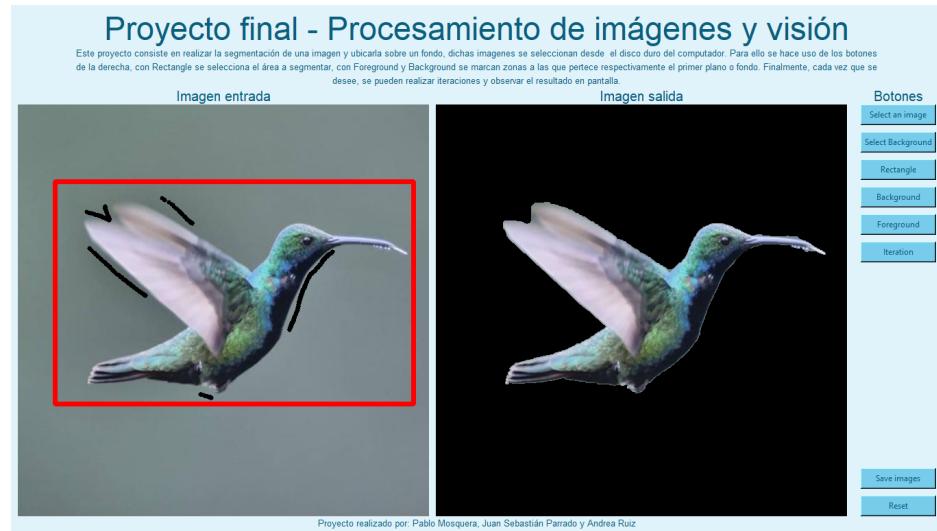


Figura 14: Procedimiento realizado en la interfaz gráfica - 3.



Figura 15: Procedimiento realizado en la interfaz gráfica - 4.

En las siguientes figuras se pueden observar los resultados obtenidos, el contenido de cada imagen de izquierda a derecha es el siguiente:

1. Imagen original.
2. Imagen original con el rectángulo realizado y las máscaras marcadas.
3. Imagen segmentada.
4. Imagen segmentada con fondo.
5. Imagen de fondo.



Figura 16: Prueba 1.



Figura 17: Prueba 2.



Figura 18: Prueba 3.



Figura 19: Prueba 4.



Figura 20: Prueba 5.



Figura 21: Prueba 6.



Figura 22: Prueba 7.



Figura 23: Prueba 8.

7. Conclusiones

Los resultados observados en las pruebas son satisfactorios. Se observa un nivel de detalle sobresaliente, especialmente en aquellas cuyo *background* tiende a ser uniforme en su tono y contrasta respecto al objeto que se desea segmentar. Es importante mencionar que cuando los resultados no son los esperados, el usuario esta en la potestad de trabajar junto al algoritmo mencionando de manera explicita donde se localizan elementos tanto del *background* como del *foreground*.

La integración de la interfaz gráfica con el algoritmo de *Grabcut* presentó una serie de problemas al no determinar las imágenes como ventanas independientes; no obstante, se logró solventar este inconveniente al determinar tamaños de imagen entrada específicos y modificando las coordenadas que requeridas por otros procesos dentro del programa.

Dentro de la documentación del algoritmo de *Grabcut* en la librería de openCV, se expiden unos valores específicos de píxel dentro de la mascara utilizada por el mencionado algoritmo. El desconocer esta información, en principio generó una serie de errores al requerir realizar iteraciones dentro del algoritmo; sin embargo, la correcta implementación de esta información

permitió generar los resultados esperados.

De manera general, este proyecto presentó un reto al grupo de trabajo, especialmente al recurrir a una programación orientada a clases. Sin embargo, el grupo determinó que es posible obtener mejores resultados si se llegase a utilizar algún método de suavizado en los bordes, evitando así el efecto de 'sierra' en los mismos.