

Universidad Autónoma Metropolitana Unidad Azcapotzalco  
División de Ciencias Básicas e Ingeniería  
Licenciatura en Ingeniería en Computación

Reporte final de proyecto terminal: Segmentación semiautomática de imágenes a partir de  
crecimiento de regiones

Adán Mateos Hernández 207330458

Trimestre 11-O

Fecha de entrega de proyecto: 6 de enero de 2012

Dr. Risto Rangel Kuoppa Titular B

Departamento de Sistemas

# INDICE

<b>Segmentación semiautomática de imágenes a partir de crecimiento de regiones.....</b>	<b>1</b>
Introducción .....	1
Módulo de carga de imagen a la interfaz de usuario .....	3
Módulo de selección libre .....	3
Módulo de selección por crecimiento de regiones .....	5
Módulo de procesamiento de la región seleccionada.....	7
Pruebas realizadas.....	9
Trabajos futuros .....	12
Conclusiones.....	14
Bibliografía.....	16

# Segmentación semiautomática de imágenes a partir de crecimiento de regiones

---

## Introducción

En el siguiente material se presenta la descripción del desarrollo de un sistema de segmentación de imágenes a partir de crecimiento de regiones. La segmentación subdivide una imagen en sus respectivas regiones u objetos que la componen. En algunos casos es necesario hacer una segmentación en imágenes muy complejas en su composición (con fondo compuesto de muchos objetos), como consecuencia surgen problemas de selección de regiones (en ocasiones compuestas de varias regiones como la fotografía de una persona). En este proyecto se resuelve el problema de segmentar regiones compuestas con fondos compuestos por regiones con poca variabilidad de colores. Para esto se aplica el método de limitar la región a segmentar y después aplicando la selección manual de puntos de regiones consideradas como fondo para, posteriormente, aplicar el método de crecimiento de regiones. A pesar de la precisión de segmentación de regiones compuestas (fotografía de una persona) el algoritmo que puede fallar en la selección de regiones consideradas como fondo y tomar parte de la región, que se desea segmentar, como parte del fondo. Esto ocurre cuando se tiene una imagen con regiones casi similares al fondo. Por ello se plantean trabajos futuros como el realce del contorno de una imagen o la eliminación de ruido antes de aplicar los métodos planteados en este proyecto. También se presentan algunas pruebas realizadas en diversas imágenes.

En la propuesta del proyecto se toma como objetivo general el siguiente:

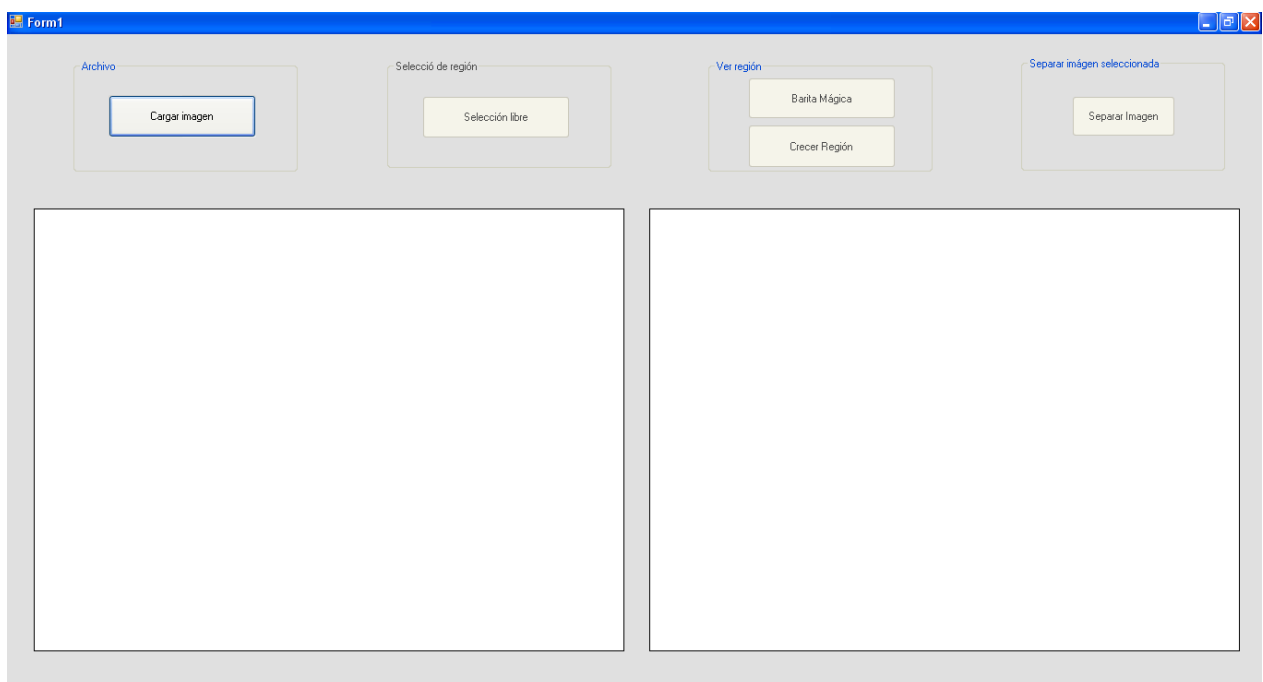
Desarrollar una aplicación que permita la segmentación semiautomática de una región específica de una imagen a partir de crecimiento de regiones.

Y en los objetivos específicos se mencionan los siguientes:

1. Diseñar una aplicación que permita visualizar la imagen y provea las herramientas necesarias (a través de una interfaz gráfica) para realizar la segmentación de dicha imagen.
2. Programar el módulo de carga de imagen a la interfaz de usuario.
3. Programar el módulo de selección libre (para selección de la región).
4. Programar el módulo de selección por crecimiento de regiones.
5. Programar el módulo de procesamiento de la región seleccionada.

El desarrollo de la aplicación se realizó en lenguaje C# bajo sistema operativo Windows.

Como primer paso se diseñó una interfaz con las herramientas necesarias para llevar a cabo la segmentación de una región específica de una imagen. Dicha interfaz (Imagen 1.1 Interfaz de usuario) está compuesta por herramientas que contienen todos los módulos que se mencionan en los objetivos específicos.



## Módulo de carga de imagen a la interfaz de usuario

La interfaz de usuario se compone por seis secciones. En la sección “archivo” se encuentra el botón de “Cargar imagen” el cual contiene el código del “Modulo de carga de imagen” del programa. En el código se utilizan funciones del lenguaje C# las cuales facilitan la programación de algunas funcionalidades del programa (como el manejo de eventos del mouse, errores, abrir ventanas de dialogo, etc.). Para cargar una imagen a la interfaz se utiliza una función que abre una ventana de dialogo en la cual se pueden visualizar los archivos de imágenes contenidas en alguna de las unidades de almacenamiento y así poder cargar alguna al *PictureBox* de la interfaz. Otra función que se utiliza en esta parte del programa es la que permite filtrar el formato de imágenes que se pueden seleccionar en la ventana de dialogo, en este caso solo permite abrir imágenes GIF y JPEG. Esto con la finalidad de eliminar errores al intentar abrir una imagen con formato no soportado por el *PictureBox*.

El tamaño de la imagen es otra de las propiedades que se tomó en cuenta ya que al momento de cargar, al *PictureBox*, una imagen de tamaño diferente de 640 X 480 pixeles ésta se acopla al tamaño del *PictureBox* (640 x 480 pixeles).

Las pruebas que se realizaron en la programación de este módulo fueron solo de soporte de formatos por el *PictureBox* especificados por el filtro, algunas de ellas se mencionan a continuación:

1. Se intentó abrir una imagen con formato diferente, pero el filtro que se utiliza no permitió mostrar imágenes con formatos diferentes en la ventana de dialogo “Abrir”, lo cual eliminó el envío de mensajes de error.
2. Se abrió la ventana de dialogo “Abrir”, pero se seleccionó la opción “Cancelar”. Esto para probar que la función *OpenFileDialog* devolviera una respuesta correcta y evitar errores.
3. Se abrió un archivo de imagen y se comprobó el buen funcionamiento, es decir, se comprobó que la imagen se cargara al *PictureBox* y se mostrara en tamaño acoplado.

## Módulo de selección libre

El código de este módulo se encuentra en la sección de “Selección de región” en el botón “Selección libre”. Se hace uso de funciones del lenguaje C# y de eventos de mouse (Mouse move, mouse down, mouse up) para el *PictureBox* que contiene la imagen original (izquierda de la interfaz). Este código solo se activa si hay una imagen cargada en el *PictureBox* y está programado de tal manera que los eventos del mouse responden en un orden que permite dibujar una línea de color negativo a cada pixel por el cual pasa el cursor del mouse, al oprimir el botón “Selección libre”

el cursor se coloca de manera automática en el centro del *PictureBox* que contiene la imagen original y se bloquea para que al momento de dibujar una línea o un punto, este se dibuje dentro del *PictureBox* evitando errores.

En este módulo las pruebas que se realizaron fueron las siguientes:

1. Se dibujaron sobre la imagen varias líneas solo para comprobar si dicha línea se dibujaba de manera continua, esto para definir de manera precisa el límite de la región seleccionada. En un principio se probó dibujando solo una secuencia de puntos, pero si el cursor se movía a una velocidad media la línea que se dibujaba era discontinua (Imagen 1.2 Línea mediante puntos). Como mejor opción se optó por dibujar una secuencia de líneas, tomando como punto de inicio, el punto final de la última línea dibujada, esto hizo que la velocidad del cursor no afectara la continuidad de dicha línea (Imagen 1.3 Línea mediante secuencia de líneas).



Imagen 1.2 Línea  
dibujada mediante  
secuencia de puntos



Imagen 1.2 Línea  
dibujada mediante  
secuencia de líneas

2. Se probó la cerradura automática de la selección, es decir, si el punto inicial y el punto final no coincidían al final de la selección había una apertura en la selección. Por esta razón se programó el dibujo automático de una línea que cierra la selección libre (Imagen 1.3 Cerrado automático de la selección). De esta manera se facilita el bosquejo de regiones de una imagen limitando de manera más precisa y sin dejar aperturas en la selección libre que pudieran afectar al aplicar la selección por crecimiento de regiones.



Imagen 1.3 Cerrado automático de la selección

3. Debido a que las imágenes tienen una variedad de colores, es muy difícil seleccionar un color de línea que se distinga. Por esta razón el color de la línea es el negativo de cada pixel por el cual pasa el cursor del mouse al momento de dibujarla (Imagen 1.4 Línea con color negativo al píxel de la imagen). Para obtener el negativo de cada pixel de la imagen se realiza la siguiente operación.
  - a. Se obtiene el pixel en formato RGB en variables tipo entero con valores de 0 a 255.
  - b. Se realiza la siguientes resta:
    - i.  $255 - \text{Valor en Rojo del pixel}$
    - ii.  $255 - \text{Valor en Verde del pixel}$
    - iii.  $255 - \text{Valor en Azul del pixel}$
  - c. Se almacena el nuevo color en una variable tipo Color y se utiliza como color de la línea de selección libre.

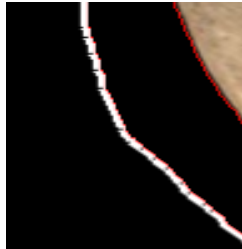


Imagen 1.4 Línea con color negativo al píxel de la imagen

4. Por último se probó el dibujo de líneas en imágenes con gran variedad de colores.

### Módulo de selección por crecimiento de regiones

Este módulo se encuentra en la sección de “Ver región”. El código está dividido en dos partes: El código del botón “Barita mágica” y el botón “Crecer región”.

#### Botón “Barita mágica”

Activa una función para dibujar un punto en la imagen, al oprimir este botón el cursor del mouse se coloca en el centro de la imagen contenida en el *PictureBox*. Se activa una función que permite, dando clic en alguna parte de la imagen, seleccionar un pixel y dibujar un punto de color rojo. Esto con la finalidad de que el usuario pueda seleccionar varios puntos pertenecientes al fondo dentro de la región bosquejada con “Selección libre” (Imagen 1.5 Pixeles pertenecientes al fondo de la región bosquejada). Al momento de seleccionar un pixel, este se marca con una bandera en una matriz tipo booleana (indicando que pertenece al fondo) y se obtienen las propiedades de este en formato RGB lo que permite obtener, mediante cálculos de promedios, la intensidad de cada color (*Red, Green and Blue* por sus siglas en ingles) y así reconocer todo pixel que pertenece al fondo indicado por el usuario. Estos puntos se conocen como “Semillas” en el algoritmo de crecimiento de regiones

y son básicos en los cálculos de segmentación de imágenes al momento de seleccionar los pixeles vecinos de las “Semillas”.

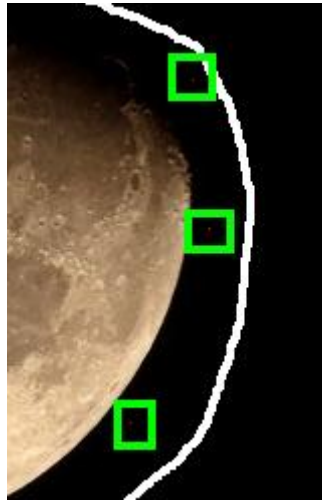


Imagen 1.5 Pixeles pertenecientes al fondo de la región bosquejada, señalados con los cuadros verdes.

Las pruebas realizadas para esta primera parte del módulo selección por crecimiento de regiones fueron las siguientes:

1. Se seleccionó un punto del fondo de la región y se comprobó el color de este (que fuera un punto rojo).
2. Se seleccionaron varios puntos dentro de la región bosquejada y nuevamente se comprobó si se dibujaban puntos rojos, además se comprobó que el botón “Creecer región” se habilitara.

#### Botón “Creecer región”

El código de este botón está compuesto por una función que, tomando como punto de partida el “punto semilla” seleccionado con la “Barita mágica”, selecciona los pixeles que pertenecen al fondo de la región bosquejada con “Selección libre” y los pone en color rojo, solo para mostrar la selección. Cada pixel seleccionado como fondo, por el algoritmo, es señalado con una bandera en la matriz tipo booleana, de esta manera la región a segmentar queda libre de pixeles con propiedades de fondo.

Las pruebas que se realizaron para la segunda parte de este módulo fueron las siguientes:

1. Para determinar el valor del error al comparar las propiedades del pixel semilla y el pixel vecino se realizaron varias pruebas en distintas imágenes. Con el valor del error elegido se comprobó que al comparar un pixel del fondo con uno de la región a segmentar, el algoritmo reconocía perfectamente el fondo de dicha región bosquejada (Imagen 1.6 Reconocimiento del fondo de la región a segmentar y puntos con propiedades distintas al fondo).



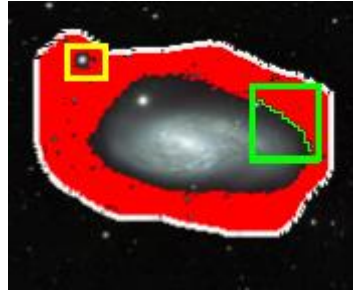


Imagen 1.6 Reconocimiento del fondo de la región a segmentar y puntos con propiedades distintas al fondo

En la imagen 1.6 se muestra los píxeles (en color rojo) que pertenecen al fondo, en el recuadro amarillo uno de los puntos o regiones que no tienen propiedades de fondo, en el recuadro verde se muestra como se crea un límite (línea verde, solo para indicar) entre el fondo de la región bosquejada (línea blanca o de color negativo del fondo) y la región a segmentar (una galaxia en este caso).

2. Se realizaron pruebas en imágenes con tonos similares a la región considerada como fondo, en este caso el algoritmo tomó píxeles de la región a segmentar como parte del fondo (Imagen 1.7 Error de algoritmo con imágenes con tonos similares al fondo). Se probó que una posible solución es seleccionar semillas con propiedades totalmente distintas a la región a segmentar o tomar otras propiedades de los píxeles como el tono de gris.



Imagen 1.7 Error de algoritmo con imágenes con tonos similares al fondo

### Módulo de procesamiento de la región seleccionada

El código que compone este módulo se encuentra en la sección “Separar imagen seleccionada” en el botón “Separar imagen”. El algoritmo de crecimiento de regiones llena una matriz tipo booleana que indica los píxeles que pertenecen al fondo, el algoritmo que se utiliza para separar la imagen a segmentar es el de crecimiento de regiones pero sin tomar en cuenta las propiedades del píxel. En este caso solo se seleccionan los píxeles que no están marcados con la bandera en la matriz booleana

y se copian al *PictureBox* de la derecha de la interfaz. De igual manera, al momento de oprimir el botón “Separar imagen”, el cursor del mouse se coloca en el centro del *PictureBox* que contiene la imagen original y se bloquea hasta oprimir en alguna parte de la región a segmentar.

En este módulo solo se realizó una prueba, ya que solo fue necesario comprobar si cada pixel se copiaba en la posición correcta en el otro *PictureBox* (Imagen 1.8 Muestra de pixeles copiados en la posición correcta).

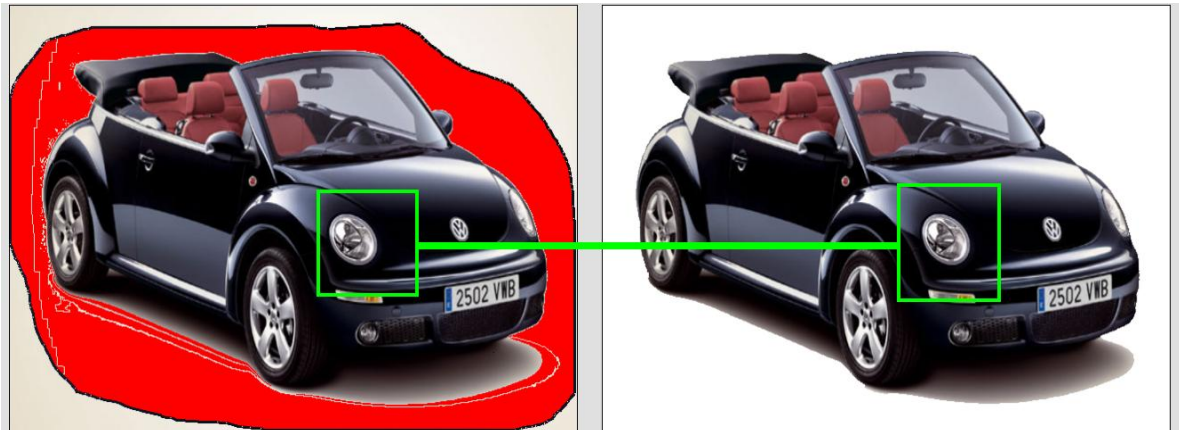
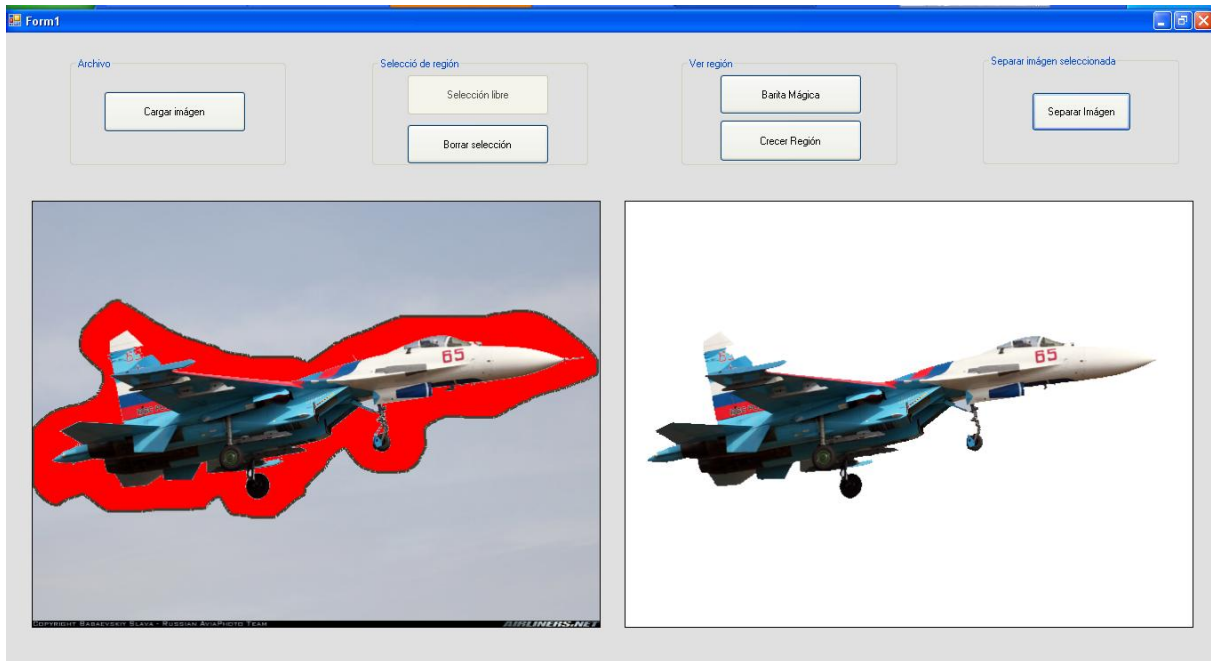
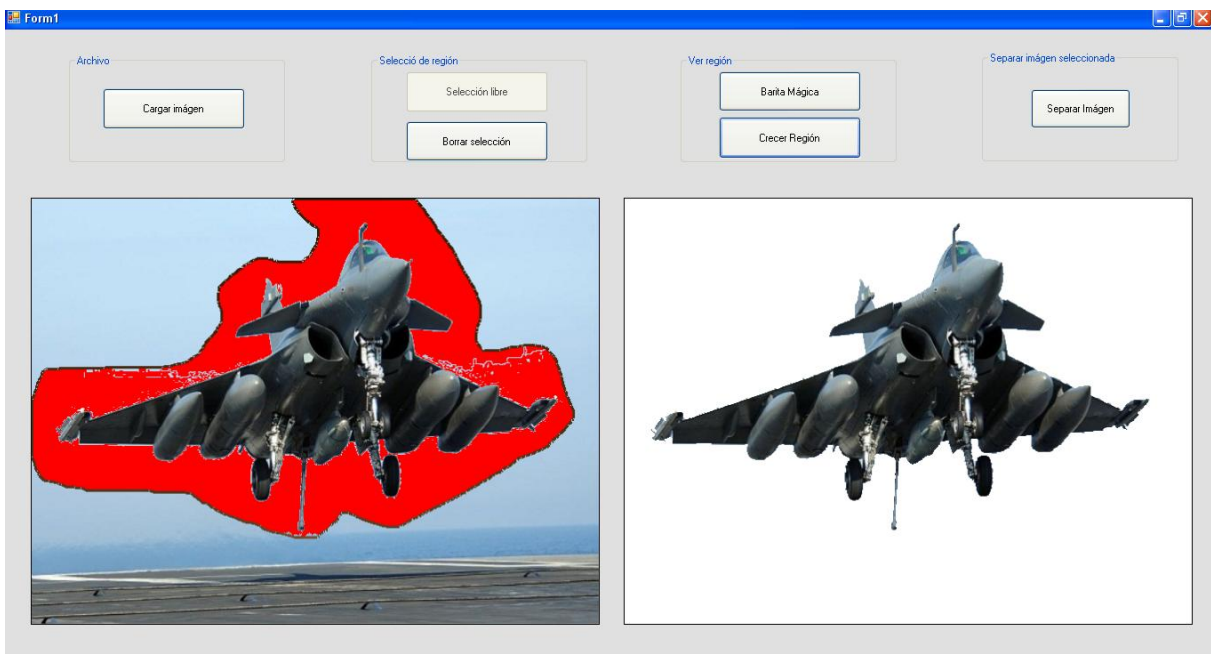


Imagen 1.8 Muestra de pixeles copiados en la posición correcta

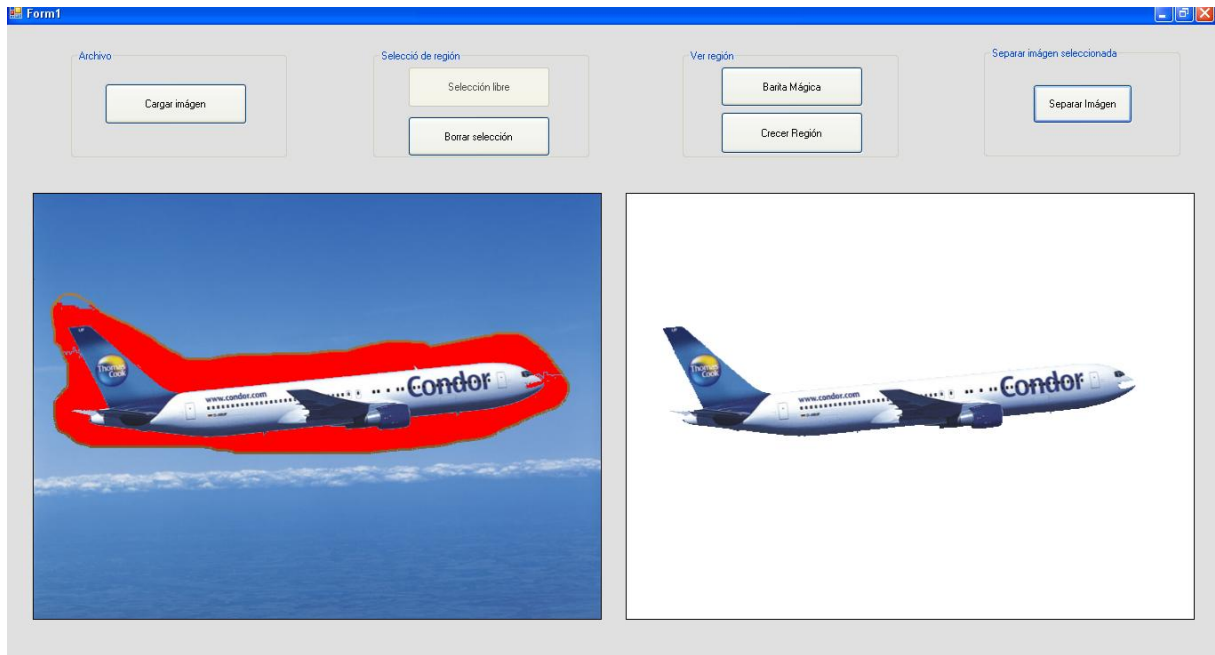
## Pruebas realizadas



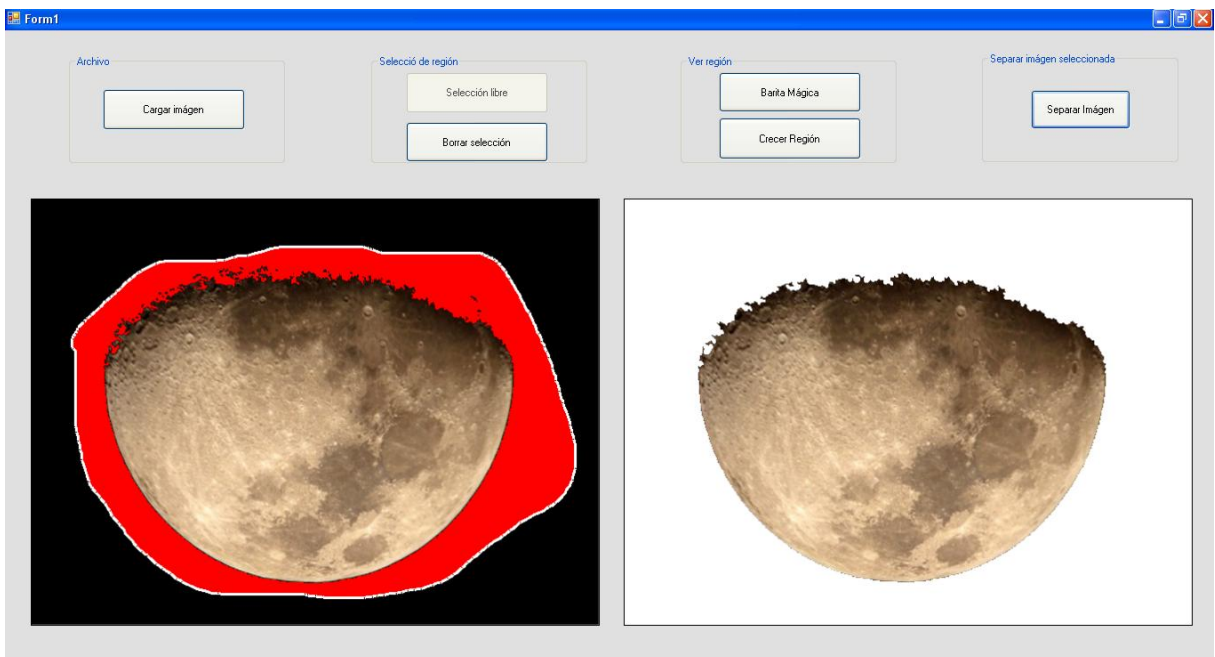
Prueba 1 Segmentación de una región compuesta por una gran variedad de colores con un fondo tenue. Se seleccionó solo una semilla del fondo.



Prueba 2. Segmentación de región con fondo en dos tonos distintos formando dos regiones diferentes. Se seleccionaron dos semillas pertenecientes al fondo.



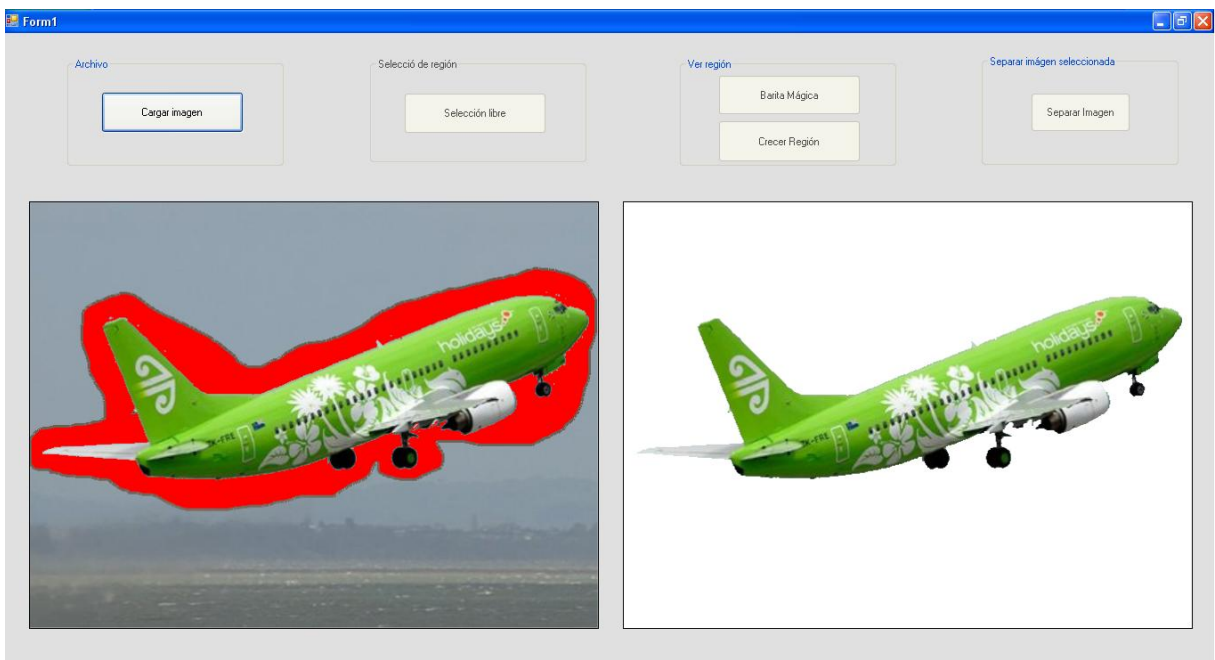
Prueba 3. Segmentación de una región completa con variedad de colores y un fondo con dos regiones compuestas en tonos diferentes de azul. Se utilizaron dos semillas.



Prueba 4. Segmentación de una región con variantes de color pero con fondo negro. Solo se seleccionó un pixel semilla.



Prueba 5. Segmentación de una región con gran variedad de colores y fondo compuesto por distintas regiones que no se distinguen a simple vista. Se utilizaron cinco semillas para hacer un mejor recorte de la imagen requerida.



Prueba 6. Segmentación de región completa con variación de colores y con fondo poco variante de colores. Se utilizó una semilla para dicha selección.

## Trabajos futuros

En el algoritmo que se utilizó para la selección de las regiones consideradas como fondo se aplicaron dos métodos para seleccionar cada pixel de acuerdo al pixel semilla.

1. Se obtiene un pixel semilla en formato RGB y se seleccionan sus ocho vecinos, extrayendo a cada uno en formato RGB.

Vecino 1	Vecino 2	Vecino 3
Vecino 8	Pixel semilla	Vecino 4
Vecino 7	Vecino 6	Vecino 5

Al obtener la cantidad de Rojo, Verde y Azul (*Red, Green and Blue* por sus siglas en ingles) en valores de 0 a 255 se obtiene el promedio de los 9 pixeles incluyendo el pixel semilla para cada color.

Después, para seleccionar los vecinos de cada uno de los 8 pixeles vecinos del pixel semilla, se obtiene el pixel en formato RGB y se aplica la misma técnica de obtener el promedio de cada uno de los colores para cada vecino, si, al comparar dicho promedio con el promedio de la suma de los nueve pixeles muestra, la diferencia es menor o igual a cierto error definido manualmente, Se realiza la prueba con el segundo método.

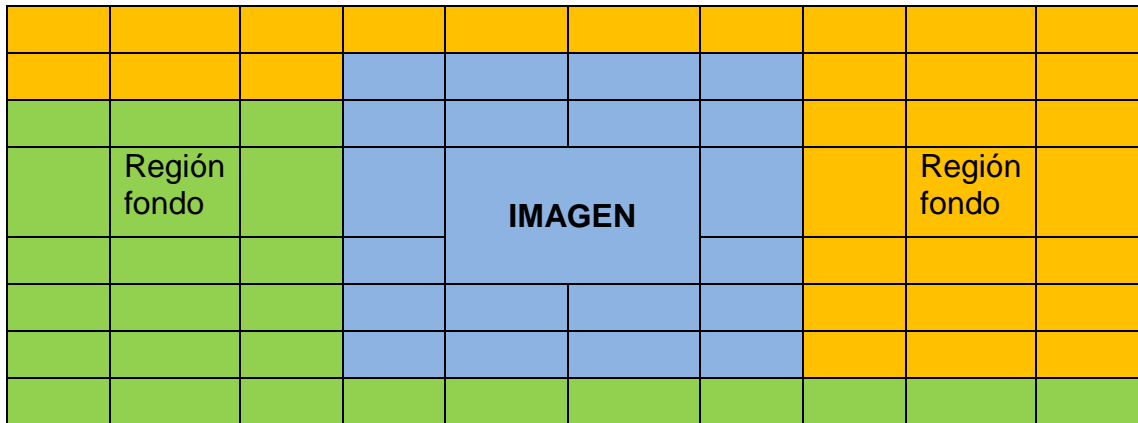
Vecino 1.1	Vecino 1.2	Vecino 1.3	
Vecino 1.8	Vecino 1	Vecino 2	Vecino 3
Vecino 1.7	Vecino 8	Pixel semilla	Vecino 4
	Vecino 7	Vecino 6	Vecino 5

Los vecinos de Azul son considerados vecinos del pixel "Vecino 1" pero ya han sido revisados y por consecuencia son descartados. Si el pixel Vecino 1.1 cumple con la condición anterior, se realiza la prueba con el segundo método.

2. El segundo método es obtener un promedio de tono de gris del conjunto de pixeles muestra, es decir, se obtiene el promedio de la suma de los 9 pixeles vecinos sumando el promedio, obtenido con el método 1, de los tres colores (RGB).

Nuevamente, para la selección de un pixel con propiedades similares al perteneciente a la región considerada como fondo, se aplica una comparación

de promedios y si el error es menor igual al definido manualmente y, además, cumple con la condición del método uno, entonces dicho pixel se agrega al conjunto de pixeles considerados como fondo.



En dicha imagen los pixeles que cumplen con las dos condiciones de los dos métodos, al seleccionar un pixel semilla en las regiones de fondo, son los pixeles de color naranja y verde.

Dichos métodos funcionan perfectamente si la región considerada como fondo se distingue totalmente de la región a segmentar. En imágenes muy tenues o con poca variabilidad de colores entre la región considerada como fondo y la región a segmentar dicho algoritmo puede fallar y seleccionar pixeles o regiones grandes de la región a segmentar (Imagen 2.1 Error de algoritmo). Por esta razón se plantea, como trabajo futuro, mejorar el algoritmo aplicando otros métodos o complementando los que ya se tienen. Como por ejemplo eliminar el ruido antes de empezar la selección de las regiones del fondo. Aplicar el realce de borde de imágenes y posteriormente aplicar los métodos de esta aplicación.

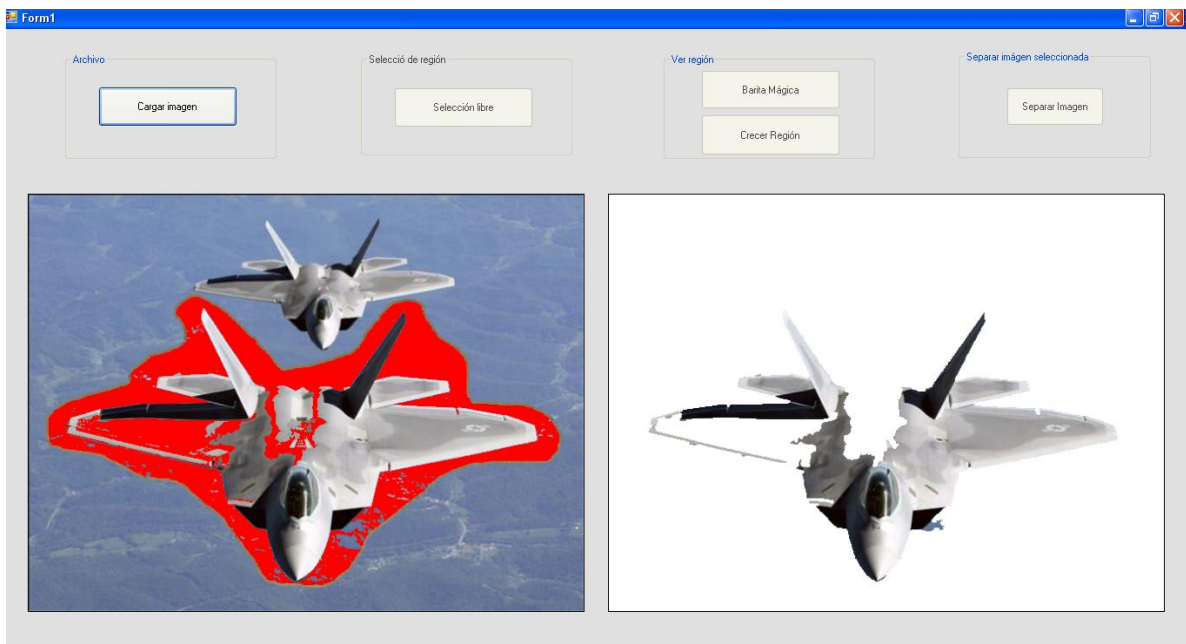


Imagen 2.1 Error de algoritmo

## Conclusiones

La segmentación de imágenes es un problema de procesamiento de imágenes que aún no se ha podido optimizar cuando se trata de imágenes tenues o se desea segmentar regiones con colores similares al fondo pero distinguible por el ojo humano.

El tema de segmentación de imágenes tiene varias aplicaciones, una de ellas está en el diseño gráfico en donde se requiere de la modificación de fotografías como mejorar la presentación.

Los actuales programas como *Photoshop*, *Gimp* y otros programas que incluyen la herramienta de segmentación o selección por crecimiento de regiones no pueden segmentar regiones completas, para hacerlo, el usuario tiene que borrar manualmente regiones consideradas como fondo lo cual es una tarea poco agradable y que requiere de mucho tiempo. Por ello se propuso este proyecto de “Segmentación semiautomática de imágenes a partir de crecimiento de regiones”, el cual permite realizar segmentación de regiones deseadas por el usuario en cinco pasos muy sencillos (cargar imagen, selección libre, selección de semillas, crecimiento de regiones y separar imagen). Dicho algoritmo puede proponerse como un *Plug-in* para cualquier programa que tenga esta herramienta de segmentación.

Para realizar la segmentación en esta aplicación se consideró tener primero un bosquejo de la región que se desea segmentar, dicho bosquejo se realiza con un color diferente al de la imagen y debido a que las imágenes tienen una gran variedad de colores se plantearon varios métodos para la selección del color del lápiz para dibujar la línea, como por ejemplo obtener el color que menos o que no se utiliza en la imagen, lo más viable fue tomar el negativo de cada pixel por el cual pasa el cursor del mouse al dibujar la línea alrededor de la región.

Otro de los temas importantes es que ésta aplicación puede complementarse con varios algoritmos para que pueda ser más completa, ya que al realizar pruebas con varias imágenes se comprobó que si se elimina el ruido que hay en algunas regiones de la imagen se obtiene una mejor segmentación. En imágenes con bordes desvanecidos y con colores similares al fondo se tienen errores de selección, ya que, al momento de crecer la región, se selecciona parte de la región que se desea segmentar.

Las pruebas realizadas en esta aplicación desenvuelve una serie de problemas como: Mejorar la calidad de la segmentación en imágenes donde se tiene un fondo muy variado (árboles y montañas por ejemplo), ya que en la aplicación planteada se requieren de varias semillas, lo cual no puede ser funcional.



Por otro lado es importante mencionar que la aplicación solo muestra el funcionamiento del algoritmo y no guarda la imagen segmentada para que pueda usarse, este módulo se plantea como un trabajo futuro.

## Bibliografía

Rafael C. Gonzalez & Richard E. Woods (2008). Digital Image Processing, third edition. Upper Saddle River, New Jersey: Prentice Hall.

*Region Growing*, Mayo 2011. [En línea]. Disponible:  
[http://en.wikipedia.org/wiki/Region\\_growing](http://en.wikipedia.org/wiki/Region_growing)

*Segmentation (Image processing)*, Mayo 2011. [En línea]. Disponible  
[http://en.wikipedia.org/wiki/Segmentation\\_%28image\\_processing%29](http://en.wikipedia.org/wiki/Segmentation_%28image_processing%29)

*Pac-n-zoom aplicacion*. Mayo 2011. [En línea]. Disponible: <http://www.pac-n-zoom.com/>

*TurtleSeg aplicacion*. Mayo 2011. [En línea]. Disponible: <http://www.turtleseg.org/>

*Gimp GNU manipulation program*. May 2011. [En línea]. Disponible: <http://www.gimp.org/>

*ITK Insight Toolkit*. Mayo 2011. [En línea]. Disponible: <http://www.itk.org/>

*ITK-SNAP*. Mayo 2011. [En línea]. Disponible: <http://www.itksnap.org/pmwiki/pmwiki.php>

*C++ Libraries for Computer Vision Research and Implementation*. Mayo 2011. [En línea].  
Disponible: <http://vxl.sourceforge.net/>

*Convert, edit and compose images*. Mayo 2011. [En línea]. Disponible:  
<http://www.imagemagick.org/script/index.php>

*Slicer*. Mayo 2011. [En línea]. Disponible: <http://www.slicer.org/pages/Introduction>

*The Medical Imaging Interaction Toolkit (MITK)*. Mayo 2011. [En línea]. Disponible:  
<http://www.mitk.org/wiki>

*OpenCV*. Mayo 2011. [En línea]. Disponible: <http://opencv.willowgarage.com/wiki/>

*Software Grass GIS*. Junio 2011. [En línea]. Disponible: <http://grass.fbk.eu/>

*Fiji (software)*. Junio 2011. [En línea]. Disponible:  
[http://en.wikipedia.org/wiki/FIJI\\_%28software%29](http://en.wikipedia.org/wiki/FIJI_%28software%29)

*Aforge.NET*. Junio 2011. [En línea]. Disponible: <http://www.aforgenet.com/>

*Introducing GemIdent*. Junio 2011. [En línea]. Disponible: <http://gemident.com/>

*CVIP Tools, A software package for the exploration of computer vision and image processing*. Junio 2011. [En línea]. Disponible: <http://www.ee.siue.edu/CVIPtools/>

*Mega Wave, What is Mega Wave2*. Junio 2011. [En línea]. Disponible: <http://megawave.cmla.ens-cachan.fr/>

Pablo Juárez Santillán, “Reconocimiento de expresiones faciales mediante procesamiento de imágenes”. Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, DF, México 2007.

Fausto Mario Díaz Cabrera, “Clasificador de objetos en banda infinita por medio de procesamiento de imágenes”. Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, DF, México 2008.

Arturo Pacheco Pineda, “Procesamiento de imágenes utilizando mapas auto-organizados”. Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, DF, México 2010.

Daniel López Martínez, “Captura de la placa de un coche mediante la cámara de un dispositivo móvil y procesamiento de la imagen para reconocimiento de la matrícula”. Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, DF, México 2010.

Marco Emmanuel García Avilés, “Software para validación antropométrica por procesamiento digital de imágenes”. Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, DF, México 2010.

Carlos A. Vega Quiroz, “Reconstrucción 3D de objetos reales mediante adquisición calibrada de su contorno”. Proyecto terminal de Ingeniería en Computación, Universidad Autónoma Metropolitana Azcapotzalco, DF, México 2010.