

# Segmentación de imágenes por medio de la función Bitand

David Alejandro Ortíz Marín – Código N° 20062005028\*

3 de Abril de 2012

## Resumen

El presente documento muestra un primer acercamiento al proceso de segmentación de imágenes, por medio de la función Bitand() en Matlab, la cual, dependiendo de la comparación lógica que se haga (de 00000000 a 11111111) se tienen resultados y tonalidades diferentes en la imagen procesada.

**Palabras clave:** Imagen, Formato de Imagen, RGB, Escala de Grises, Función uint8, Función Bitand.

---

\*Estudiante proyecto curricular de Ingeniería electrónica de la faculta de Ingeniería de la Universidad Distrital Francisco José de Caldas. Correo electrónico [daortizm@correo.udistrital.edu.co](mailto:daortizm@correo.udistrital.edu.co)

## 1. Introducción

Una imagen puede tener diferentes formatos de representación, dicha representación posee la codificación de la información de cada uno de los píxeles que la conforman, para el caso Matlab posee varios formatos de representación para los datos al igual que cualquier programa de programación, dichos datos se representan en forma binaria, entre mas bits tenga la cadena de datos, la resolución del píxel es mayor pero se aumenta el tamaño en disco de cada píxel y en general de la imagen es su totalidad.

A la hora del tratamiento de una imagen, y considerando sus características de formato, hay diferentes áreas de alto interés, zonas de alta o baja luminosidad en las cuales se pueden encontrar bordes o regiones de separación, extraer objetos o formas características o simplemente resaltar algunos detalles propios o requeridos de la imagen.

Debido a la representación en bits, al preservar la parte alta de una imagen se resaltan los brillos, en cambio al preservar la parte baja de la cadena de bits se resaltan los tonos oscuros, con esto se pueden segmentar zonas de la imagen como los bordes de las figuras, con lo que se obtienen zonas divisorias de ésta las cuales pueden tener gran valor para quien realice el procesamiento de la imagen.

Para el caso de interés de este documento se trabajó una imagen RGB en formato Uint8 el cual es una cadena de 8 bits y se manipularon estos con la función Bitand; el método utilizado para solucionar el problema, los resultados obtenidos y las conclusiones arrojadas, serán discutidos a lo largo de los siguientes incisos.

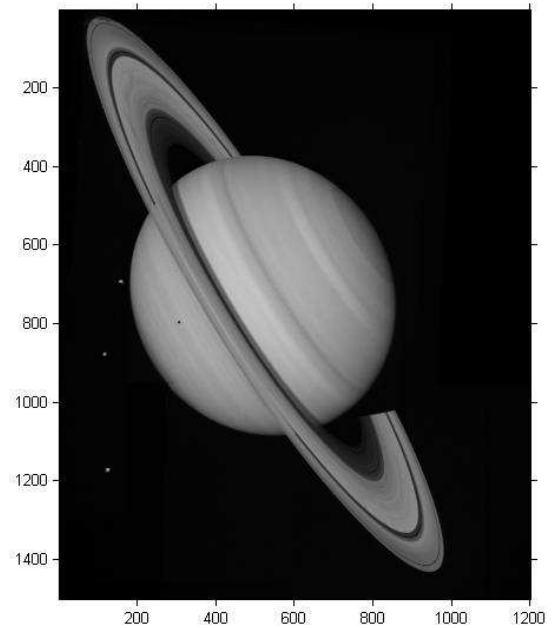
## 2. Formulación del Problema

Con una imagen de cualquier formato, de color o blanco y negro, aplicar la función Bitand con el fin de obtener 8 imágenes representativas con 1 bit activo para cada una (00000001 a 10000000).

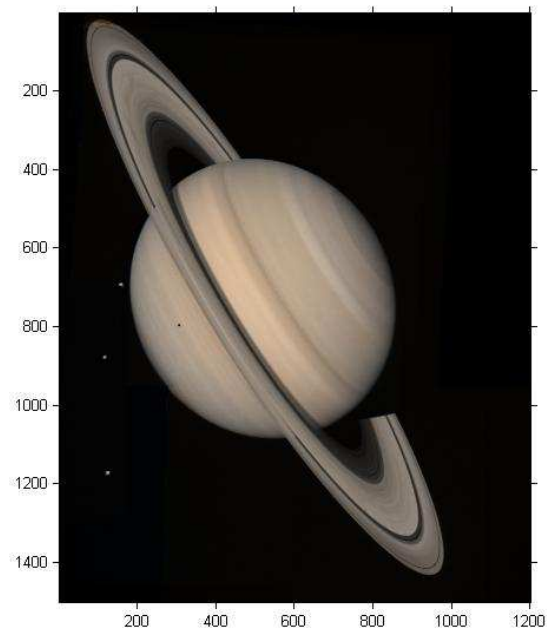
## 3. Solución del Problema

Para este problema se usó la imagen Saturn de las muestras que tiene Matlab, esta imagen esta en formato Uint8 y es una imagen en color (RGB), se usó la función Bitand la cual hace la función lógica and bit a bit a una imagen en formato uint8 para cada píxel. La imagen se trabajo en color y en escala de grises para poder contrastar los resultados y obtener diferencias relevantes a la hora de su segmentación.

La figura 1 y 2 representan las imágenes en escala de grises y en colores respectivamente a las cuales se les aplicó las máscaras descritas y con las cuales se obtuvieron datos relevantes acerca del comportamiento de la imagen dependiendo de la comparación lógica que se haga.

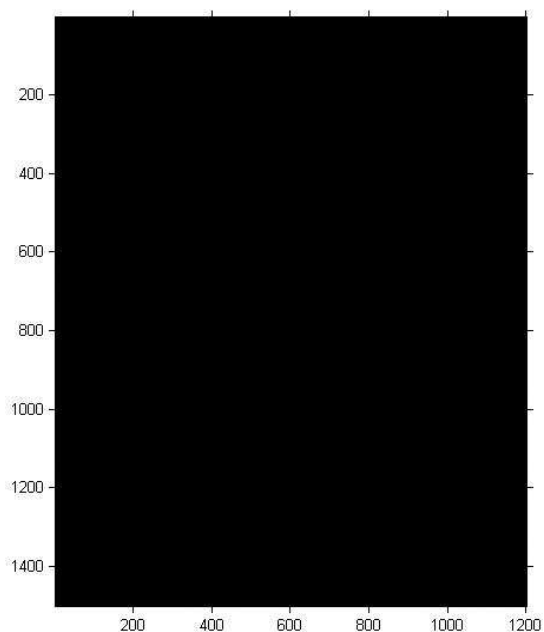


**Figura 1:** Imagen en escala de grises de la imagen original y formato uint8.

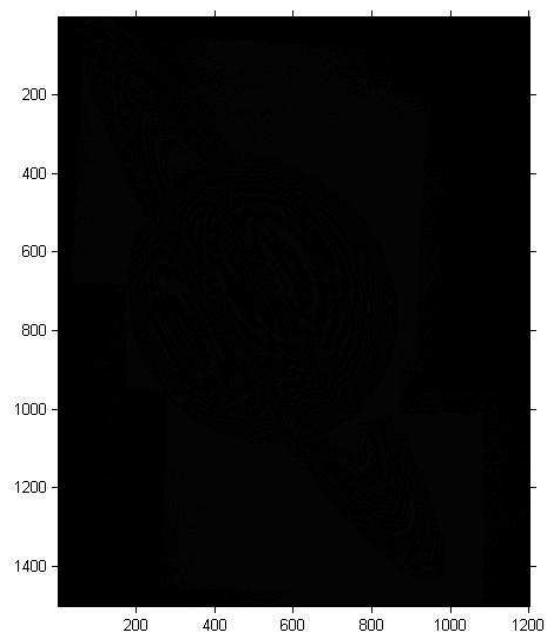


**Figura 4:** Imagen original de color (RGB) y formato uint8.

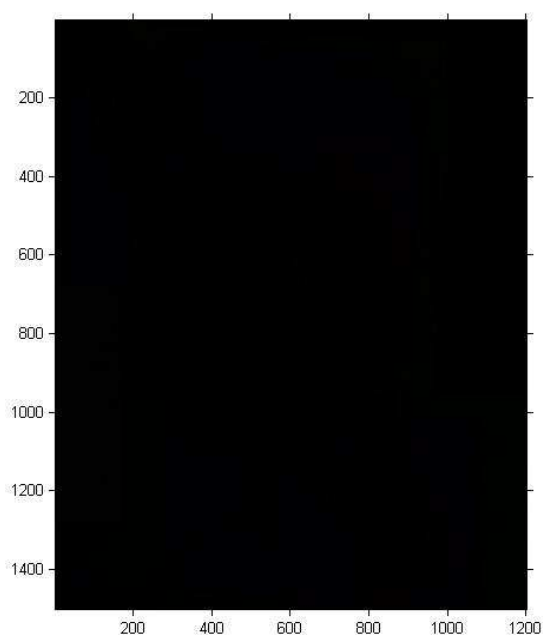
Para las máscaras 00000001 a 00000100 se tienen imágenes con alta presencia de tonos negros, debido a que los bits activos son los de la parte baja de la máscara, por lo cual se resaltan los negros, de la figura 3 a la figura 4 se muestra dicho comportamiento.



**Figura 3:** Imagen en escala de grises con argumento de entrada de Bitand en 00000001 (1 decimal).

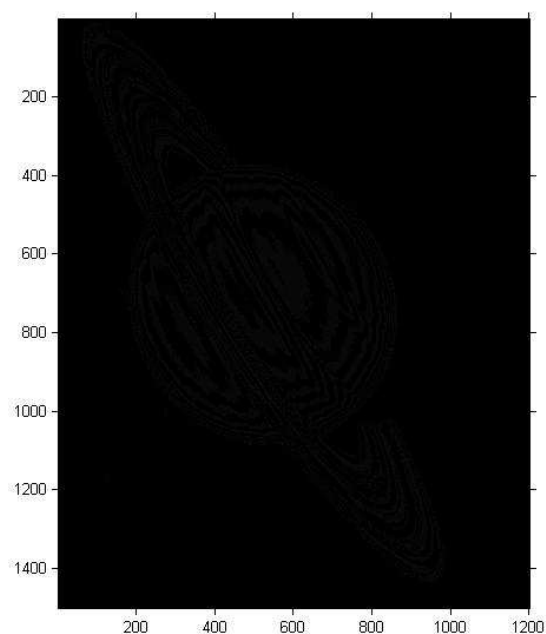


**Figura 5:** Imagen en escala de grises con argumento de entrada de Bitand en 00000100 (4 en decimal).

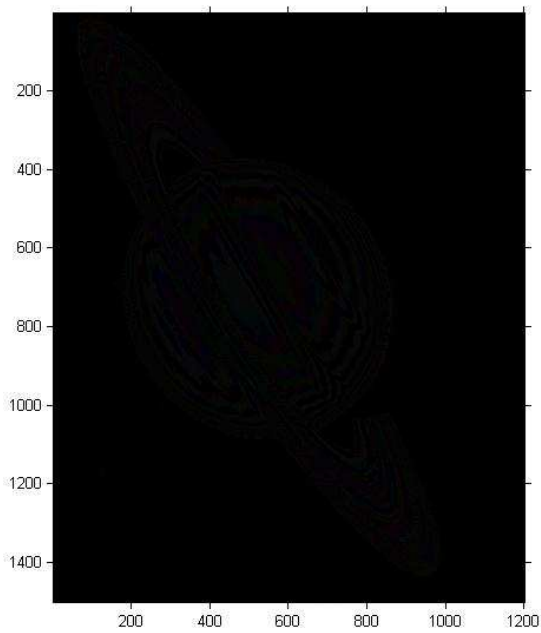


**Figura 4:** Imagen RGB con argumento de entrada de Bitand en 00000010 (2 decimal).

Para la figura 6 y 7 (escala de grises y RGB respectivamente) se le aplica una máscara 00001000, en las dos imágenes se empieza a ver el contorno de la figura y ciertas partes luminosas de esta, pero se resaltan más en la imagen en escala de grises.

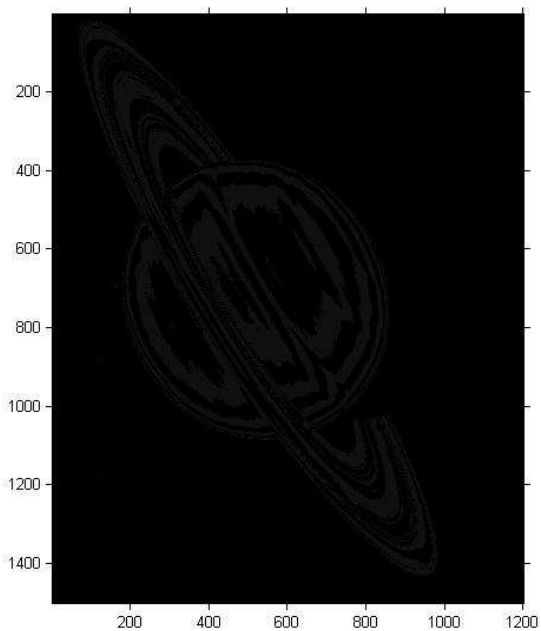


**Figura 6:** Imagen en escala de grises con argumento de entrada de Bitand en 00001000 (8 decimal).

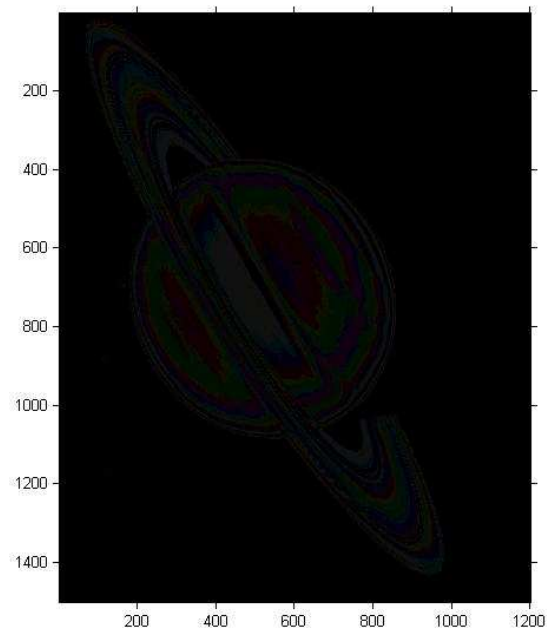


**Figura 7:** Imagen RGB con argumento de Bitand en 00001000 (4 decimal).

La figura 8 y 9 reflejan la función Bitand con una máscara de 00010000, se observa que se resaltan aún más los contornos para las dos imágenes, empieza a aparecer el anillo del planeta y la forma redondeada de éste, para la imagen en blanco y negro se nota más lúcida que la imagen en color, pero esta última empieza a resaltar colores en verde y rojos.

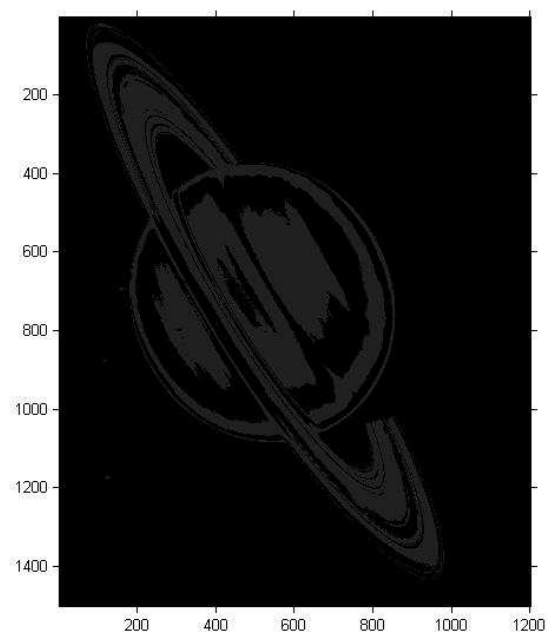


**Figura 8:** Imagen en escala de grises con argumento de entrada de Bitand en 00010000 (16 decimal).

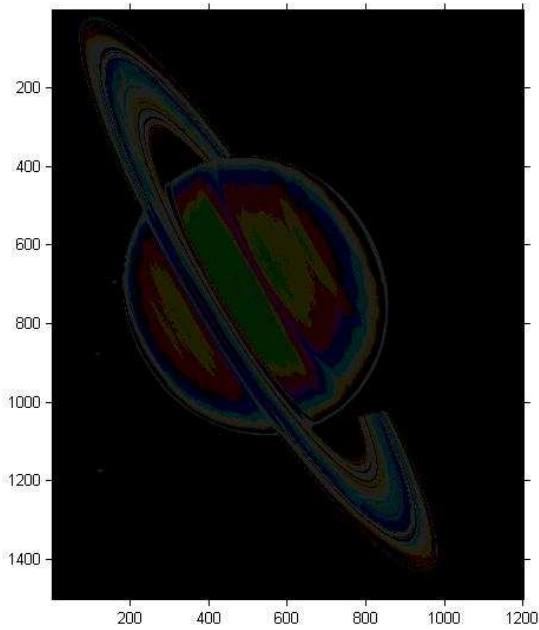


**Figura 9:** Imagen en escala de grises con argumento de entrada de Bitand en 00010000 (16 decimal).

En las figuras 10 y 11 se resaltan aún más los bordes de las imágenes, para la figura 8 se va a acercando al color blanco debido a que se va a acercando al bit más significativo de cada píxel, aparecen por primera vez las lunas, lo cual denota que en ese tipo de puntos pequeños están activos los tonos blancos; para la imagen en color se empiezan a notar tonos azules en los extremos de las zonas de la imagen, verdes y rojos en el interior y empiezan a aparecer combinaciones de estos.

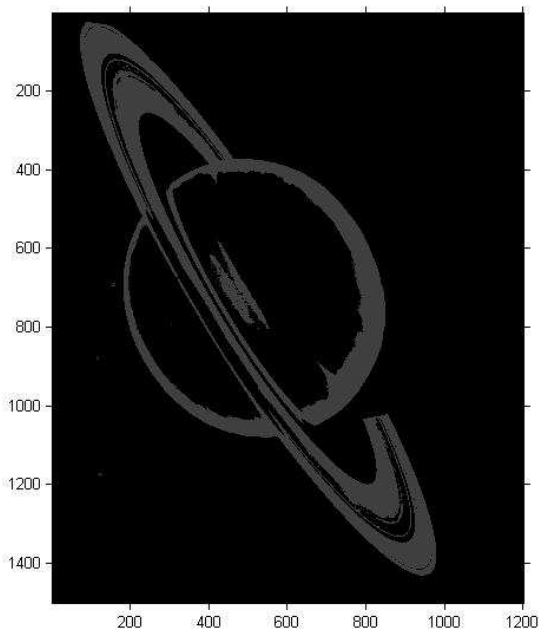


**Figura 10:** Imagen en escala de grises con argumento de entrada de Bitand en 00100000 (32 decimal).

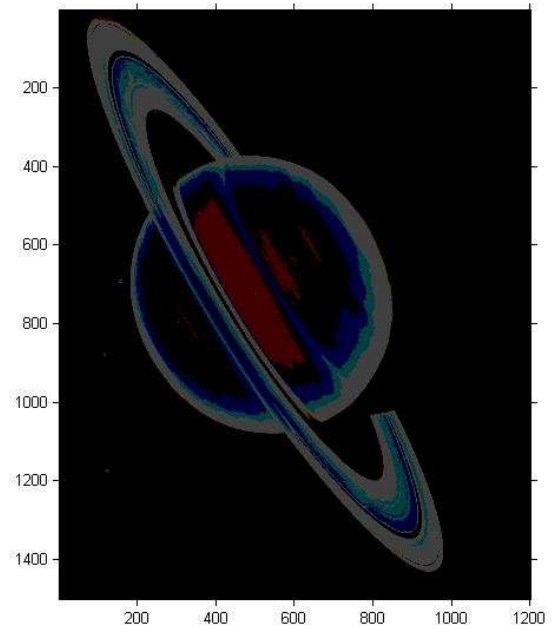


**Figura 11:** Imagen RGB con argumento de entrada de Bitand en 00100000 (32 decimal).

En lo que respecta a la figura 12 y 13 al estar activa la comparación lógica en la parte alta se empieza a notar cada vez mas el contorno blanco de la imagen en escala de grises, empiezan a desaparecer las partes internas pero conserva la esencia de la figura; para la imagen en color, desaparece el color verde y aumentan los tonos azules en los contornos.

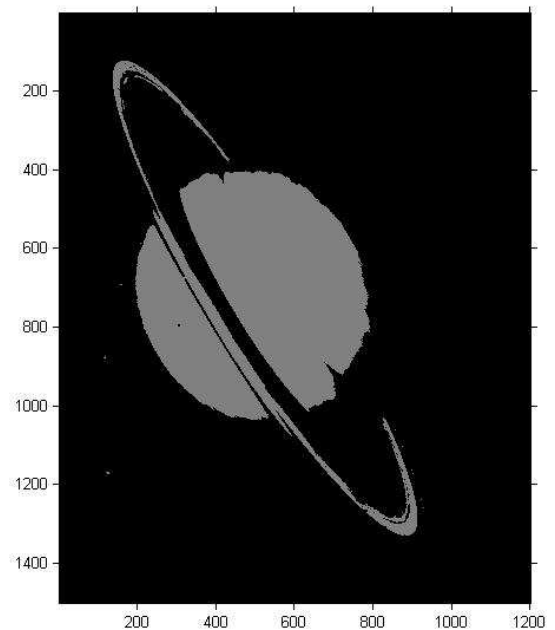


**Figura 12:** Imagen en escala de grises con argumento de entrada de Bitand en 01000000 (64 decimal).

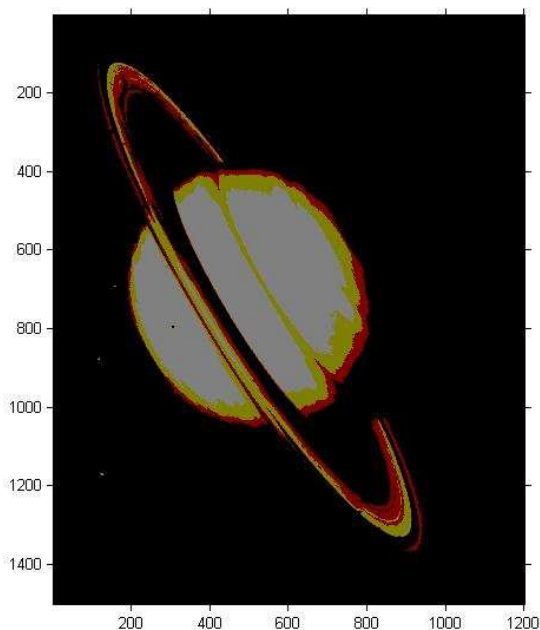


**Figura 13:** Imagen RGB con argumento de entrada de Bitand en 01000000 (64 decimal).

Por último para las figuras 14 y 15 pierden definición de contornos las imágenes, desaparece poco a poco el anillo del planeta, se ve aún más blanca la imagen en escala de grises y se resaltan los amarillos y rojos en los contornos de la imagen en color.



**Figura 14:** Imagen en escala de grises con argumento de entrada de Bitand en 10000000 (128 decimal).



**Figura 15:** Imagen RGB grises con argumento de entrada de Bitand en 10000000 (128 decimal).

#### 4. Análisis de Resultados

La secuencia de imágenes muestra el proceso en el cual se va aumentando a la siguiente potencia de 2 la máscara de la comparación lógica, en dicho proceso se vio el aumento de tonalidad de la imagen, debido a que el ojo humano no es capaz de observar cambio pequeños de tonalidad las tres primeras imágenes se ven negras, al aumentar la máscara a 8 se empiezan a notar los ciertos contornos grises en las imágenes pero resulta algo difuso de distinguir esto se debe a que los bits presentes en la and solo son los que tengan el 5 bit en 1 el resto se vuelve 0 por lo cual sigue predominando el negro. Para las siguientes dos imágenes empieza a aparecer la parte alta de la máscara, del bit 4 al bit 2 se nota el contorno redondeado del planeta y el anillo que lo rodea, pero empiezan a desaparecer las partes internas, esto se debe a que la intensidad se concentra para el caso más los bordes que en el centro para la imagen en escala de grises; para la imagen de color se nota una variación en el color de esta, empieza con colores tendientes al verde y termina con matices tendientes al azul, esto se debe a que en la matriz de los verdes están activos los bits de la parte media a baja el rojo en la parte media alta y el azul al igual q el rojo tiene bits activos en la parte media alta. Por último se notó que si la máscara tiene el bit más significativo activo e pierde el contorno de la imagen pero se aumenta la presencia de tonos tendientes al blanco, esto se debe a que se llegó a la representación en bits de los tonos blancos, por esto en la imagen en blanco y negro todo el centro se ve tendiente al blanco y desaparece la definición del aro del planeta y para la imagen de color se resaltan los tonos blancos y los tendientes a este como el amarillo y el rojo claro.

#### 5. Conclusiones

Dependiendo de la máscara usada se puede segmentar de una manera más efectiva una imagen, si se quieren contornos, se debería usar una máscara tendiente del quinto al séptimo bit de la representación de cada píxel, pero si se quieren definiciones internas del objeto, se debe tener más en cuenta el bit más significativo del píxel, este caso es particular, puede que para una imagen diferente se necesite otro tipo de máscara o un filtrado más riguroso, inclusive, si se quiere obtener alguna característica relevante de la imagen se deberían tener en cuenta particularidades de esta, como formas básicas, colores, tonos y cualquier otro tipo de características.

#### 6. Referencias

- [1] Apuntes de clase.
- [2] James H. McClellan, Ronald W. Schafer y Mark A. Yoder, DSP First: A Multimedia Approach Prentice-Hall, Inc., New Jersey, USA, 1998.
- [3] <http://www.slideshare.net/dpatriciocastillom/images-matlab>
- [4] <http://www.slideshare.net/lonely113/procesamiento-digital-de-imagenes-con-matlab>