TP Especial Nº1: Segmentación de Imágenes

Bigio, Rafael Martín
Coffey, Santiago Andrés
Gregoire, Andrés Santiago

Introducción

En este trabajo se analizan los siguientes métodos de segmentación de imágenes: k-means, split and merge y antipole tree. Para esto se implementó un framework en Java que permite experimentar con los diferentes métodos, así como aplicarle filtros a la imagen antes y después de ser segmentada.

Los métodos se basan en alguna característica de la imagen para poder segmentarla. A esta característica se la denomina *feature*, y un mismo método puede ser aplicado con distintas *features* de la imagen según el resultado que se desee obtener. Si se toman varias características juntas para cada píxel, entonces se habla de un *feature vector*.

Descripción del framework

El framework permite realizar la siguiente secuencia de acciones: abrir una imagen, aplicarle varios filtros, segmentarla y aplicarle filtros al resultado de la segmentación. Una vez abierta una imagen, es posible aplicarle una serie de filtros. Los filtros implementados son los siguientes:

- **Blur**: suaviza la imagen.
- **Sharpen**: marca los bordes.
- **Ecualización**: ecualiza la imagen.
- **Reducir resolución**: reduce la resolución de la imagen siguiendo las pautas de [Shamik Sural, Gang Qian and Sakti Pramanik]
- **Max**: le asigna a cada componente RGB de cada pixel el máximo de sus aledaños.
- **Min**: le asigna a cada componente RGB de cada pixel el máximo de sus aledaños.
- **Max-Min**: Max: le asigna a cada componente RGB de cada pixel la diferencia entre el máximo y el mínimo de sus aledaños.
- **Midpoint**: le asigna a cada componente RGB de cada pixel el promedio entre el máximo y el mínimo de sus aledaños.
- **Average**: le asigna a cada componente RGB el promedio de los aledaños.

Para realizar la segmentación de la imagen resultante, se debe indicar qué característica (*feature*) de la misma se desea considerar para hacer dicho proceso. El cálculo de estas características está desacoplado de los métodos de segmentación, por lo tanto se puede combinar cualquier característica con cualquier método. Las características que se pueden considerar son las siguientes:

- **Color**: color de cada pixel en RGB o HSB.
- **Histograma de color**: histograma de color (RGB o HSB) de cada pixel y sus ocho aledaños. Se puede parametrizar con la cantidad de clases por canal.
- **Histograma espacial de color**: A diferencia del anterior, en este se considera la posición de pixel. Se puede indicar la influencia de la coordenada horizontal y de la vertical.

Una vez especificada la *feature* a utilizar, se procede a segmentar la imagen con algún método. Los métodos de segmentación asignan un número de segmento a cada píxel de la imagen. Luego, el framework se encarga de asignarle diferentes colores a los números de segmento, para poder visualizarlos. El criterio es simplemente tomar colores equidistantes en el cono de HSB, en la región de máxima saturación y brillo. Se implementaron los siguientes métodos de segmentación:

- **K-Means**: implementación del algoritmo básico de clusterización. Se le debe indicar la cantidad de zonas en que se debe segmentar la imagen. Para que la imagen no se sobre-segmente, es posible indicarle la cantidad máxima de segmentos para particionar y de ser necesario, el método utilizará menos.

- **Antipole Tree**: Es una alternativa para no tener que indicar la cantidad de clusters en que se desea segmentar la imagen. La idea del método es agrupar los pixeles que se encuentran a una distancia menor que un radio del centroide del cluster. En consecuencia, el único parámetro de este método es el radio de los clusters.

- **Split & Merge**: implementación del algoritmo básico de split and merge, que divide la imagen a modo de quadtree hasta que todas las regiones obtenidas cumplen un determinado criterio de homogeneidad, y luego junta las regiones vecinas que sigan satisfaciendo dicho criterio. Recibe como parámetro el desvío estándar que se utiliza como criterio de corte para la etapa de split, el que se utiliza para la etapa de merge, y el área mínima de las regiones obtenidas.

Finalizado todo este proceso, es posible guardar la imagen generada.

Como características adicionales, el framework permite agrandar y achicar las imágenes para poder comparar con mayor detalle el funcionamiento de los métodos. Asimismo, es posible ver dos imágenes con las que se trabaja distribuídas vertical y horizontalmente en la pantalla. Por último se pueden todos los formatos de imágenes soportados por Java pero solo se puende grabar en jpg.

Análisis de los métodos de segmentación

K-means

El primer de caso de prueba se realiza sobre la siguiente imagen

Si se segmenta dicha imagen utilizando como característica a extraer únicamente el RGB de cada pixel y se solicita que se generen 6 segmentos, el resultado es el siguiente:

Como se puede observar, en las zonas de los brazos y el cuello hay micro-segmentos que deberían ser erosionados. Para solucionar, hay dos alternativas: aplicarle un filtro de suavizado (blur) previo a la segmentación o considerar histograma RGB con 5 intervalos de clase por canal como feature de cada pixel. En las siguientes imágenes se exponen los resultados de las dos alternativas respectivamente:

La única deferencia notoria entre los dos resultados es que aplicando histograma de color, el contorno de la persona se marca homogéneamente. En ambos casos, se notan mejorías sobre el problema de las micro-zonas.

En la siguiente imagen se intenta agrupar en un mismo segmento el avión completo. Si bien la imagen no es muy compleja, el objeto que se quiere identificar tiene varios colores y zonas muy iluminadas.

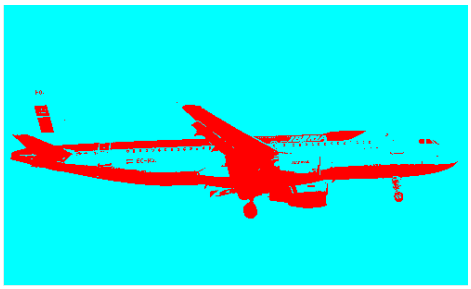
Considerando como característica el histograma HSB de cada pixel con 5 clases por canal, se obtiene el siguiente resultado:

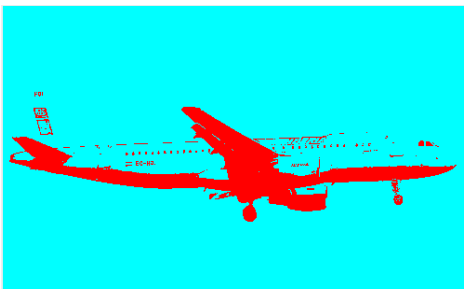
Si previo al proceso de segmentación se realiza una ecualización de la imagen y luego se considera como feature, el histograma RGB de cada pixel y se fija la influencia espacial en ambas coordenadas en 500, el resultado pasa a ser:

La ecualización de la imagen ayuda un poquito ya que amplía el espectro de color. El hecho de considerar además de histograma de color, la influencia espacial en ambas coordenadas, ayuda a que partes del avión que antes quedaban en el segmento del cielo, pasen a formar parte de él.

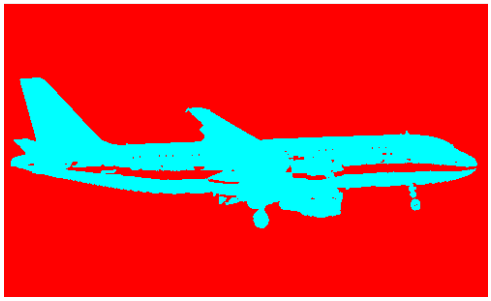
En esta otra imagen también se intenta separar el avión del cielo. La imagen no es tan sencilla para segmentar en dos zonas porque el avión tiene zonas con sombras, otras con brillo y además tiene varios colores.

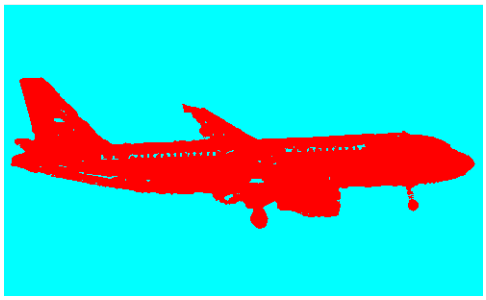
Si se considera únicamente como característica el RGB o HSB de cada pixel los resultados tras la segmentación son los siguientes respectivamente:





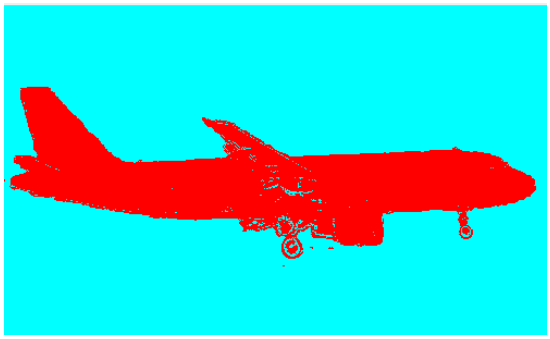
Si se considera en vez del color, el histograma de RGB HSB, los resultados son los siguientes respectivamente:



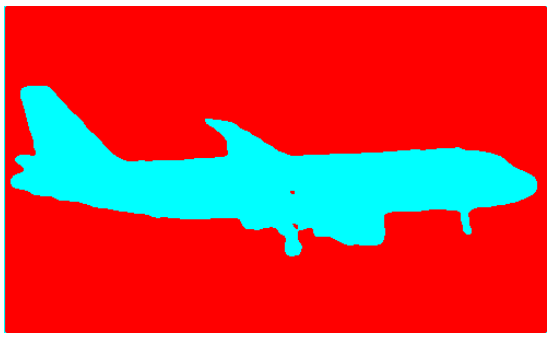


Si bien en ambos casos se notan mejorías, aun quedan zonas internas del avión en el segmento del cielo.

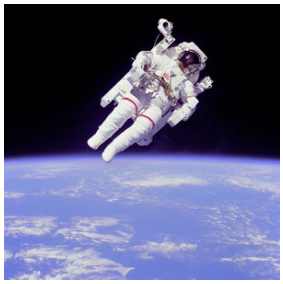
El resultado de la segmentación mejora aplicando el filtro para reducir la resolución y luego considerar como característica el RGB de cada pixel. La imagen resultante es la siguiente:



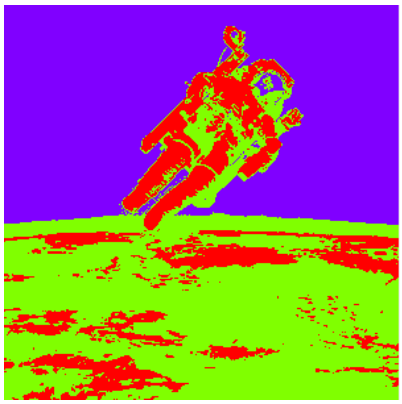
Para eliminar las micro-zonas, se puede después de aplicar el filtro para reducir la resolución aplicar un suavizado. El resultado es:



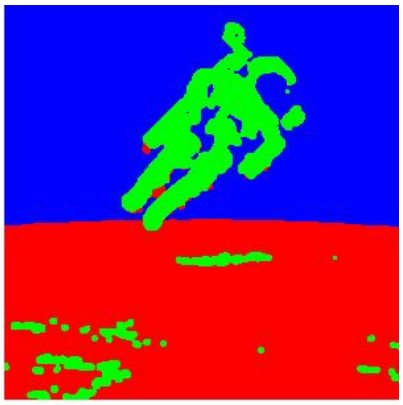
En la siguiente imagen se busca dividir el astronauta del espacio y de la tierra.



Si solo se considera como feature el color, los resultados no son nada buenos:

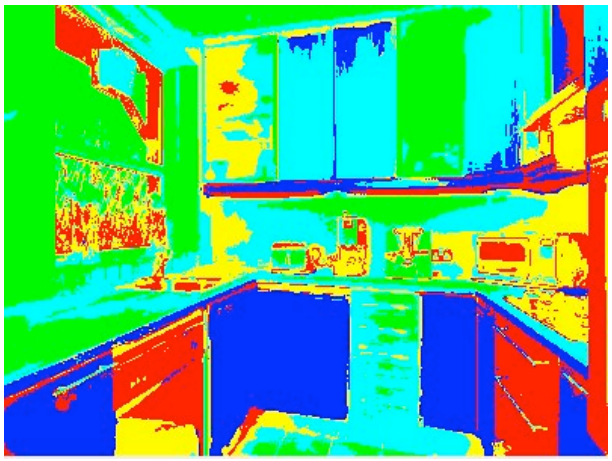


Si se aplica el filtro midpoint y como característica se toma histograma RGB con influencia espacial de ambas coordenadas en 1200, el resultado es el siguiente:



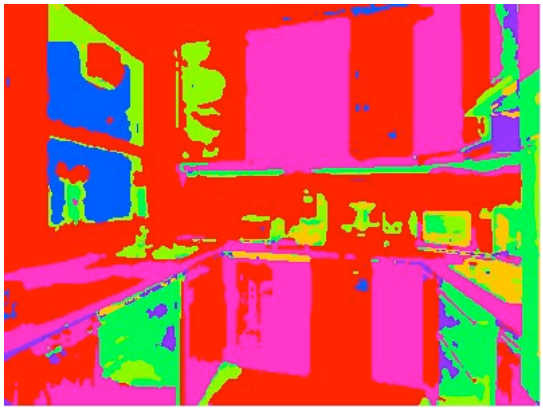
La siguiente imagen no es sencilla para segmentar porque tiene varios objetos de diferentes texturas. Asimismo, hay varias zonas con reflejos.

Utilizando como característica el RGB de cada pixel y pidiendo 10 segmentos como máximo, se obtiene la siguiente agrupación:

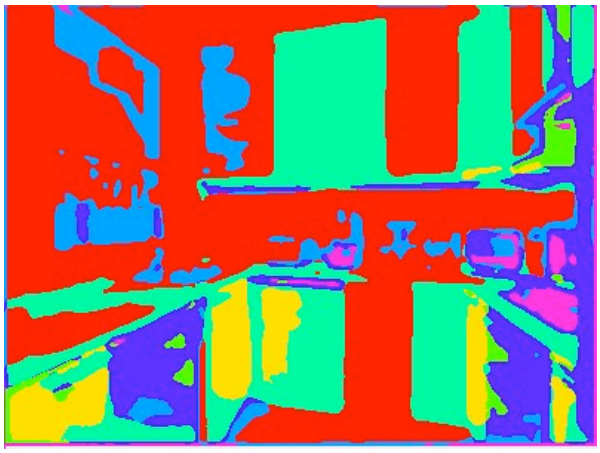


Se puede ver que la imagen está demasiado erosionada en ciertas zonas; por ejemplo, no se llega a completar adecuadamente las alacenas superiores.

Utilizando como característica histogramas RGB o HSB con 5 clases por canal respectivamente, los resultados son muy similares. En ambos casos, se logra segmentar bastante bien las alacenas pero tiene problemas con las zonas donde hay cambio de brillo.



En este último caso, además de extraer como característica de cada pixel el histograma hsb con 5 intervalos de clase, se realiza un blur de la imagen. A veces resulta muy útil hacer esta operación ya que suaviza la imagen eliminando patrones de textura. Se puede ver que la imagen segmentada hay menos zonas que requieren erosión. La aplicación de un filtro de suavizado tiende a eliminar los objetos pequeños que a la hora de segmentar la imagen puede que no tengan importancia. Por el contrario, el filtro sharpen, tiende resaltar dichos objetos dificultando (por lo menos en este caso) la segmentación.



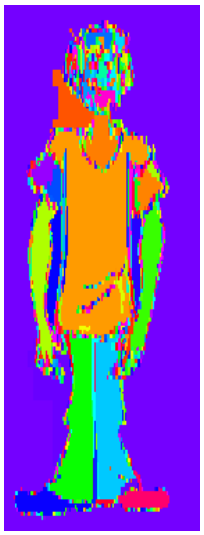
Split and merge

La etapa de split se implementó dividiendo la imagen en cuatro partes, de la manera que lo haría un quadtree. Cada vez que se obtiene una nueva zona, se le aplica un criterio de homogeneidad. Si es homogénea no se divide, y si no lo es, sí. El criterio de homogeneidad implementado consiste en tomar el desvío estándar de cada componente del *feature vector* de cada píxel de la imagen. Luego, se verifica si todas las componentes son menores que una cota determinada, llamada el desvío estándar de split. Si todas son menores, entonces la zona se considera homogénea. Si alguna es mayor, entonces es heterogénea y debe volverse a dividir.

Además, se utiliza un segundo criterio que es el área mínima de una zona. Si un segmento llega a un área menor o igual que este parámetro, entonces no se lo divide, independientemente del criterio de homogeneidad.

Luego, para la etapa del merge, se itera sobre cada segmento obtenido y para cada segmento vecino se evalúa si juntos seguirían cumpliendo el criterio de homogeneidad. Si bien esta cota debería ser la misma que para la etapa de split, se la parametrizó aparte a través del parámetro desvío estándar de merge, para darle mayor flexibilidad al algoritmo. A veces es preferible que la etapa de split sobre-segmente, y ser menos restrictivos en la etapa de merge, para que los segmentos se vuelvan a juntar.

En la siguiente imagen, se aplica este algoritmo, tomando como característica el color en RGB, un desvío de split de 5, un desvío de merge de 15, y un área máxima de 1 píxel.



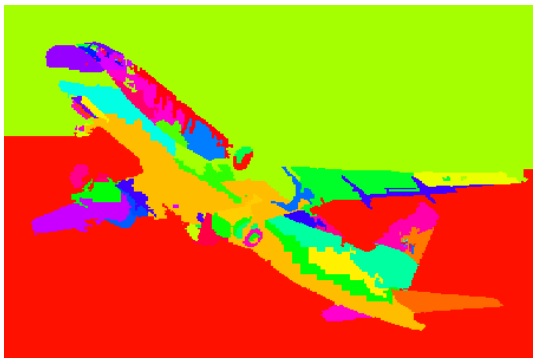
Se puede observar que está sobre-segmentada, que es el problema que ocurre cuando se utiliza este algoritmo. Una alternativa para reducir la sobre-segmentación es aumentar el desvío de merge. Subiéndo dicho umbral a 30, el resultado es el siguiente:

Se puede observar que hay menos segmentos pequeños en la zona de los brazos y las piernas, pero sin embargo el problema persiste en la cabeza y la cara. Esto está muy relacionado con la geometría de los segmentos que genera este método. Como siempre divide en cuatro partes un segmento no homogéneo, la forma de los segmentos de tamaño mínimo depende de la proporción de la imagen inicial. En este caso, como la proporción entre ancho y alto es casi 1:3, entonces el tamaño del segmento mínimo será de 1 x 3 píxeles. Por lo tanto, por más que se indique que el área mínima deseada para un segmento es de 1 píxel, esto no garantiza que el menor segmento tendrá este área, ya que en el ejemplo del caso anterior, los segmentos de área 3 ya son indivisibles. Esto hace que dos segmentos contiguos puedan ser clasificados como heterogéneos en zonas de mucha variabilidad como la cara o la cabeza de la imagen del ejemplo.

Otra alternativa es pensar en aplicar un blur antes para reducir estos cambios bruscos entre píxeles. El resultado con los mismos parámetros que antes pero aplicando previamente un filtro blur es el siguiente:

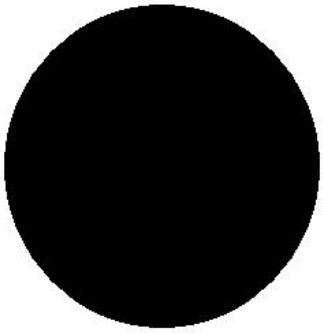
Se observa que se redujo bastante la sobre-segmentación, pero se perdió completamente el detalle de la cara y la cabeza. Por lo tanto este método puede resultar útil para detectar zonas "grandes" en imágenes que no tienen una relación 1:1 entre ancho y alto, ya que este factor puede afectar bastante el resultado de la segmentación para segmentos de tamaño pequeño.

En el ejemplo del avión, si se aplica este método tomando como *feature* el RGB de cada píxel, desvío de split 5, desvío de merge 15 y área mínima 3 se obtiene el siguiente resultado:

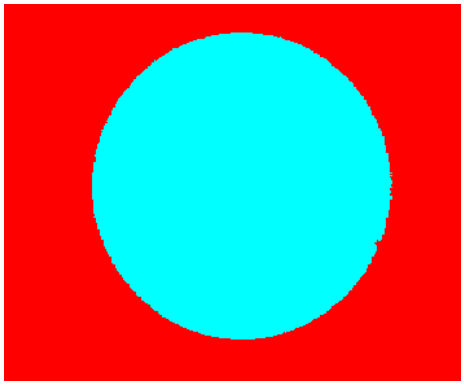


Nuevamente se puede ver que la imagen queda sobresegmentada. El problema particular que presenta esta imagen, es que como se utiliza como feature el color, entonces si se sube el umbral de merge, parte del avión se confunde con el cielo, entonces se necesita tomar otra característica. Si se utiliza color por HSB los resultados no son favorables, ya que segmenta aún más el avión, por las diferencias que posee en saturación y brillo. Incluso tomando histograma por RGB como característica, el problema aumenta ya que esto hace que haya mayor diferencia entre píxeles aledaños, y esto dificulta la segmentación.

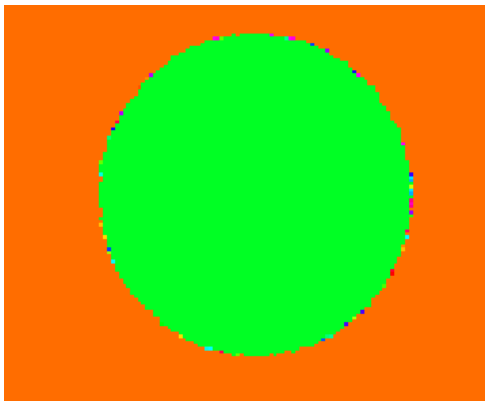
Una prueba interesante para ver las desventajas de la geometría de este método es segmentar un círculo, como en la siguiente imagen:



Tomando como área mínima 1 píxel, y separando a través de RGB, el resultado es el siguiente:

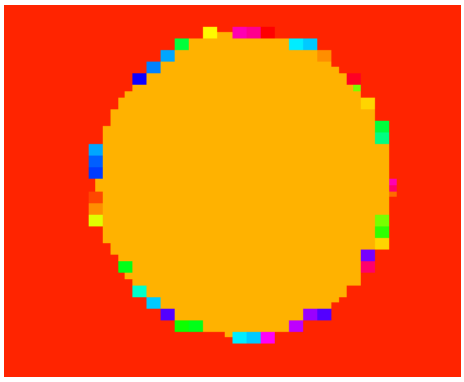


Se puede observar que prácticamente detectó el círculo sin problemas. Sin embargo, si la imagen es más grande, y se aumenta el área mínima a 9 píxeles (tal vez porque se requiere mayor eficiencia en el procesamiento), se observa lo siguiente:



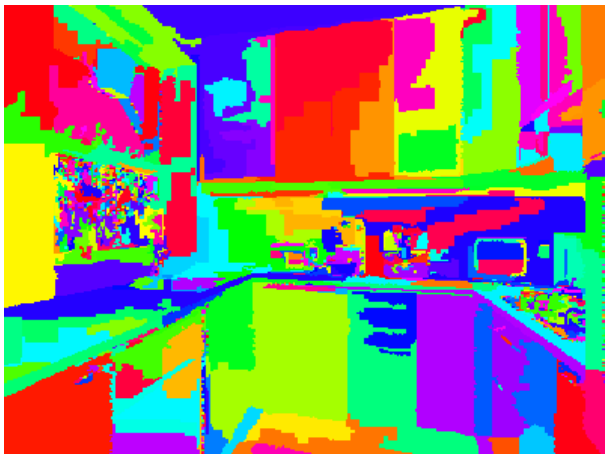
En este caso ya se pueden observar los "cuadrados" que genera este método, así como también la sobre-segmentación. Al tener segmentos mínimos de mayor tamaño, los bordes (casos límites) pueden no unirse a ninguno de los dos segmentos aledaños, y permanecer como un segmento diferente, como se observa en algunos puntos de la imagen.

Finalmente, llevando esto al extremo para ponerlo aún más evidencia, si el área mínima se aumenta a 100 píxeles (segmentos de 10 x 10), entonces el resultado es:



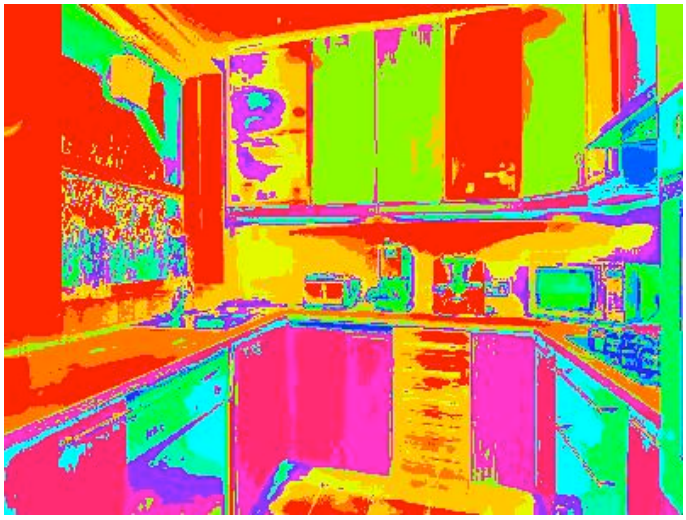
Se puede ver la cantidad de segmentos de más que generó. Un blur anterior solucionaría parcialmente el problema como en el primer ejemplo, pero seguirá existiendo un trade-off entre el área mínima y el tiempo del que se dispone para la segmentación. Si solamente se quieren detectar segmentos "grandes", entonces se puede aumentar este parámetro para lograr eficiencia. En caso contrario, no.

Por último, en el ejemplo de la cocina se observa que tomando como feature el RGB de cada píxel, 5 como desvío de split, 15 como desvío de merge y área mínima 3 se detectan con bastante claridad los segmentos (algunas partes sobre-segmentadas), y además se puede ver que en el caso de las plantas que se ven a través de la ventana quedó extremadamente sobre-segmentada la imagen (nuevamente, por el problema de los cambios bruscos de color en zonas pequeñas).



Antipole tree

Se introduje este método de segmentación para evitar el problema de tener que definir la cantidad de segmentos en que se desea agrupar la imagen. Podría esperarse que conociendo la imagen a procesar, sea más sencillo definir el radio de los segmentos que se desean. No obstante los resultados en general son peores que otros métodos porque sobre-segmenta la imagen. En los mejores casos, los resultados son prácticamente los mismos que utilizando K-Means. Por este motivo solo se realiza un caso de prueba del método. Para el mismo, se utiliza como característica el RGB de cada pixel y se parametriza el método definiendo el radio máximo de cada cluster en 100. La imagen resultante es la siguiente:



Se puede observar que fue sobre-segmentada teniendo más de 15 segmentos.

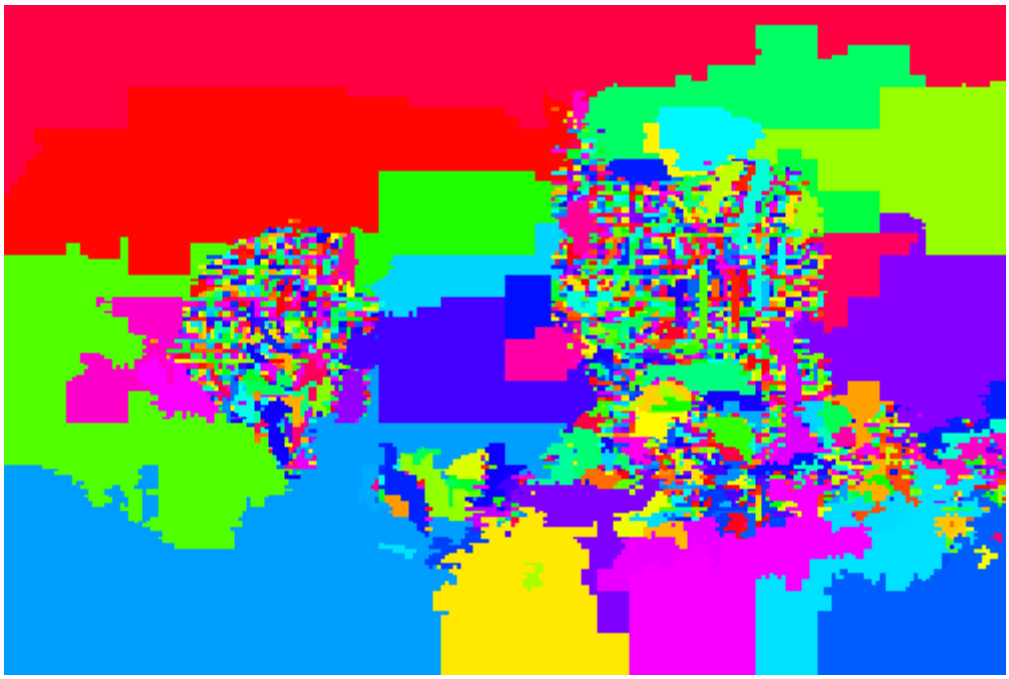
Otros ejemplos

Archivo: cebras.jpg
Filtros: Ninguno
Feature: Color RGB
Segmentación: K-means con 18 clusters

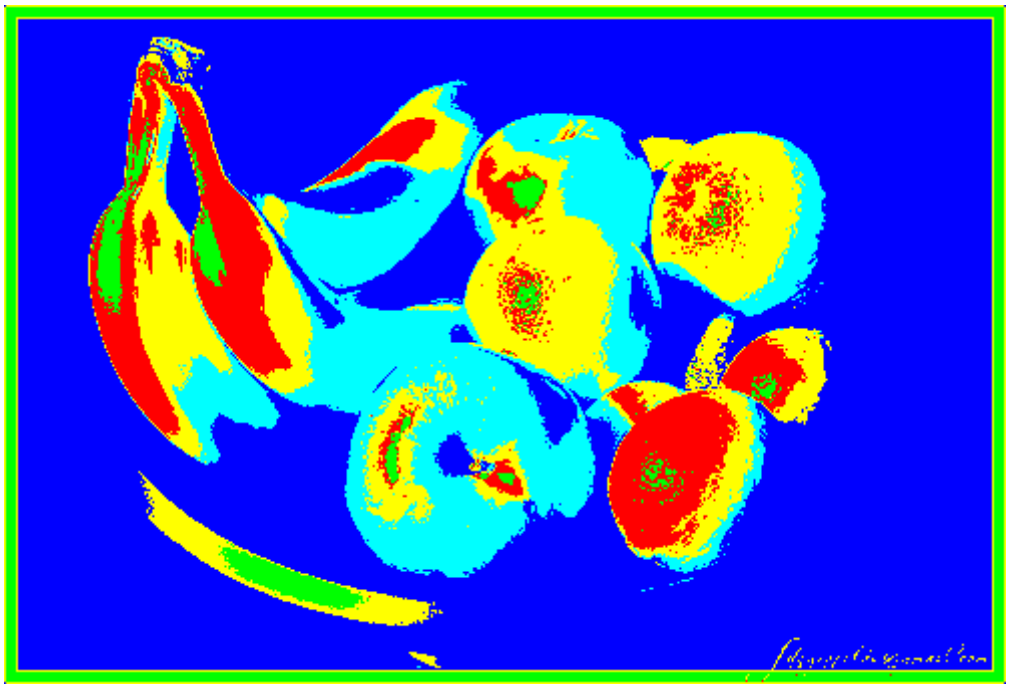
Archivo: cebras.jpg
Filtros: Ninguno
Feature: Color HSB
Segmentación: K-means con 12 clusters

Archivo: cebras.jpg
Filtros: Blur
Feature: Color RGB
Segmentación: K-means con 15 clusters

Archivo: cebras.jpg
Filtros: Max-min
Feature: Color RGB
Segmentación: K-means con 18 clusters



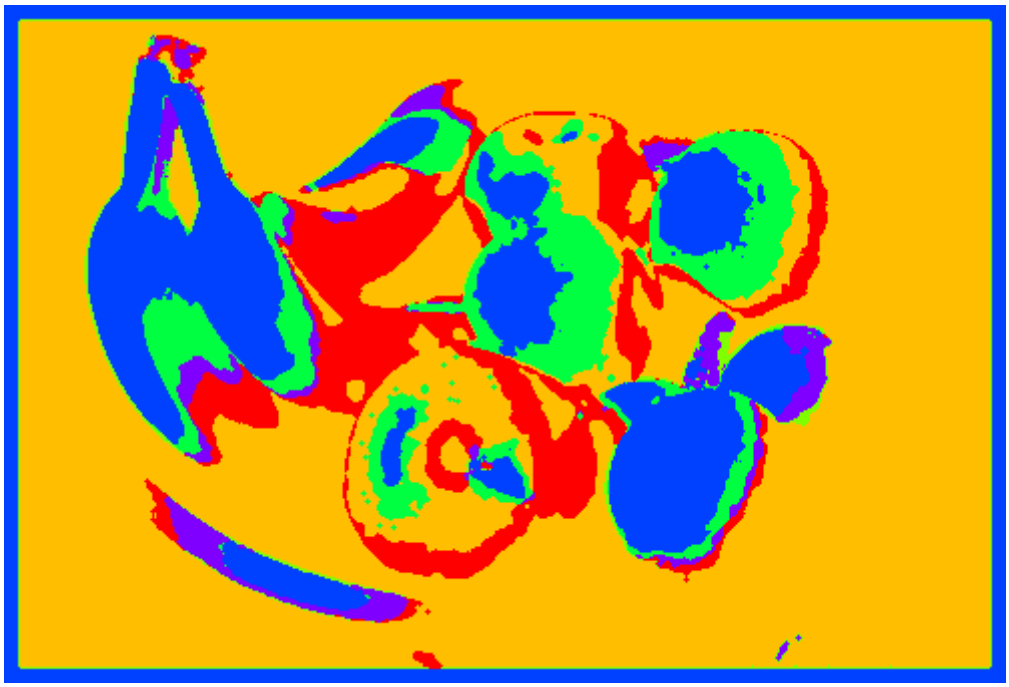
Archivo: cebras.jpg
Filtros: Ninguno
Feature: Color RGB
Segmentación: Split & Merge con límites 10 y 20 y área mínima 10. (En general este método no daba buenos resultados con esta imagen.)



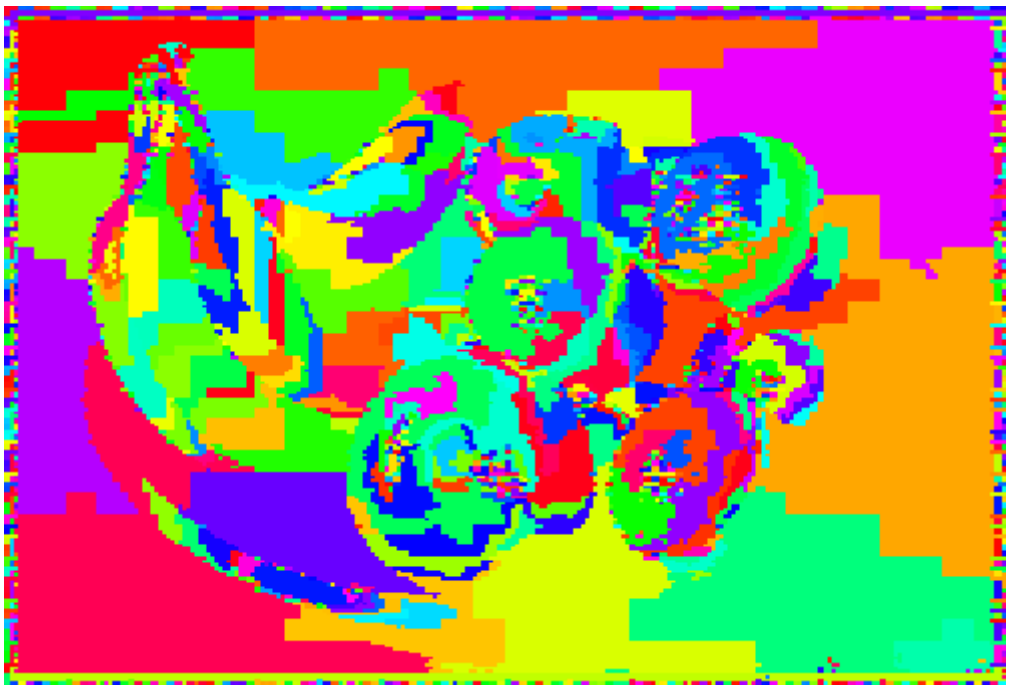
Archivo: frutas.jpg
Filtros: Ninguno
Feature: RGB
Segmentación: K-means con 10 clusters

Archivo: frutas.jpg
Filtros: Average
Feature: Histograma HSB
Segmentación: K-means con 10 clusters

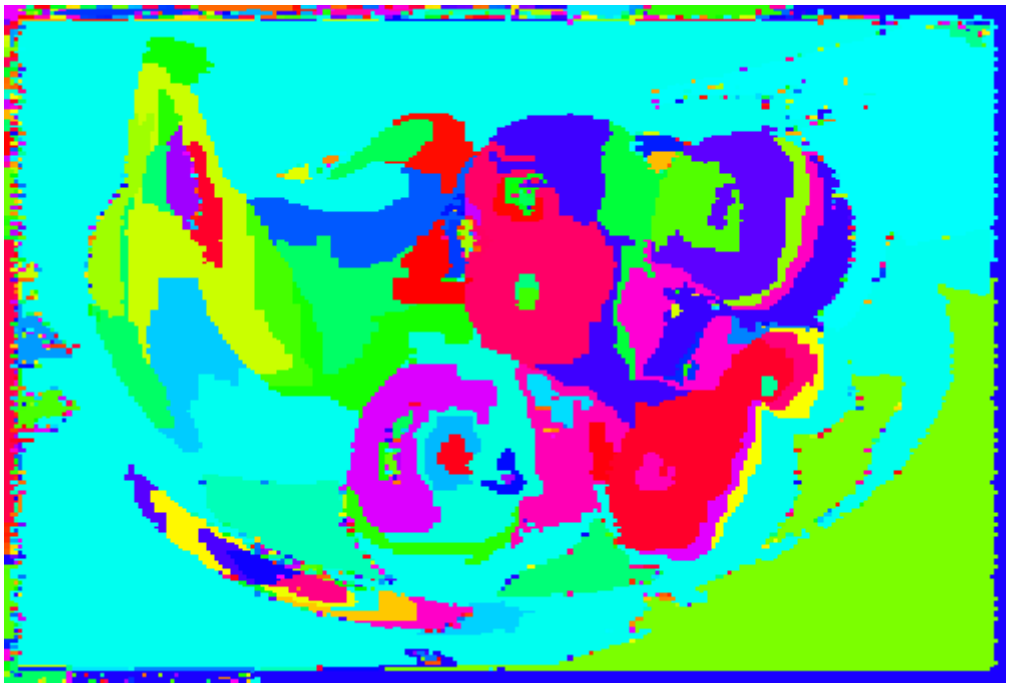
Archivo: frutas.jpg
Filtros: Average
Feature: HSB
Segmentación: K-means con 8 clusters



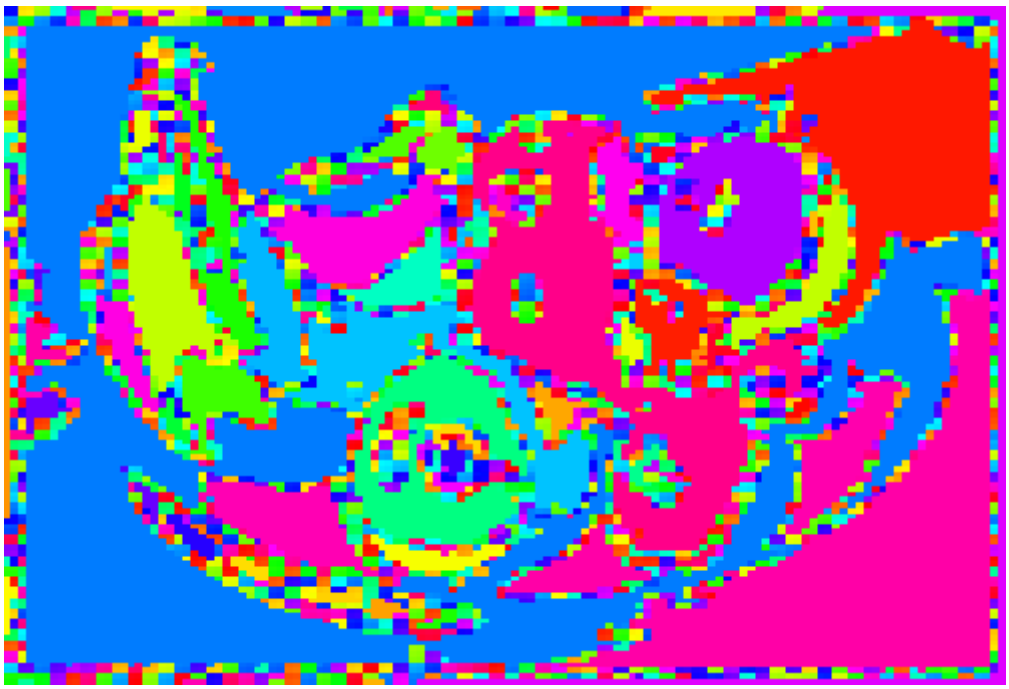
Archivo: frutas.jpg
Filtros: Max-min
Feature: Histograma HSB
Segmentación: K-means con 10 clusters



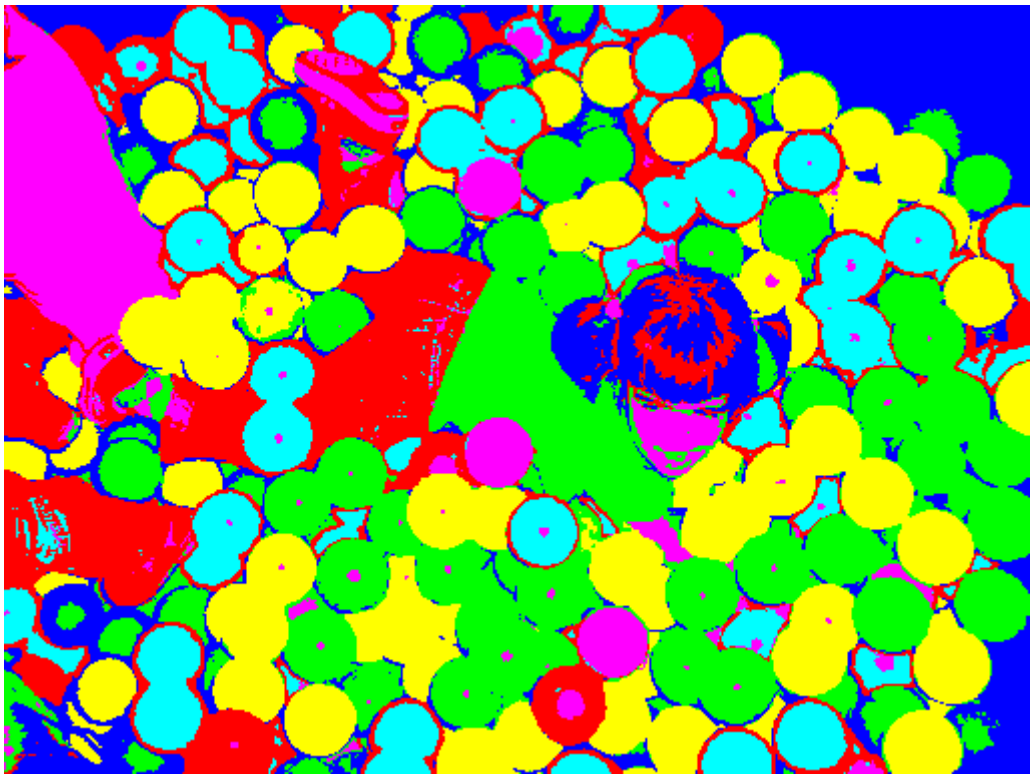
Archivo: frutas.jpg
Filtros: Ninguno
Feature: RGB
Segmentación: Split & Merge con límites 10 y 20 y área mínima 10.



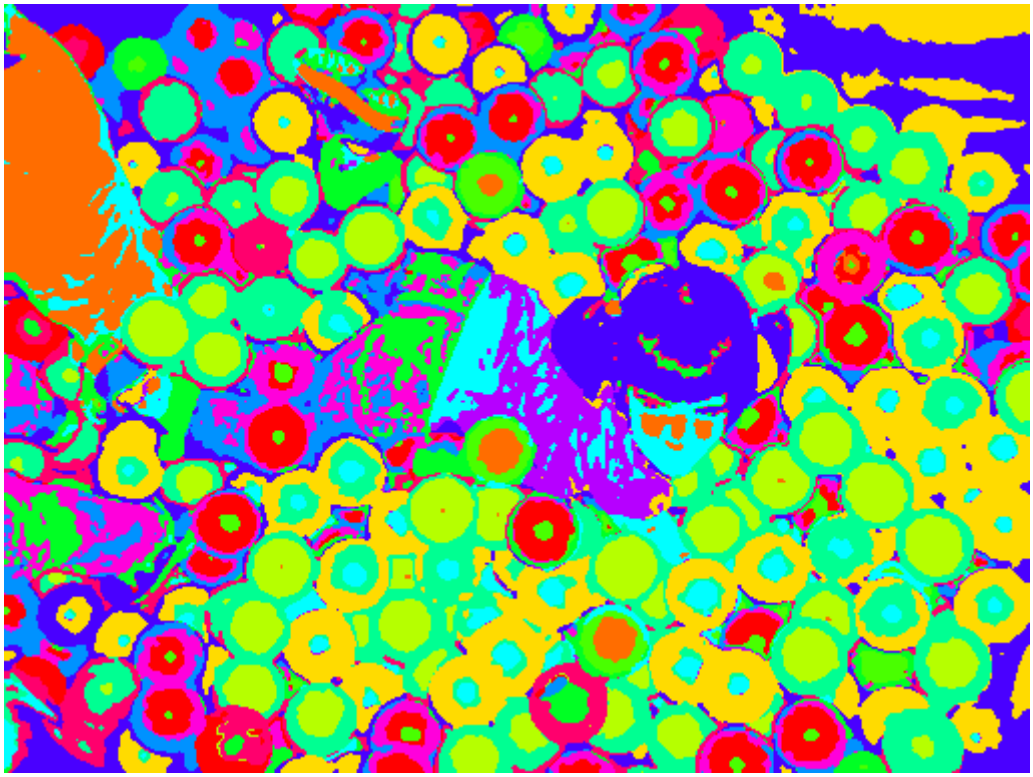
Archivo: frutas.jpg
Filtros: Average
Feature: HSB
Segmentación: Split & Merge con límites 5 y 30 y área mínima 10.



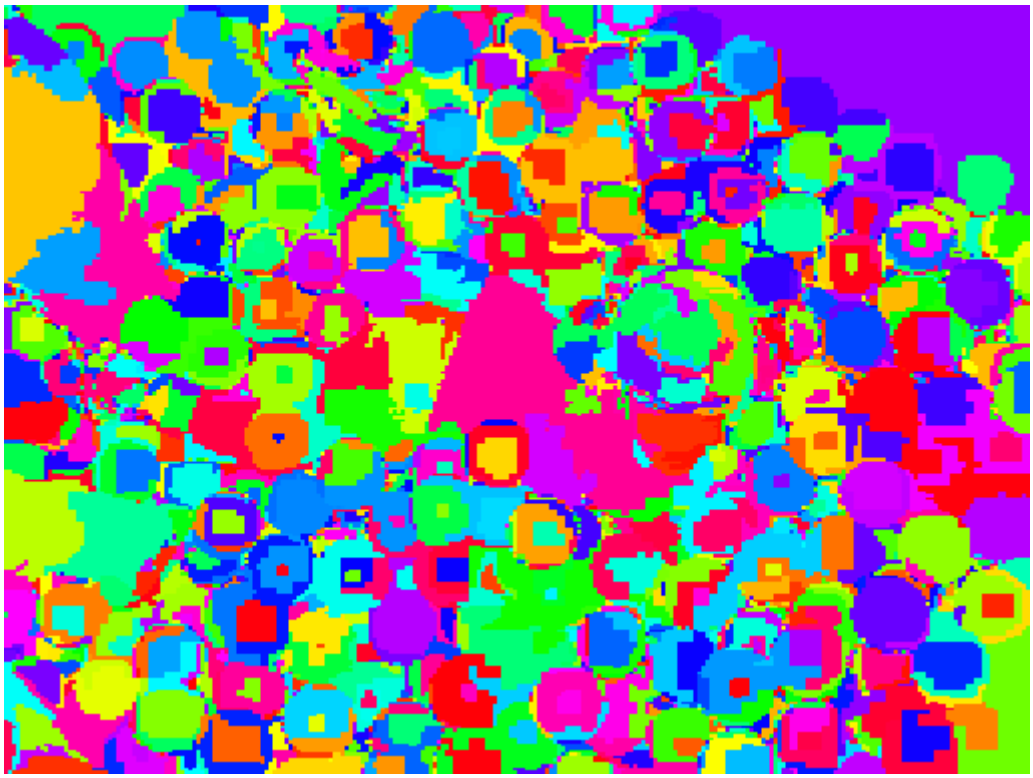
Archivo: frutas.jpg
Filtros: Blur
Feature: Histograma HSB
Segmentación: Split & Merge con límites 10 y 20 y área mínima 10.



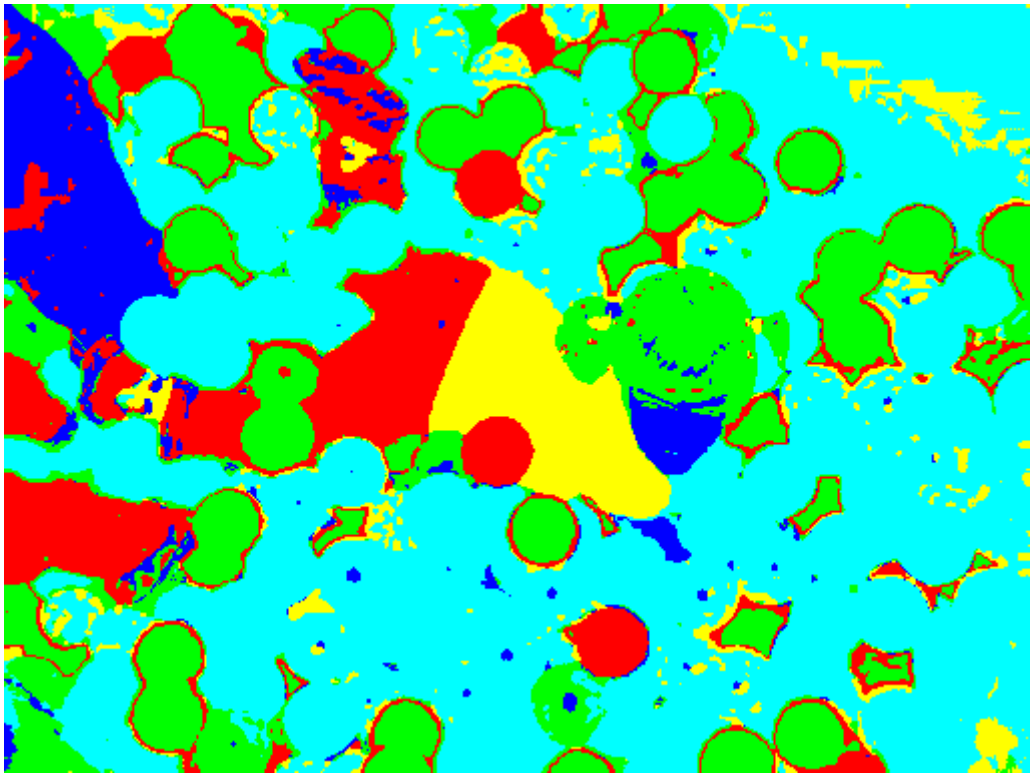
Archivo: pelotero.jpg
Filtros: Ninguno
Feature: RGB
Segmentación: K-means con 10 clusters.



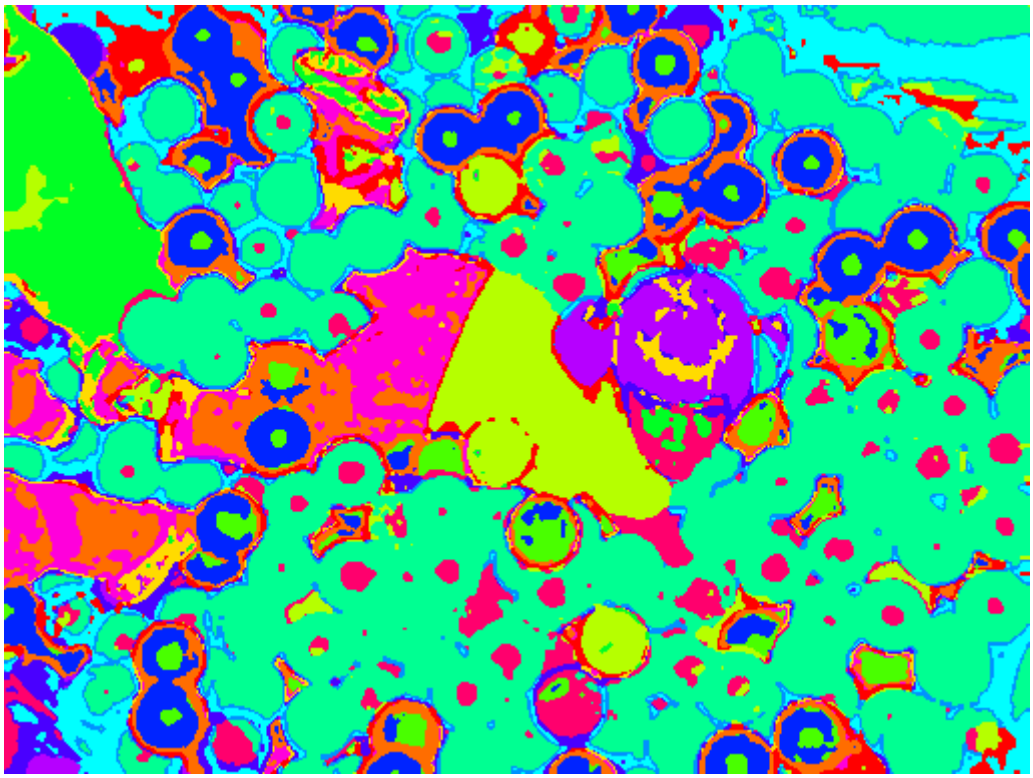
Archivo: pelotero.jpg
Filtros: Ninguno
Feature: Histograma RGB
Segmentación: K-means con 10 clusters.



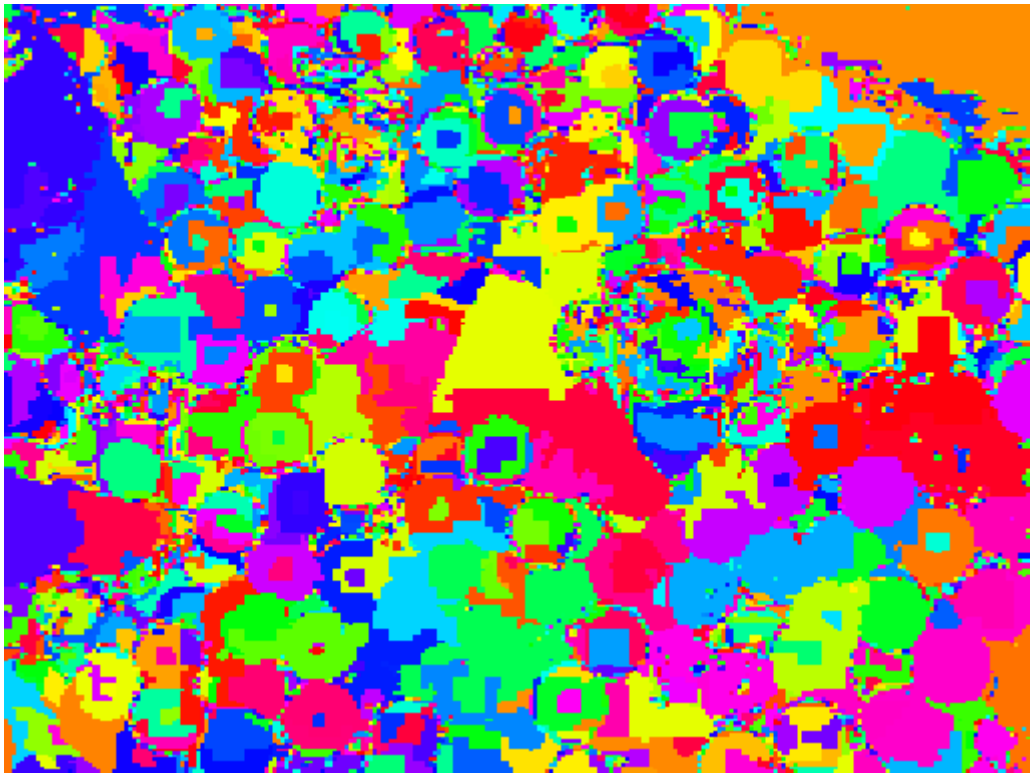
Archivo: pelotero.jpg
Filtros: Ninguno
Feature: RGB
Segmentación: Split & Merge con límites 10 y 20 y área mínima 10.



Archivo: pelotero.jpg
Filtros: Ninguno
Feature: HSB
Segmentación: K-means con 10 clusters.



Archivo: pelotero.jpg
Filtros: Ninguno
Feature: Histograma HSB
Segmentación: K-means con 15 clusters.



Archivo: pelotero.jpg
Filtros: Ninguno
Feature: HSB
Segmentación: Split & Merge con límites 10 y 20 y área mínima 10.

Futuras extensiones:

- Permitir configurar la ventana de los histogramas
- Considerar como features de las imágenes también la textura.
- Implementar otra forma de medir la distancia entre histogramas que distancia euclidiana.
- Como la forma de los cuadrados indivisibles por el Split depende de las proporciones de la imagen original puede pasar que queden zonas de 1 pixel x N pixels que el algoritmo no pueda dividir. Sería bueno modificar la implementación para poder splitear también estas zonas.

Conclusiones

El método de segmentación Split and Merge inherentemente tiene en cuenta la cercanía de zonas a la hora de segmentar. En cambio, K-Means y Antipole deben considerarlo como parte de la característica.

El método que utiliza el Antipole Tree experimentalmente no presenta ventajas por sobre K-Means.

El Split and -Merge puede resutar útil para detectar zonas "grandes" en imágenes que no tienen una relación 1:1 entre ancho y alto, ya que este factor puede afectar bastante el resultado de la segmenta.ción para segmentos de tamaño pequeño.

Los filtros de suavizado (blur, max, min, max-min, average) tienden a facilitar la segmentación cuando hay zonas de la imagen que tienen una textura definida y no se considera tal característica como feature de la imagen. También el uso de filtros de dicho tipo tiende a eliminar la influencia de objetos pequeños o cambios abruptos que generan segmentos muy pequeños que habría que erosionar. También evitan que zonas que tengan cierta textura sobre-segmenten la imagen.

Para varias imágenes el uso de histogramas de color genera mejores resultados que el uso simplemente de color ya que si hay zonas muy pequeñas que tienen un gran cambio de color, evitan que el feature sea muy distinto y en consecuencia se generen zonas muy pequeñas.

Bibliografía

- Luo, M., Ma, Y., Zhang, H. "A spatial constrained K-Means approach to image segmentation"
- Yang, H., Lee, Sang. "Split-and-merge segmentation employing thresholding technique”
- Sural, S., Qian, G., Pramanik, S. "Segmentation and histogram generation using the hsv color space for image retrieval".
- Cantone, D., Ferro, A., Pulvirenti, A., Reforgiato, D. "Antipole tree indexing to support range search and k-nearest neighbor search in metric spaces".