

# Corrección de ojos rojos

# Contenido

1. Resumen del problema
2. Descripción del Algoritmo
3. Influencia de las variables
4. Limitaciones
5. Bibliografía

## Resumen del problema

Este algoritmo trata de resolver un problema muy común que ocurre en la fotografía, el efecto de ojos rojos. Esto sucede cuando se realiza una foto con flash en ambientes de poca luz. Para resolverlo, se aplicarán técnicas de procesamiento de imágenes, las cuales tratarán de diferenciar el efecto de otros rojos en la imagen.

Se utilizará el entorno Matlab con un número de imágenes en las que se presenta este efecto. Las imágenes serán de diferentes escenas y gamas de colores, para así poder poner a prueba la calidad del algoritmo. De este modo, se asegurará que el algoritmo es aplicable (o no) a diferentes tipos de imagen.

## Descripción del algoritmo

Este método hace uso del modelo de color HSV para encontrar la localización de los ojos. Las variables Hue y Saturation (Matiz y Saturación) indican el color del píxel y su intensidad. Por tanto, este modelo es de gran utilidad para encontrar un color concreto con una intensidad mínima. De este modo obtendremos una máscara como la siguiente:

Imagen original



Máscara

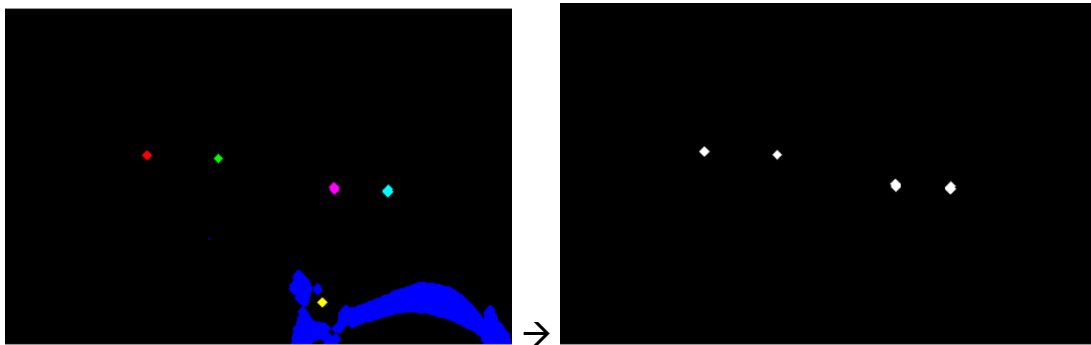


Sin embargo, la máscara posee numerosos puntos que no se identifican con ojos. Para solucionar esto es necesario aplicar dos técnicas, una muy sencilla y otra más complicada. La primera trata de eliminar los puntos más pequeños con una simple máscara sobre esta máscara. Aquí entra en juego un criterio del usuario, que debe definir un tamaño aproximado de los ojos, para que se pueda borrar los puntos blancos más pequeños que dicho tamaño. De este modo se obtiene la siguiente máscara:



Como podemos observar, los puntos pequeños se han eliminado. Sin embargo, siguen existiendo regiones que no se identifican con ojos en la imagen. Es necesario aplicar un filtrado de regiones para poder obtener la máscara definitiva.

De estas regiones, se calculará la media de las áreas, para luego eliminar aquellas que superen la media establecida y todo lo que se halle dentro de ellas (de este modo se eliminaría también posible ruido, como por ejemplo la region amarilla de la imagen). Obteniendo la siguiente imagen:



Finalmente, se aplica la máscara sobre la imagen para saber dónde se encuentran los ojos. Ahora basta con aplicar cierta corrección sobre esas regiones. En este algoritmo se ha decidido con reducir la intensidad del color Rojo a un término medio entre el color Azul y Verde. La imagen final obtenida es:



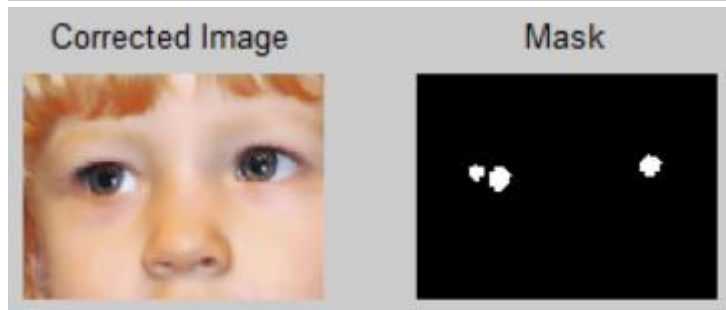
## Influencia de las variables

Existen tres parámetros que definen las bases del algoritmo. Estas variables poseen unos valores predeterminados que son los que se suelen ajustar a casi todas las imágenes. Sin embargo, a veces es necesario modificar estas variables para adaptarlos a imágenes más rebuscadas. Aun así, esto no determina un éxito asegurado por diferentes motivos (que se describen en el siguiente apartado). Cada variable influye de un modo diferente en el algoritmo:

- **sizeEye** → indica el tamaño de los ojos aproximado. Se utiliza en la primera limpieza de ruido y puede conllevar a una mejor definición del contorno del ojo y la eliminación del ruido si se ajusta bien, o todo lo contrario.



→ sizeEye = 1



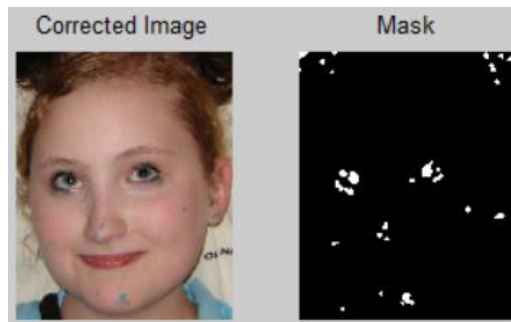
→ sizeEye = 2



→ sizeEye = 3

(satLevel = 0.4 en todas las imágenes)

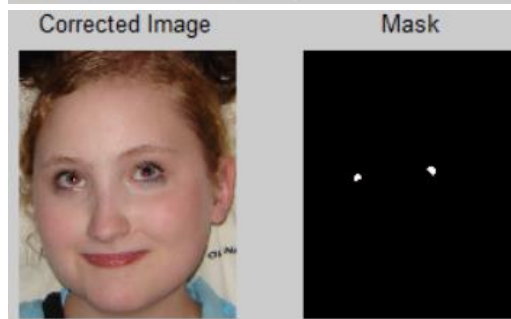
- **satLevel** → es el nivel mínimo de saturación de color rojo que se acepta a la hora de crear la máscara. Un nivel bajo acepta rojos más claros, y un nivel alto sólo acepta rojos más intensos. Los modos de intensidad baja pueden llegar a generar muchos falsos positivos, pero permiten que los ojos no necesiten ser demasiado rojos para que se les aplique la corrección.



→ satLevel = 0.3



→ satLevel = 0.5



→ satLevel = 0.9

## Limitaciones

Sin embargo, este algoritmo no funciona para todas las imágenes, ya que influye mucho de la posibilidad de generar una buena máscara. Esto depende de los siguientes factores:

1. **La intensidad o matiz de rojo de los ojos** → En caso de que los ojos tengan un matiz fuera del rango considerado rojo ( $0.9 < H < 0.02$ ), no se detectará el ojo con la máscara y, por tanto, no se podrá aplicar la corrección de ojos. Esto se puede solucionar modificando el rango del matiz, ajustándolo a la imagen a corregir.

2. **Intensidad del rojo de los ojos VS intensidad de rojo en otros objetos pequeños** → una baja intensidad de rojo, combinada con una intensidad superior en objetos rojos circulares, puede llegar a confundir el algoritmo y detectar ojos donde no los hay, sin ni si quiera llegar a detectar los ojos originales.
3. **Tamaño de los ojos** → Unos ojos demasiado pequeños pueden no ser detectados o eliminados en la primera fase de limpieza de ruido en la máscara. Por contra, unos ojos demasiado grandes con muchos objetos redondos podrían causar una media de regiones más pequeña que la región de los ojos (segundo paso del filtrado de la máscara), con lo que se descartaría la región de los ojos.

## Bibliografía

Se provee en la entrega un documento que se ha utilizado para desarrollar este algoritmo.