

Área de Ingeniería en Computadores

CE 5201 - Procesamiento y Análisis de Imágenes Digitales

Tarea 3 - Parte 1:

Manipulación de Espacios de Color

Estudiante:

Isaac Benavides Mata - 2017104821

Kevin Steven Cordero Zuñiga - 2018088034

Joseph Gonzalez Pastora - 2017216987

Profesor:

Luis Chavarría Zamora

II Semestre 2021

Selective Color

Este algoritmo consiste en resaltar un color en particular en una imagen. El color elegido destaca en la imagen ya que los demás colores se desaturan, ocasionando que la imagen sea a escala de grises con excepción del color elegido (Mishra, n.d.).

Este algoritmo hace uso del modelo de colores *HSV*, el cual es un modelo cilíndrico que remapea los colores *RGB* con el fin de que sean más fáciles de entender para las personas. *H* representa el matiz (*hue*), que indica el ángulo del color (0° es rojo, 120° es verde y 240° es azul), *S* representa la saturación, la cual indica la cantidad de color utilizado y *V* representa el valor, el cual controla brillo de la imagen (*Color Models and Color Spaces*, n.d.).

Adicionalmente, este algoritmo también hace uso del método de *Otsu*, el cual es utilizado para binarizar imágenes con base en las intensidades de los píxeles (Murzova & Seth, 2020).

En la imagen #1, se puede observar un ejemplo del algoritmo selective color, donde se resalta el color amarillo:

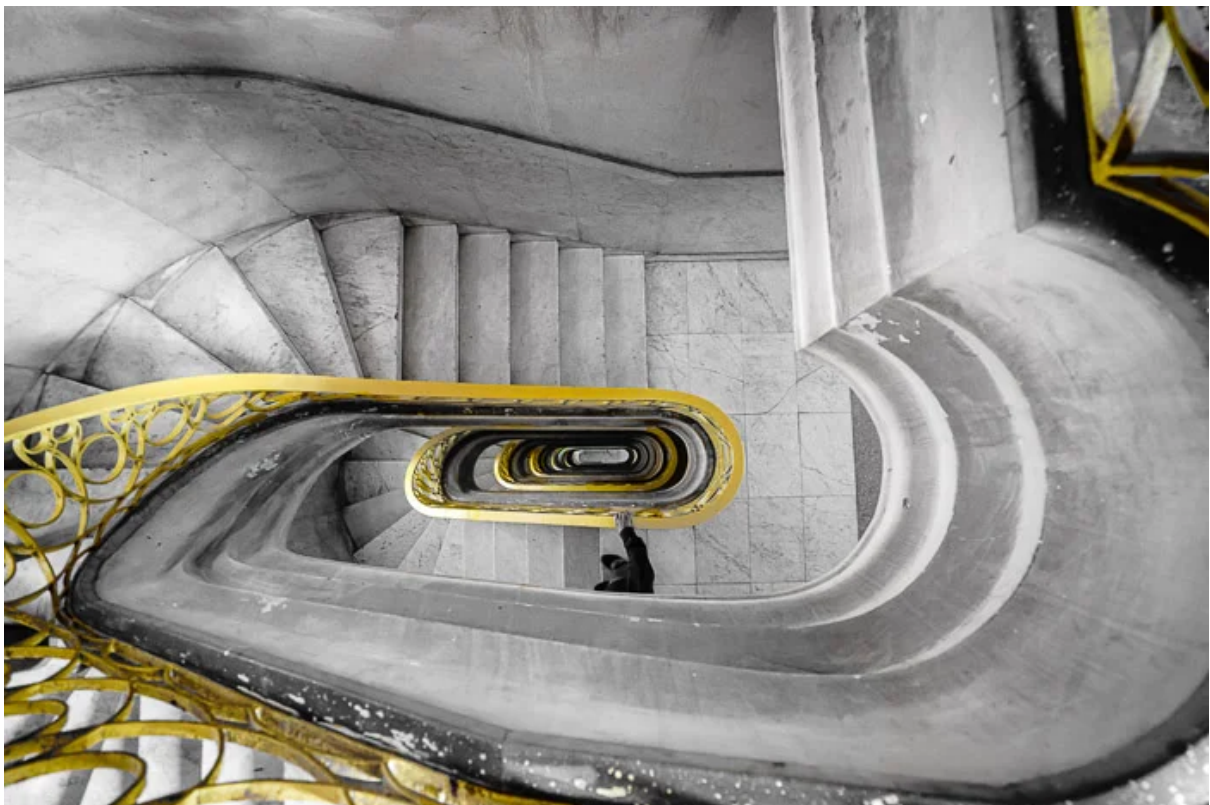


Imagen #1. Ejemplo del Algoritmo Selective Color (Mishra, n.d.).

Pseudocódigo:

- Paso 1: Se lee la imagen (I), la cual se utilizará para aplicar el algoritmo.
- Paso 2: Se convierte la imagen (I) a escala de grises (I_g).
- Paso 3: Se convierten los canales RGB de la imagen a HSV .
- Paso 4: Se obtiene una máscara binarizada (Im) de la imagen (I) que divide los elementos en el color seleccionado y los demás colores:
 - Paso 4.1: Se obtiene la máscara (Im) del color elegido según el ángulo de este en el modelo HSV : $Im = |I_h - (cA / 360)|$, donde I_h es el componente H de la imagen y cA es el ángulo del color elegido (según el cilindro HSV).
 - Paso 4.2: En caso de que el color elegido **no** sea rojo, se debe realizar un ajuste a la máscara, ya que ningún valor de Im puede ser mayor que 0.5, sino esto puede ocasionar errores con el método de *Otsu*:
 - Paso 4.2.1: Se obtiene una matriz Id con todos los valores mayores a 0.5 de Im .
 - Paso 4.2.2: Se restan los valores obtenidos con los originales:
$$Im = |Id - Im|.$$
 - Paso 4.3: Se aplica el método de *Otsu* con tal de obtener el *threshold* de la imagen Im .
 - Paso 4.4: Se binariza la imagen Im utilizando el *threshold* obtenido.
- Paso 5: Se utiliza la máscara para definir cuál color se debe mantener de la imagen original y cuáles colores pasarán a ser desaturados:
$$I_{sc} = I * (1 - Im) + I_g * Im.$$

Selective Color Transfer

Este algoritmo es similar al anterior, la diferencia radica en que el color elegido, será reemplazado por otro. En la imagen #2, se puede observar un ejemplo donde se cambia el color de una mochila de rojo a verde:



Imagen #2. Ejemplo del Algoritmo Selective Color Transfer (Mireles, n.d.).

Pseudocódigo:

- Paso 1: Se lee la imagen (I), la cual se utilizará para aplicar el algoritmo.
- Paso 2: Se convierten los canales RGB de la imagen a HSV .
- Paso 3: Se obtiene una máscara binarizada (Im) de la imagen (I) que divide los elementos en el color seleccionado y los demás colores:
 - Paso 3.1: Se obtiene la máscara (Im) del color elegido según el ángulo de este en el modelo HSV : $Im = |I_h - (iCA / 360)|$, donde I_h es el componente H de la imagen y iCA es el ángulo del color elegido inicial (según el cilindro HSV).
 - Paso 3.2: En caso de que el color elegido **no** sea rojo, se debe realizar un ajuste a la máscara, ya que ningún valor de Im puede ser mayor que 0.5, sino esto puede ocasionar errores con el método de *Otsu*:
 - Paso 3.2.1: Se obtiene una matriz Id con todos los valores mayores a 0.5 de Im .
 - Paso 3.2.2: Se restan los valores obtenidos con los originales:
 $Im = |Id - Im|$.
 - Paso 3.3: Se aplica el método de *Otsu* con tal de obtener el *threshold* de la imagen Im .
 - Paso 3.4: Se binariza la imagen Im utilizando el *threshold* obtenido.
- Paso 4: Se suma el ángulo del color nuevo a los valores de la matriz I_h : $I_h = (I_h + (oCA / 360)) \% 1$, donde oCA es el ángulo del color final y adicionalmente, se aplica el módulo de uno, con tal de que los valores se encuentren en el rango $[0, 1]$.
- Paso 5: Se genera la imagen modificada I_{mod} con el color nuevo en el modelo RGB .
- Paso 5: Se utiliza la máscara para definir cuáles colores se deben mantener de la imagen original y cuál color será modificado:
 $I_{sct} = I * Im + I_{mod} * (1 - Im)$.

Croma o Clave de Color (Chroma Key)

Este algoritmo es similar a los dos anteriores. La mayor diferencia radica en que este algoritmo realiza la composición de dos o más imágenes. Esto es útil para cambiar fondos de imágenes o videos, además es muy utilizado en la industria del cine (Yeager, 2019).

En la imagen #3, se puede observar la imagen de una periodista a la que se le aplica esta técnica para cambiar el fondo verde a la imagen de un país con tal de mostrar las condiciones meteorológicas de este:

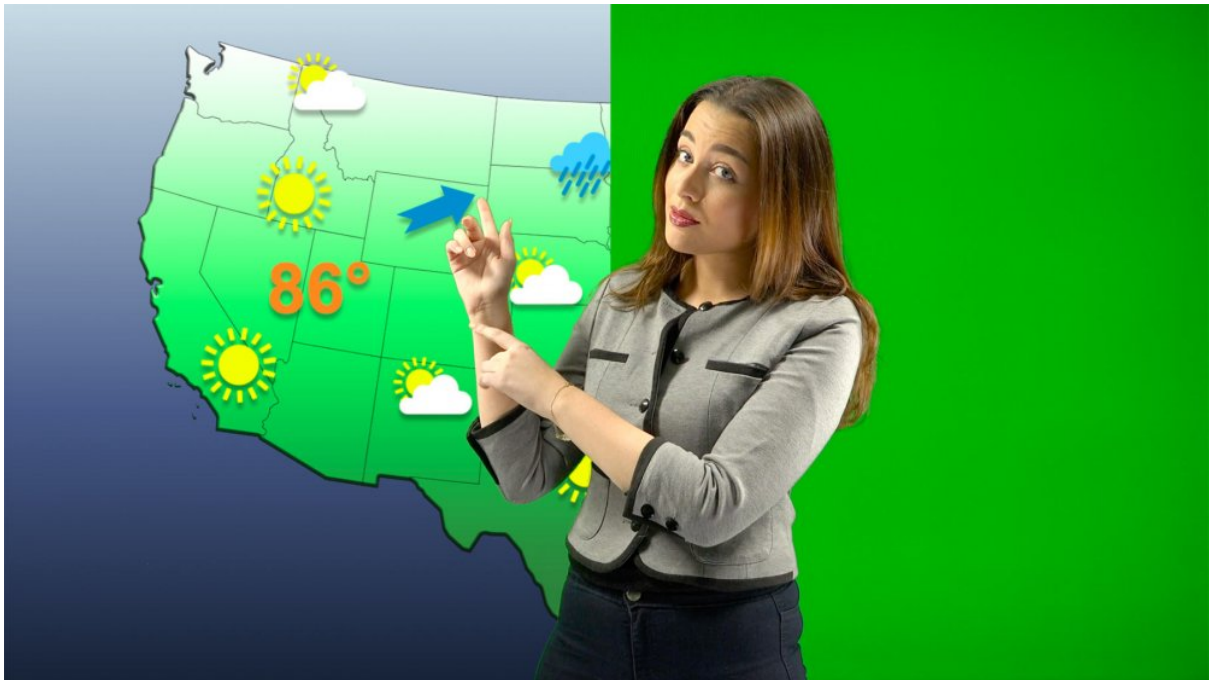


Imagen #3. Ejemplo del Algoritmo Chroma Key (*Chroma Keying (Green Screen)*, n.d.).

Pseudocódigo:

- Paso 1: Se lee la imagen (I), la cual se utilizará para aplicar el algoritmo.
- Paso 2: Se convierten los canales RGB de la imagen a HSV .
- Paso 3: Se obtiene una máscara binarizada (Im) de la imagen (I) que divide los elementos en el color seleccionado y los demás colores:
 - Paso 3.1: Se obtiene la máscara (Im) del color elegido según el ángulo de este en el modelo HSV : $Im = |I_h - (iCA / 360)|$, donde I_h es el componente H de la imagen y iCA es el ángulo del color elegido inicial (según el cilindro HSV).
 - Paso 3.2: En caso de que el color elegido **no** sea rojo, se debe realizar un ajuste a la máscara, ya que ningún valor de Im puede ser mayor que 0.5, sino esto puede ocasionar errores con el método de *Otsu*:
 - Paso 3.2.1: Se obtiene una matriz Id con todos los valores mayores a 0.5 de Im .
 - Paso 3.2.2: Se restan los valores obtenidos con los originales:
$$Im = |Id - Im|.$$
 - Paso 3.3: Se aplica el método de *Otsu* con tal de obtener el *threshold* de la imagen Im .
 - Paso 3.4: Se binariza la imagen Im utilizando el *threshold* obtenido.
- Paso 4: Se utiliza la máscara para definir cuáles colores se deben mantener de la imagen original y cuáles colores se deben mantener del fondo aplicado (Ib): $I_{ck} = I * Im + Ib * (1 - Im)$.

Referencias

Chroma Keying (Green Screen). (n.d.). mimoLive. Retrieved November 19, 2021, from <https://mimolive.com/manual/5/en/topic/chroma-keying-green-screen2>

Color models and color spaces. (n.d.). Programming Design Systems. Retrieved November 19, 2021, from <https://programmingdesignsystems.com/color/color-models-and-color-spaces/index.html>

Mireles, A. (n.d.). *The EASY Way to Change a Color in Photoshop 2021*. Shotkit. Retrieved November 19, 2021, from <https://shotkit.com/change-color-object-photoshop/>

Mishra, J. (n.d.). *How to Use Selective Color in Photoshop (Step by Step!)*. ExpertPhotography. Retrieved November 19, 2021, from <https://expertphotography.com/selective-colour-photoshop/>

Murzova, A., & Seth, S. (2020, August 5). *Otsu's Thresholding Technique*. LearnOpenCV. Retrieved November 19, 2021, from <https://learnopencv.com/otsu-thresholding-with-opencv/>

Yeager, C. (2019, July 29). *Everything You Need to Know About Chroma Key and Green Screen Footage - The Beat: A Blog by PremiumBeat*. PremiumBeat. Retrieved November 19, 2021, from <https://www.premiumbeat.com/blog/chroma-key-green-screen-guide/>