

Implementación de un Algoritmo para la Colorización de Video a Partir de una Imagen Muestra

Implementation of an Algorithm for the Colorization of Video from a Sample Image

Canché Chan, Y.A.¹, Capetillo Loeza, A.¹, Sandoval Ramírez, N.A.¹,
Moreno Sabido, M.R.², Hernández López, F.J.²

¹ Depto. de Sistemas y Computación, Instituto Tecnológico de Mérida
Av. Tecnológico Km 4.5, s/n, AP 911, 97118 Mérida, Yucatán. México

² Depto. de Ciencias de la Computación, Centro de Investigación en Matemáticas A.C.
PCTY, Sierra Papacal 97302 Mérida, Yucatán. México

¹yamili1693@gmail.com, april.mraz94@gmail.com, alesan_090@hotmail.com,
mario@itmerida.mx, ²fcoj23@cimat.mx

Fecha de recepción: 15 de junio 2017

Fecha de aceptación: 2 de febrero 2018

Resumen. En este artículo se propone la utilización de la técnica de colorización de video en escala de grises mediante la adaptación del algoritmo de colorización de imágenes. La colorización es una técnica fotográfica que se basa en transferir colores a una imagen en escala de grises, sepia o monocromática; en este trabajo se propone colorizar un video tomando el color de una imagen muestra, donde se comparan las luminancias de la imagen muestra con el video. La principal contribución de este trabajo consiste en la implementación del algoritmo para colorizar un video a partir de una imagen muestra, ya que anteriormente a esta investigación, esta técnica sólo funcionaba para imágenes. Los resultados de las pruebas muestran una colorización homogénea frame a frame, por secciones de video.

Palabras Clave: Colorización, Procesamiento de Imágenes, Procesamiento de Video.

Summary. This article proposes the use of the video colorization technique in grayscale by adapting the image coloring algorithm. Colorization is a photographic technique that is based on transferring colors to an image in grayscale, sepia or monochrome; in this paper is proposed coloring a video taking the color of a sample image, where the luminance of the image is compared with the video. The main contribution of this work is the implementation of the algorithm to coloring a video from a sample image, since prior to this research, this technique only worked for images. The results of the tests show a homogeneous colorization frame to frame, by sections of video.

Keywords: Colorization, Image Processing, Video Processing.

1 Introducción

La colorización es una técnica fotográfica que se basa en transferir colores a una imagen en escala de grises, sepia o monocromáticos [1].

Se le llama colorización al proceso asistido de añadir color a una imagen monocromática o una película por medio de un ordenador. El proceso implica la segmentación de imágenes en ciertas regiones y el seguimiento de estas a través de secuencias.

Inicialmente, la técnica de colorización inicia con la separación de la imagen o las secuencias de imágenes en regiones en las cuales se asume que se tiene el mismo color. A pesar de todas las desventajas que se pueda tener, en los últimos años se han desarrollado varios y diferentes tipos de algoritmos para la solución de este problema.

Los algoritmos que en la actualidad existen aún no están diseñados para colorear varios cuadros que tengan las mismas probabilidades o intensidades; debido a esto, el colorear un video se hace aún más complicado, ya que un video contiene más de 100 cuadros diferentes que no varían en casi nada, haciendo que el trabajo se vuelva tedioso y extenso al momento de colorizar cada cuadro, y dificultando seguir la secuencia del video.

Debido a la falta de software para la colorización de video, este trabajo de investigación será de gran utilidad, ya que se pretende implementar un algoritmo usando métodos y técnicas tanto sencillas como eficientes, para obtener una colorización coherente a través de toda la secuencia de video, y así poder disminuir los costos de manufactura y económicos que se tienen en la actualidad, debido a que se agilizará el procedimiento de colorizar un video.

2 Estado del arte

En esta sección se presenta el algoritmo para colorizar una imagen, que es la técnica que se implementó en la colorización de video que se describe en este artículo.

2.1 Transferencia de color a imágenes en escala de grises

El algoritmo busca minimizar la intervención humana y elige los colores de una paleta RGB para colorear la imagen. Compara la iluminación y la textura entre las imágenes y podrá aplicarse a varias imágenes/videos

mientras que la textura e iluminación se puedan distinguir.

Entre las tareas que deberá realizar, está el asignar valores a los píxeles en los canales RGB, ya que diferentes colores pueden tener el mismo valor, pero podrán variar en matiz o saturación. El método comparará la distribución de las 3 dimensiones de color entre las imágenes, y luego transformará la distribución de color en la imagen final para compararla con la distribución en la imagen original.

Se deberá tener en cuenta que un sólo valor de iluminación puede representar partes completamente diferentes de la imagen, y una vez que el píxel se compare, la información será transferida sin modificar el valor original.

En el caso de los videos, el procedimiento es similar, consta de la aplicación del algoritmo sobre un solo cuadro de la secuencia, y después se comparará cada frame con el primero, siempre y cuando los objetos en la imagen no cambien demasiado.

La imagen deberá convertirse al espacio de color $\alpha\beta$ donde se realizará un muestreo para elegir el píxel base; se realiza el recorrido para cada píxel en la imagen en escala de grises en un orden lineal para seleccionar la mejor muestra que se asemeje, y así los valores de $\alpha\beta$ puedan ser transferidos a la imagen final.

El espacio de color $\alpha\beta$ provee 3 canales no correlativos que corresponden a uno acromático que será el canal de iluminación, y 2 canales cromáticos $\alpha\beta$ [2].

3 Metodología usada

En esta sección se presenta una descripción general del algoritmo, así como el proceso de implementación del mismo; la herramienta que se utilizó para el desarrollo fue la plataforma Matlab.

Se analizó el algoritmo y se llegó a la determinación de realizar 4 pasos para obtener la imagen resultante, los cuales se presentan a continuación:

3.1 Cambio de espacio de color

En primera instancia se eligieron aleatoriamente las imágenes para realizar la prueba; en este primer intento se utilizó la misma imagen a color la cual se llamó "Im", convirtiéndola a escala de grises con la función "rgb2gray"; a esta imagen se le nombró "It".

El primer paso del algoritmo fue convertir las imágenes "Im" e "It" al espacio de color " $\alpha\beta$ ", mediante la función "rgb2ycbcr" llamando a las imágenes "Im_lab", "It_lab" respectivamente, mostrándolas en pantalla mediante la función "figure" para crear la ventana, y la función "imagesc (nombre_de_la_imagen)" para mostrarla.

3.2 Jittered sampling

El segundo paso se realizó a través del método "jittered sampling"; este método trata de una selección de "números", en este caso píxeles de forma aleatoria; se tomó una muestra aleatoria de 200 píxeles de la imagen muestra denominada "Im".

La muestra aleatoria se seleccionó mediante un ciclo for que a continuación se presenta en el fragmento de código:

Algoritmo 1. Obtención de 200 muestras aleatorias por el método Jittered Sampling.

```
%Número de píxeles sub-muestra
M=200;

%Para guardar las posiciones de los píxeles muestra elegidos de forma
%aleatoria
D=zeros(M,2);
for i=1:M
    D(i,1)=uint16(rand*(imageW-1)+1);
    D(i,2)=uint16(rand*(imageH-1)+1);
end
```

Para obtener el tamaño de las imágenes se utilizó la función "[imageH, imageW]=size(It);" siendo imagenH el número de filas que hay en la imagen, y imageW el número de columnas. En la variable "M" se guardó el número de píxeles que se requieren para obtener la muestra. En la matriz "D" se guardaron las coordenadas de los píxeles que se eligieron al azar. El ciclo for va de 1 a M, siendo M la cantidad de píxeles que se necesitaron; en la primera posición de la matriz se guardaron las coordenadas de las columnas, y en la segunda posición se guardaron las coordenadas de las filas.

A partir de que se obtuvieron los pixeles aleatorios se sacó el promedio y la varianza de cada pixel, mediante la fórmula:

$$S(i) = [\mu_s(i), \sigma_s^2(i)], \quad (1)$$

Siendo “S” una matriz de 2 x 200 donde se guardarán los promedios y las varianzas. Para sacar el promedio y la varianza de cada pixel se creó una ventana de “W x W”; en los experimentos que se realizaron el tamaño de “W” fue de “3, 5, 7, 9, 21”.

En la Figura 1 se representa la forma en la que se genera la ventana alrededor del pixel el cuál se va a testear.

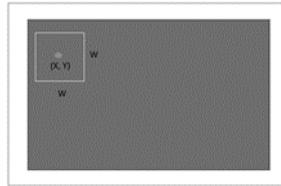


Figura 4. Representación de la creación de la ventana para testear el pixel.

La fórmula con la cual se buscó el promedio es la siguiente:

$$\mu_s(i) = \frac{1}{W * W} \sum_{k1=-\frac{W}{2}}^{\frac{W}{2}} \sum_{k2=-\frac{W}{2}}^{\frac{W}{2}} [l_m(x - k1, y - k2)]. \quad (2)$$

La fórmula con la cual se buscó la varianza es la siguiente:

$$\sigma_s(i) = \frac{1}{W * W} \sum_{k1=-\frac{W}{2}}^{\frac{W}{2}} \sum_{k2=-\frac{W}{2}}^{\frac{W}{2}} [l_m(x - k1, y - k2) - \mu_s(i)]^2. \quad (3)$$

3.3 Comparación de varianzas y promedios

En el tercer paso del algoritmo, se recorrió cada pixel de la imagen “It_lab” y se sacó el promedio y la varianza; una vez que se obtuvieron se hizo un “matching” (el valor que más se asemeje) con los promedios y varianzas de los pixeles aleatorios de la imagen muestra “Im_lab”, y se seleccionó el que tenga menor distancia entre cada promedio y varianza.

Se realizó una estrategia para ayudar a identificar el mejor “matching”, la cual fue llevada a cabo mediante la siguiente fórmula:

$$error(i) = [\mu_t(x, y) - \mu_s(i)]^2 + [\sigma_t^2(x, y) - \sigma_s^2(x, y)]^2, \quad (4)$$

Donde la μ_t y la σ_t se compararon con los 200 promedios y varianzas, y se seleccionó el que tenga menor distancia entre promedios y varianzas; una vez que se obtuvieron los pixeles que más se asemejen, se tomaron las coordenadas de dicho pixel, y se tomó la crominancia de éste y se le asignó al pixel que se testeó.

3.4 Transferencia de crominancias

En el cuarto paso se transfirió la crominancia de la imagen muestra “Im” a la imagen que se quiere colorear “It”, y se creó una nueva imagen llamada “It_final”; después de que se terminaron de recorrer todos los pixeles se convirtió la imagen a una imagen RGB “It_final” mediante la función “ycbcr2rgb” llamando a la imagen “It_resultante”.

Se realizó una estrategia que se encuentra en el trabajo Welsh et. Al [2] Transferring Color Grey Scale Images, el cual hace mención del trabajo de Hertzmann et al [3] en donde se hace un re-mapeo de las luminancias de las imágenes.

Dicha estrategia se presenta en la función “LuminanceRemapping”, la cual permite sacar el promedio de luminancia de toda la imagen para lograr que la luminancia quede uniforme en toda la imagen.

Se sacó la varianza y el promedio global de las imágenes respectivamente para poder usar la función “var” y “mean” predefinida en Matlab. Se convirtió la matriz de la imagen en un vector con la función “NuevoVector = matriz (:);” después de calcular la varianza y promedio de cada imagen, se realizó el re-mapeo.

Como segunda instancia se probó el algoritmo para colorear un video en escala de grises, tomando una única imagen muestra para un video.

El proceso que se realizó con una sola imagen, se utilizó para cada frame del video; como primer paso se contaron los frames que contiene el video con la función “nframes=get (VideoMuestra, 'numberOfFrames');” la cual es una función propia de Matlab; después de conocer el número de frames se creó un ciclo for que va de 1 a n frames (cantidad de frames en el video); dentro de este ciclo se encuentra los cuatro pasos vistos anteriormente en esta sección; antes del ciclo se creó el video final con la función “videoFinal=VideoWriter('videofinal.avi','Motion JPEG AVI'); open(videoFinal);”; dicha función crea un video final en donde se van guardando los frames ya coloreados; se separa cada frame con la función “Iframe=read (VideoMuestra, F);” donde F es el contador del ciclo for para al momento de crear el video final tenga la misma secuencia que el video original.

Después de obtener la imagen resultante, se guarda en el video final con la función “writeVideo (videoFinal, Img_resultante);”; cuando el ciclo for termine, se cierra el video para que se guarde, y se reproduce el video.

4 Resultados experimentales

La primera prueba fue donde la imagen muestra y la imagen “It” fueron las mismas. En la Figura 2 se muestra el correcto coloreado de la imagen final, quedando del mismo color que la imagen muestra.

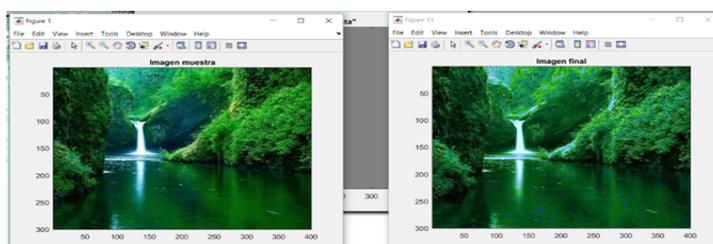


Figura 5. Resultados transferencia de color.

Para la colorización de video se implementaron conocimientos obtenidos en el artículo “Transferring Color Grey Scale”; al igual que con las imágenes se ingresó primero una imagen muestra a colores, y después el video que se requiera colorizar. El resultado de estas pruebas se puede ver en la Figura 3 donde se muestra la imagen que se utilizó de muestra para colorear el vídeo; en la Figura 4 se muestran algunos de los 83 frames que contiene el video, observando una colorización con movimiento; como se puede observar, a partir del frame 13 se torna en colores azules, esto debido a que las intensidades se asemejan más a las intensidades del color azul de la imagen muestra, por lo que se torna de ese tono.



Figura 6. Imagen muestra (Prueba 1).



Figura 7. Frames del video colorizado con transferencia de color (Prueba 1).

Otros de los resultados que se obtuvieron en las pruebas realizadas se ilustran en las Figuras 5 y 6, en donde la Figura 5 es la imagen muestra, y la Figura 6 son algunos de los 80 frames que contiene el video; como se puede observar, se realizó una colorización con movimiento, donde los frames se empiezan a tornar en tonos rojos; los primeros frames se tornan de un rojo pálido, y conforme avanza la secuencia del video las intensidades van cambiando tornándose de un rojo más fuerte.



Figura 8. Imagen muestra (Prueba 2).



Figura 9. Frames del video colorizado con transferencia de color (Prueba 2).

Se realizó una tercera prueba dejando como resultados los señalados en las Figuras 7 y 8, donde en la Figura 7 se presenta la imagen muestra con la que se coloreó el video, y en la Figura 8 se muestran algunos de los 179 frames que contiene el video, en donde se puede apreciar la colorización en movimiento la cual se torna de tonos azules, verdes y rosas por las diferentes luminancias que contiene el video.



Figura 10. Imagen muestra (Prueba 3).

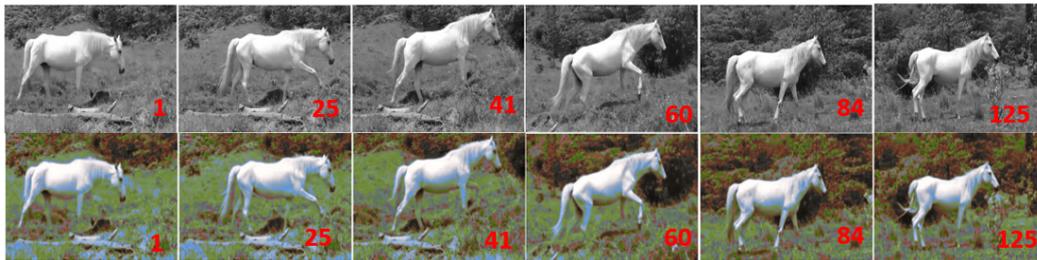


Figura 11. Frames del video colorizado con transferencia de color (Prueba 3).

5 Conclusiones y trabajos futuros

Como se investigó anteriormente, se sabe que la colorización de vídeo es costosa y complicada, ya que, a diferencia de la colorización de imágenes, se sigue una secuencia de frames donde puede haber una deformación del objeto que se está coloreando lo que necesita de la intervención humana, haciendo más costoso el trabajo hombre; en este artículo se propone una técnica que disminuye el trabajo-hombre.

Tras haber realizado y analizado el método de colorización, se obtuvieron los resultados esperados en cuanto a los objetivos propuestos al iniciar el proyecto, los cuales se basaban principalmente en hacer una colorización siguiendo la secuencia de un vídeo; se observó una colorización en un 90% de satisfacción, ya que hay homogeneidad frame a frame por secciones del vídeo. Durante las pruebas, se notó que las imágenes muestra deben tener una intensidad parecida a la del vídeo, ya que es con las luminancias que se realiza la colorización.

Como trabajo futuro se puede realizar el método de “Swatches” propuesto en el artículo “Transferring color grey scale” [2], el cual se basa en seleccionar regiones de la imagen muestra; estas regiones seleccionadas pasarían el color a las secciones seleccionadas de la imagen en escala de grises.

Agradecimientos.

Se agradece al Tecnológico Nacional de México (TecNM), Instituto Tecnológico de Mérida (ITM) y Centro de Investigación en Matemáticas A.C. Unidad Mérida (CIMAT A.C. Unidad Mérida) por el valioso apoyo brindado para la realización de este proyecto.

Referencias

- 1 Dalmau, O.; Rivera, M.; Mayorga, P: "Computing the a-Channel with probabilistic Segmentation for Image Colorization" *Computer Vision Workshops (ICCVW) 2013 IEEE International Conference on*, pp. 860-867, 2013. (2007).
- 2 Welsh, T.; Ashikhmin, M.; Mueller, K. “Transferring color to greyscale images” *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 277-280 (2002).
- 3 Hertzmann, A., Jacobs, C., Oliver, N., Curless, B., Salesin, D., *Image Analogies*, In *Proceedings of ACM SIGGRAPH 2002*, 341-346, (2001)