Trabajo 1 – Efectos

## Visión por Computador – 2019/2020

### Francisco Ferraz (737312) – Ignacio Galve (735508)

# Resumen

En este trabajo práctico se han implementado 4 técnicas de procesamiento de imágenes en vivo utilizando OpenCV y C++ de forma sencilla y eficiente.

# Contraste

Para la mejora del contraste de la imágen se ha realizado un ajuste de la luminosidad de la escena mediante la corrección de la ganancia y el sesgo.

Por cada *frame* recuperado, éste ha sido transformado del espacio de color estándar RGB a uno que tiene en cuenta la luminosidad, en este caso L\*a\*b\*. También se hicieron pruebas con HSV, pero los resultados eran menos notables.

Tras la conversión, se separan los canales de la imágen en matrices de valores separadas y, sobre la matriz correspondiente a la luminosidad, se aplica una corrección utilizando la fórmula:



Donde *a* es la ganancia y *b* el sesgo.

A contuniación se presentan varias imágenes probando distintos valores:

Original



Mejorada (alpha = 1.5, beta = 30):



Mejorada (alpha = 1.5, beta = 10):



En cuanto a la ecualización del histograma, se ha aprovechado la función previamente implementada en OpenCV *equalizeHist*, que devuelve una imágen con el histograma ecualizado. A continuación, una comparativa de las distintas imágenes con sus histogramas:

Original



Ecualizda



# Alien

El filtro de color *alien* cambia el color de la piel de una persona a otro establecido. Para ello se debe detectar qué zona de la imágen es piel y qué zona no, y posteriormente aplicar un filtro sobre los píxeles indicados.

La fórmula de la detección de piel ha sido extraída del paper de Kolkur et al (1). por ser la más precisa encontrada en base a los espacios de color y métodos estudiados en la asignatura.

En el documento se concluye que, utilizando una mezcla de espacios de color HSB y RGBa se puede detectar la piel de manera óptima utilizando:

*0.0 <= H <= 50.0 and 0.23 <= S <= 0.68 and R > 95 and G > 40 and B > 20 and R > G and R > B and | R - G | > 15 and A > 15*

Estos valores son medianamente dependientes de la cámara utilizada, por lo que se ha tenido que utilizar una webcam para que los resultados fueran convincientes. En cualquier caso, se demuestrán a continuación los efectos resultantes:

//TODO

# Póster

El efecto de póster se basa en reducir el espacio de color de una imágen. Para lograrlo, se ha implementado una función que, para cada canal de color de una imágen (RGB), elige un factor de división D de 1 a 256 y reduce el número de valores de cada canal a *256/D*.

Para cáda píxel de la imágen, se hace:

data[i] = data[i] / divisor \* divisor + divisor / 2;

Aproximando cada color a su valor más cercano dentro de los establecidos. Aquí algunas imágnes resultantes.

Original



2 colores por canal:



4 colores por canal:



# Distorsión

Finalmente, se han implementado tanto la distorsión de barril como la de cojín. Las fórmulas de ambas se han recuperado del desarrollo realizado en Wang et al (2).

Primeramente, se realiza una normalización de la posición de cada píxel en la imagen obteniendo su valor sobre un rango 0 – 1.

Considerando *x* e *y* los índices del píxel en la imagen, se realiza una distorisión radial respecto de la posición tal que:



Siendo *k1* y *k2* los índices de distorsión. Finalmente se desnormaliza la posición para obtener la imágen final.

A pesar de que matemáticamente sean operaciones similares cambiando la polaridad de un coeficiente respecto del otro, se han implementado dos funciones por separado. Una para el filtro de barril y otra para el de cojín. Aquí se muestran algunos resultados:

Barril (K1 = 55, K2 = 95):



Cojín (K1 = 40, K2 = 50):



# Referencias

1. Human Skin Detection Using RGB, HSV and YCbCr Color Models, S. Kolkur et al.
2. A new calibration model of camera lens distortion, J. Wang et al.