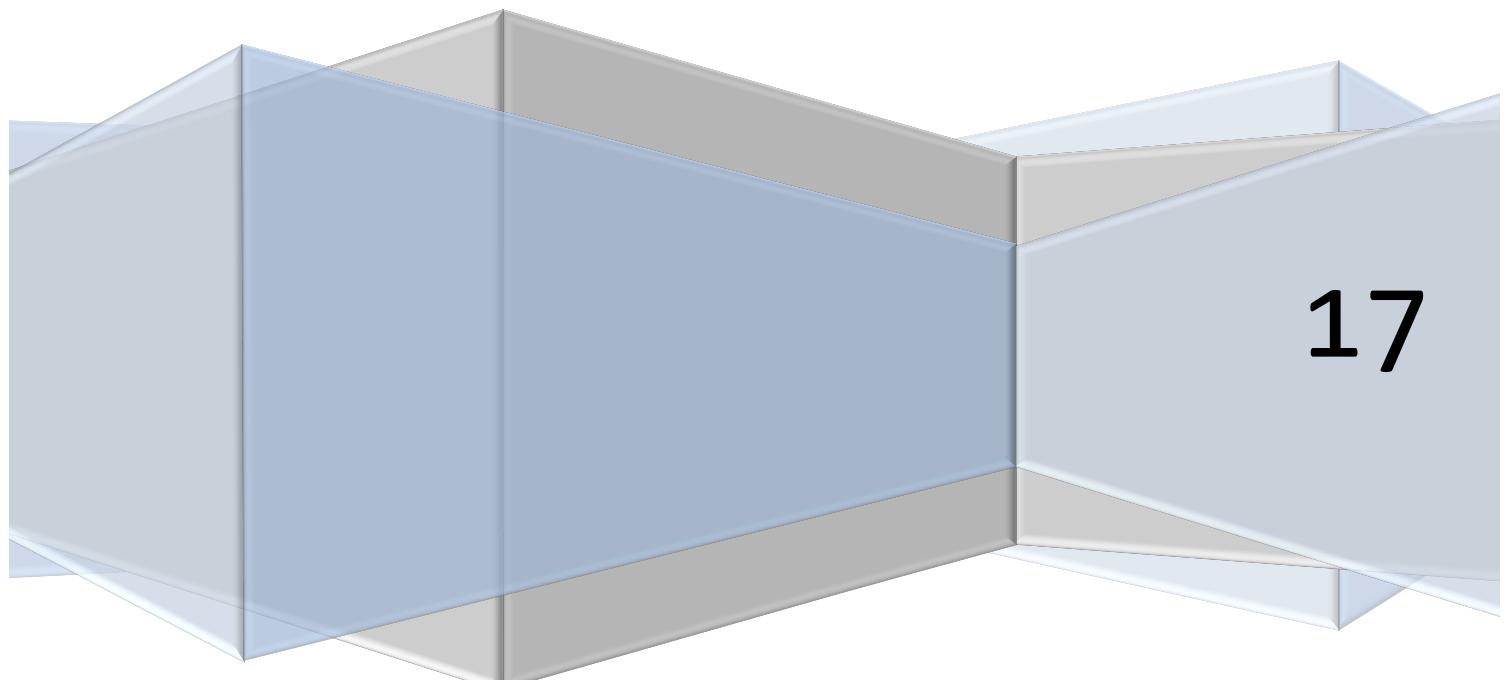


Trabajo 1

Efectos

Jorge Andrés - 679155

Javier Aranda - 679184



INTRODUCCIÓN

Para este trabajo se ha desarrollado en “opencv” una aplicación capaz de aplicar diferentes efectos a las imágenes captadas en vivo por la cámara web de un ordenador. Estos efectos se pueden elegir entre un menú de operaciones que asocia una letra del teclado a cada uno, pudiendo fijar además algunos parámetros a gusto del usuario para conseguir los resultados requeridos.

Para poder realizar el trabajo, lo primero que se hizo fue capturar imágenes en vivo a través de la cámara, para lo cual simplemente se empleó un bucle que capturase y mostrase continuamente imágenes de la cámara. En dicho bucle, además, se emplea la función “waitkey”, que permite saber que teclas se han pulsado, para poder aplicar los efectos deseados por el usuario (menú de operaciones comentado anteriormente).

EFFECTOS

A continuación, se describirán los diferentes efectos incluidos en la aplicación y la forma en que se han implementado. Se han realizado aquellos exigidos por el enunciado de la práctica, añadiendo, además, una serie de efectos extra que parecieron a los dos miembros del grupo de prácticas interesantes, consiguiendo así un mejor manejo de “opencv” para futuros trabajos de la asignatura.

Contraste y ecualización de histograma

Para mejorar el contraste de las imágenes captadas por la cámara, se realizaron dos métodos principales. El primero de ellos, y además el más sencillo, consistía en multiplicar el valor de cada píxel por un escalar (se ha elegido el valor 2) y sumarle otro escalar para ajustar el histograma (10), consiguiendo de este modo una mayor diferencia entre los píxeles claros y los oscuros, apreciándose más claramente el contraste.

$$g(\mathbf{x}) = af(\mathbf{x}) + b$$

Para obtener los valores ideales para las imágenes, se ha mostrado por pantalla en tiempo real los histogramas de la imagen con los tres canales, observando también el resultado obtenido al aplicarle un cierto escalar y desplazarlo por otro. De este modo, se han obtenido los valores más apropiados, y mediante la función “convertTo” de “opencv”, se ha aplicado dichos valores a la imagen para obtener el resultado final.

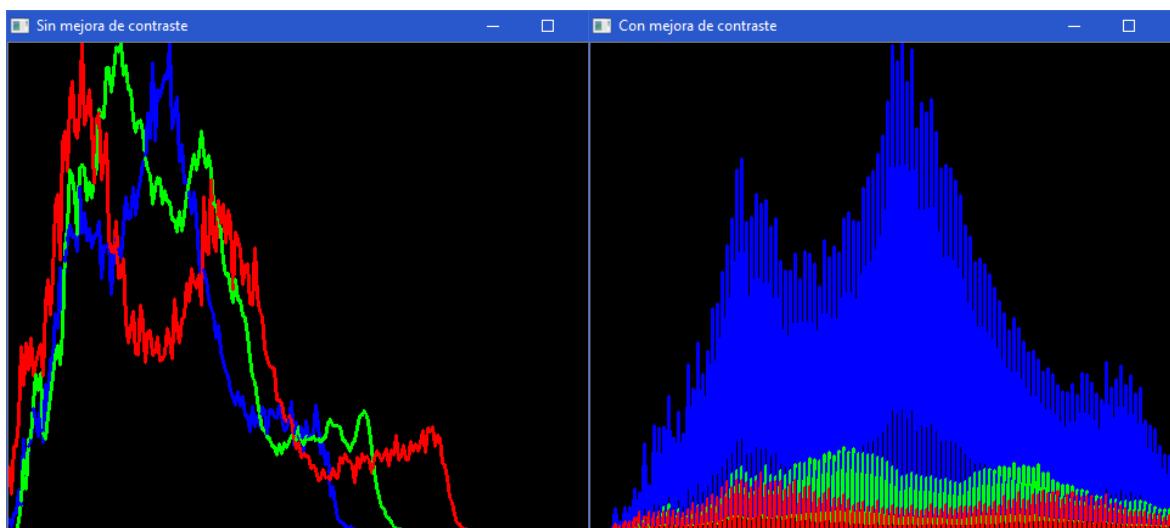
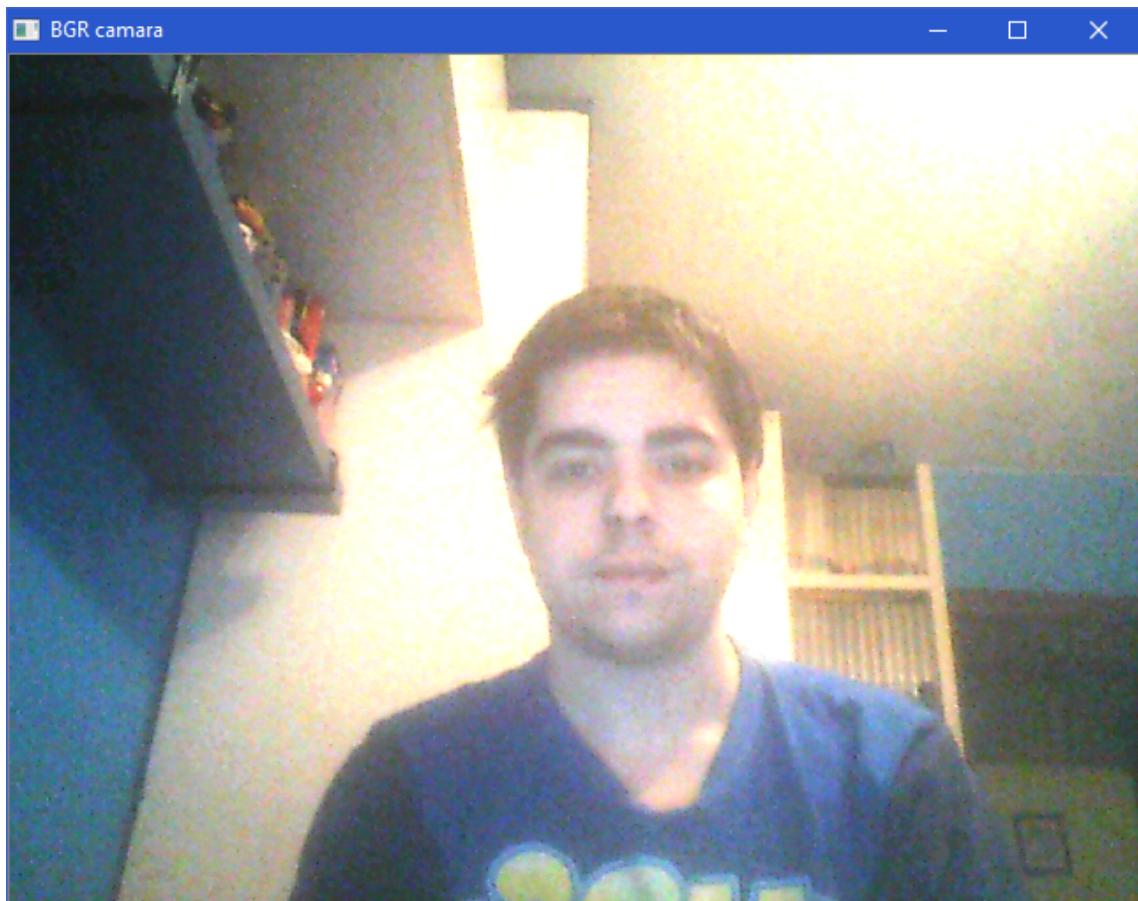


Ilustración 1.- Imagen con contraste e histogramas.

Además, se implementó otro de los métodos comentados en las clases de teoría de la asignatura. Dicho método se basa en la idea de ecualizar el histograma; es decir, cambiar la distribución de los píxeles para cada posible valor, a una distribución más uniforme de los distintos valores de intensidad. El resultado se consiguió separando

los tres canales de la imagen mediante la función “split”, llamando con cada uno a la función “equalizeList”, para posteriormente volverlos a juntar. Para ver más claramente el resultado de ecualizar un histograma, se puede apreciar el resultado en la *Ilustración 2*.

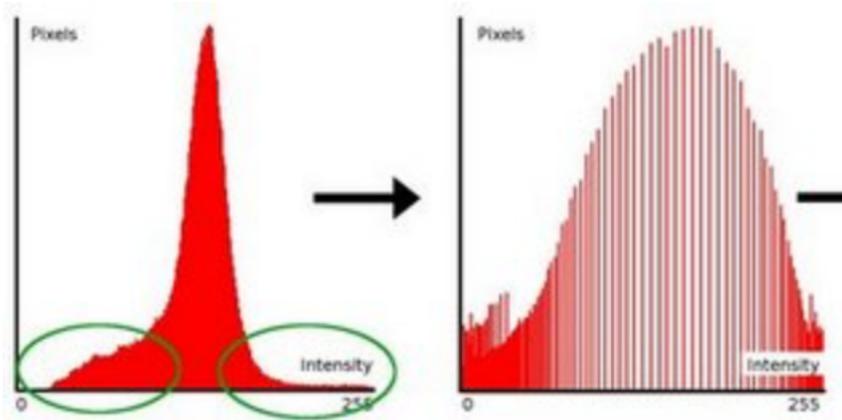


Ilustración 2.- Comparación entre histogramas

El histograma de la izquierda muestra la distribución del histograma de la imagen inicial, en la que se puede ver claramente como la mayor cantidad de píxeles se concentran en valores centrales de intensidad, mientras que en la imagen de la derecha se aprecia un histograma más uniformemente repartido con una menor diferencia de número de píxeles entre distintos valores de intensidades. Esto se refleja también muy claramente en el histograma acumulado comentado anteriormente ya que, para el histograma de la izquierda, la pendiente del acumulado sería elevada para las intensidades centrales, mientras que en el de la derecha tendría una pendiente más constante a lo largo de todas las intensidades.

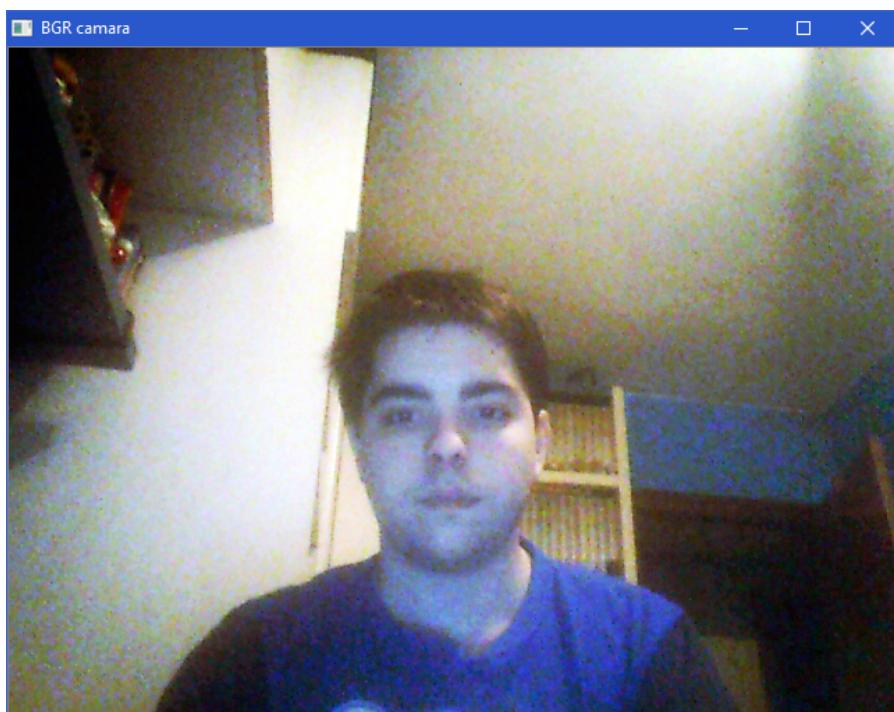


Ilustración 3.- Imagen con ecualización de histograma

Cambio de Color

Este efecto lo que pretende es modificar el color de la piel a distintos colores, dando un efecto de “alien” a la persona que haya sido captada por la cámara web del ordenador. El color de la piel puede ser cambiado entre rojo, azul y verde, simplemente indicándolo con la tecla reflejada en el menú de operaciones que aparece al iniciar el programa.

Para aplicar este efecto, ha sido necesario emplear el sistema de representación HSV, en lugar del BGR que utiliza “opencv” por defecto (función “cvtColor”). Se decidió utilizar este sistema de modelado de colores debido a que era más fácil encontrar el rango de colores en el que estaría encerrado el color de la piel, ya que tan sólo mirar el valor de “H” (ángulo) y dejando a un lado los valores de “S” y “V”, se puede aproximar con bastante exactitud los distintos rangos del color de la piel. Para entenderlo con más claridad se puede ver en la siguiente imagen una representación del modelo de “HSV”.

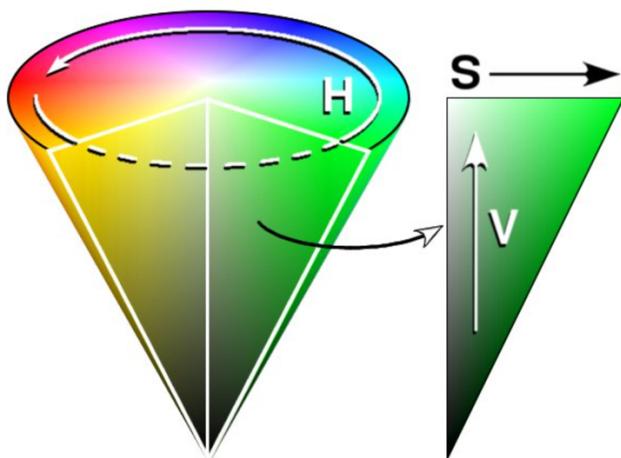


Ilustración 4.- Modelo de colores HSV

Como se puede ver, simplemente mirando los valores de “H”, que es el ángulo que representa los distintos colores, se puede aproximar el rango de la piel que estaría en torno a los valores [0,25] y [162,180]. Dichos valores se aproxiaron mediante una imagen tomada por la cámara y el programa “Photoshop”, que da unos valores aproximados para cada variable (“H”, “S” y “V”). La “S” representa la saturación en este modelo, y la “V” el valor o altura dentro del cono que varía entre blanco y negro.

Para conseguir el efecto final, tan sólo faltaba recorrer uno a uno los valores de cada píxel y comprobar si el ángulo “H” estaba comprendido en los valores citados, sustituyendo su valor por el ángulo correspondiente al rojo, azul o verde según fuera el caso. Los valores de “S” y “V” se dejaron igual para poder seguir apreciando las distintas escalas de colores de la imagen, en vez de pasar toda la piel a un mismo color. Cabe destacar que en “opencv” los valores de “H” oscilan entre 0 y 180, en vez de llegar a 360 como se representa generalmente, así que se tuvieron que adaptar.

Finalmente, una vez se tenía toda la imagen transformada para cambiar el color de piel, se volvió a pasar la imagen al modelo “BGR” para poder representarlo en la

pantalla final, observando cómo se conseguían buenos resultados, pero que al cambiar la iluminación los valores “H” de la piel cambiaban, por lo que se producían fallos en la detección de la piel.

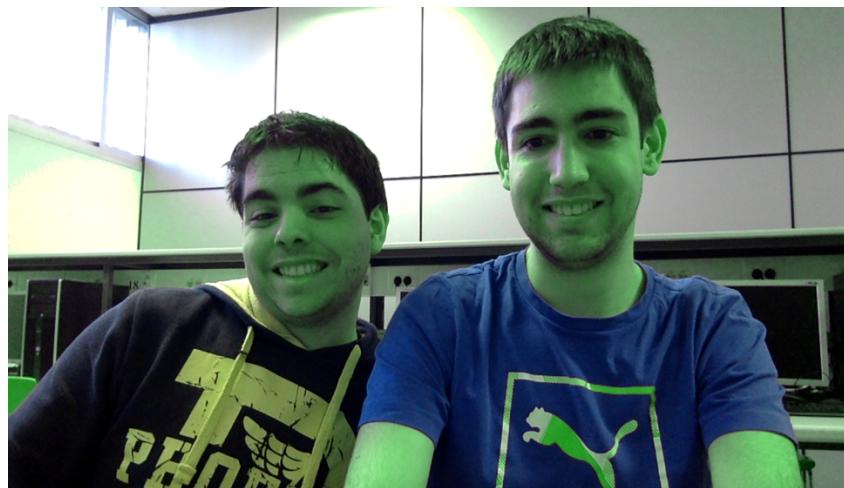


Ilustración 5.- Cambio de color de la piel a verde.



Ilustración 6.- Cambio de color de la piel a rojo.

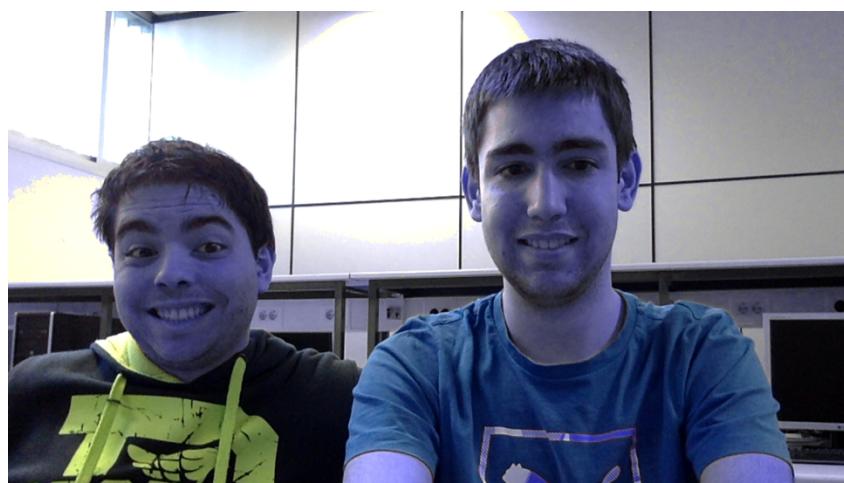


Ilustración 7.- Cambio de color de la piel a azul.

Reducción de Colores

El objetivo de este efecto es reducir el número de colores que contiene la imagen captada por la cámara del ordenador, logrando así que la imagen tenga un aspecto parecido al que tendría un poster. La reducción de colores se puede realizar de distintos modos, pero decidimos utilizar un método que ya usamos en la asignatura de Aprendizaje Automático.

Para aplicar este efecto, en primer lugar, se pregunta al propio usuario cuántos colores quiere que contenga la imagen (dentro del rango permitido). Una vez que es conocido este dato, se procederá a modificar la imagen captada para que solo tenga el número de colores introducidos por el usuario. En este caso, al contrario que en el efecto para cambiar el color de la piel, no fue necesario cambiar el modelo de representación del color.

Para modificar la imagen, se ha utilizado un algoritmo de “clustering” mediante el método de “k-medias” que ya se encontraba implementado en “opencv”. Este método etiquetará a cada color de la imagen para que pertenezca a un conjunto que tendrá a su vez un color (centroide) que representará a todo el conjunto. Como es previsible, habrá tantos conjuntos y “centroídes” como colores haya deseado el usuario para su efecto de póster. El algoritmo lo que hace es elegir unos “centroídes” aleatorios (un cierto píxel), y calcular la distancia de los demás con respecto a cada “centroide”, haciendo que pertenezcan al conjunto cuyo valor de la distancia sea menor.

Para poder emplear este método, ha sido necesario separar previamente la imagen en los distintos píxeles, ya que el método de “k-means” de “opencv” acepta los valores de entrada como una matriz columna en la que cada fila es una muestra. Una vez aplicado el algoritmo, sólo queda recorrer los píxeles, viendo con qué “centroide” han sido etiquetados, y sustituyendo su valor por el del correspondiente “centroide”.

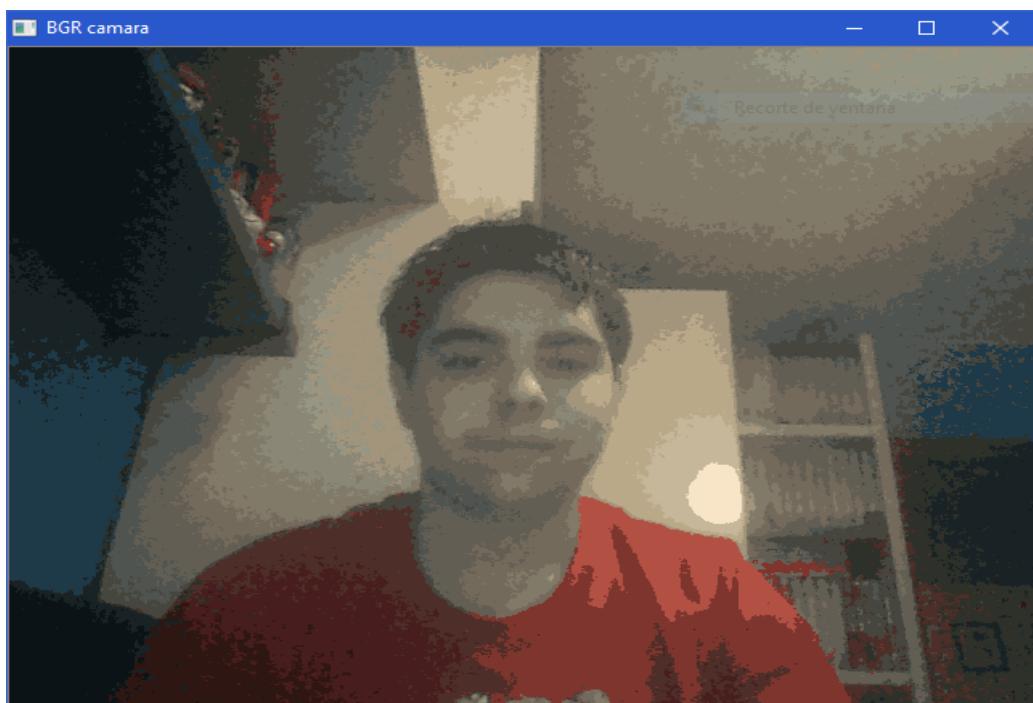


Ilustración 8.- Imagen reducida a 20 colores

Cojín y barril

Los efectos de cojín y barril son distorsiones de lente producidas dependiendo de la distancia focal que producen deformaciones de la imagen alejándolas de la realidad. En este efecto, se va a tratar de simular dichas dos distorsiones, mostrando los resultados en tiempo real dependiendo de un valor constante que determina el grado de deformación.

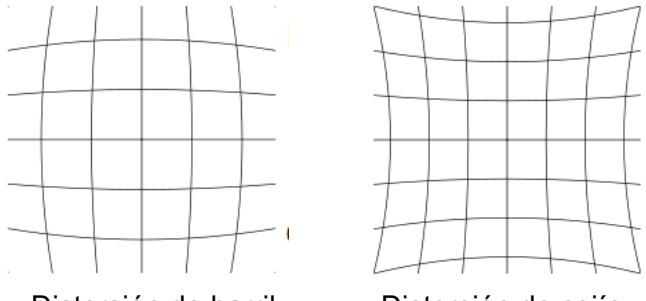


Ilustración 9.- Efecto de distorsión de barril y cojín.

Ambas distorsiones, como se puede ver en la imagen, dependen del radio desde el centro de la imagen, por lo que son radiales a lo largo de todos los píxeles. Para provocarla, se aplicó una fórmula encontrada en una fuente en internet. La fórmula es la siguiente:

$$coordX = \left(\frac{y - Cy}{1 + \left(\frac{\text{coeficiente}}{1000000} \right) * r^2} + Cy \right)$$
$$coordY = \left(\frac{x - Cx}{1 + \left(\frac{\text{coeficiente}}{1000000} \right) * r^2} + Cx \right)$$

Siendo cada variable:

- “y” y “x”: coordenada de cada píxel en el mapa para el “remap”.
- “Cx” y “Cy”: valores del radio de la imagen en cada dimensión.
- coeficiente: valor del coeficiente de distorsión.
- r2: distancia euclídea del punto al centro de la imagen.

Aplicando esta fórmula, y dado el valor del coeficiente de distorsión, se pueden conseguir los efectos deseados, diferenciando tan sólo ambos en que el coeficiente adquiere valor positivo para conseguir una distorsión de cojín, y un valor negativo para obtener el efecto de barril.

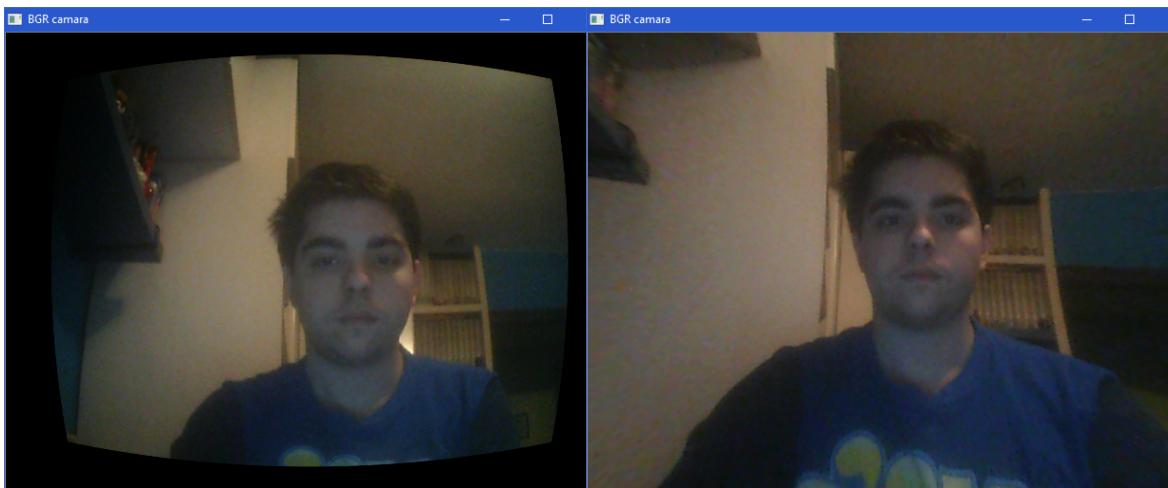


Ilustración 10.- Imágenes con distorsión de barril (izq.) y de cojín (der.).

Invertir imagen

De igual modo que para el efecto anterior, este se basa principalmente en la función “remap”. El efecto consiste en invertir la imagen y mostrar igual la coordenada “x”, pero invirtiendo de posición la “y”, consiguiendo así un efecto como si fuera un espejo común. Al obtener una imagen normal, lo que aparecía frente a la pantalla por la parte derecha, en la imagen mostrada final aparecía en la izquierda visto frente al ordenador. Con este efecto, se consigue que se reflejen ambos movimientos igual como pasaría al estar frente a un espejo.

Para conseguir el efecto, se hizo uso otra vez de la función “remap”, recorriendo las matrices para el cambio de coordenadas, y rellenándolas con la misma coordenada “x” pero cambiando la “y” por la nueva esperada. Así pues, se consigue como resultado en la matriz de salida el efecto deseado listo para ser mostrado por pantalla al pulsar el usuario la opción del menú que lo representa.

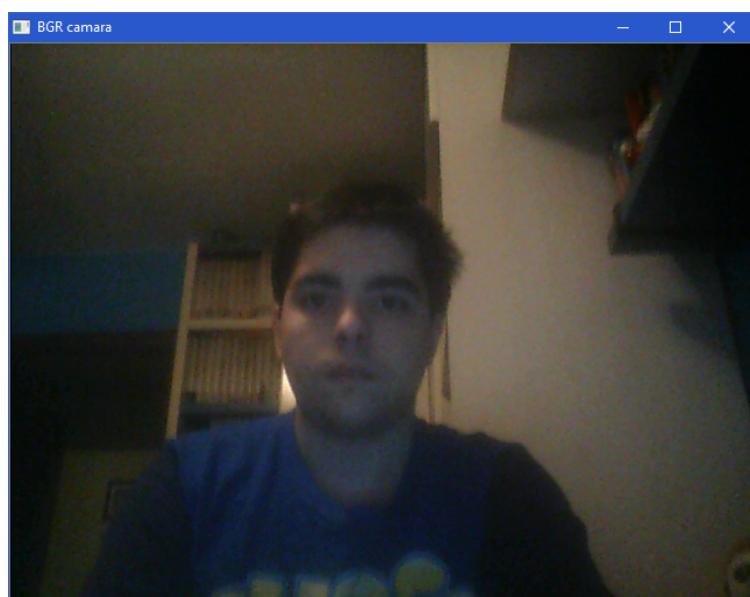


Ilustración 11.- Imagen invertida.

Rotar imagen 180°

Para realizar este efecto, se utilizó nuevamente la función “remap”. Simplemente, se llenaron las dos matrices que representan las nuevas coordenadas “x” e “y” que deben tomar cada píxel de la imagen original en la final, sabiendo que la coordenada “y” iba a ser la misma, y tan sólo variaba la “x”, ya que los píxeles de primeras filas pasaban a últimas y viceversa. Por lo tanto, se llamó nuevamente a la función “remap” con las matrices de entrada y salida y las de cambio de coordenadas, y se consiguió como resultado la imagen original rotada 180°.

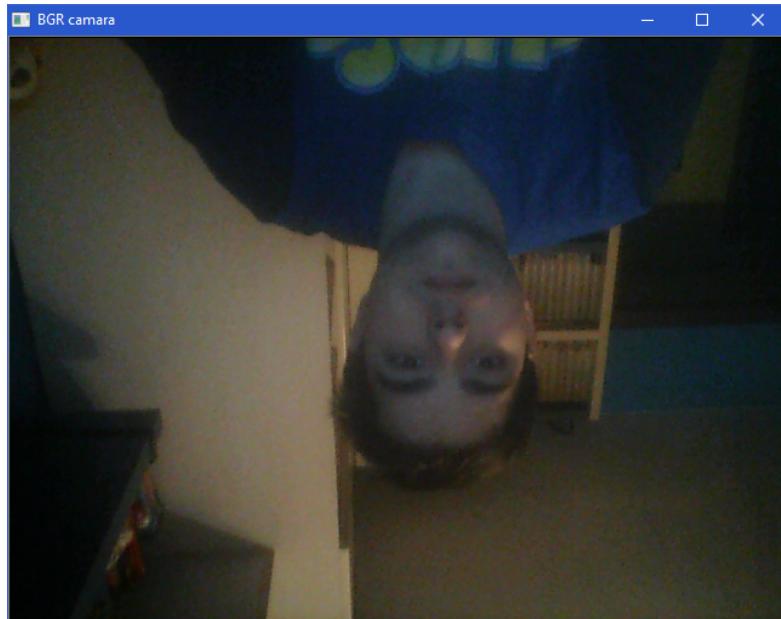


Ilustración 12.- Imagen rotada.

Escala de grises

Otro efecto implementado de manera opcional y uno de los más sencillos era cambiar la imagen original a escala de grises. Se miró como se podía implementar “a mano”, pero se vio que la función utilizada para otros efectos que permitía cambiar el modelo de representación del color también daba la posibilidad de cambiar la imagen a escala de grises. Así pues, se utilizó la función “cvtColor” con la matriz original, la matriz resultado y la indicación del cambio de representación de origen a destino (CV_BGR2GRAY).

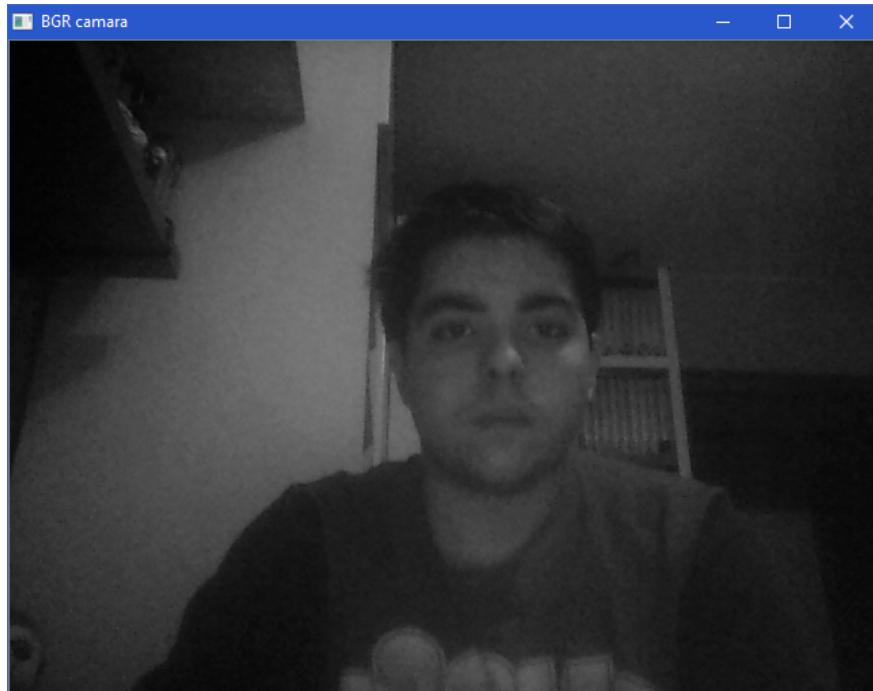


Ilustración 13.- Imagen en escala de grises.

Simétrica respecto a un eje

Otro de los efectos implementados como opcionales consiste en representar la imagen en una pantalla y su simétrica respecto a un eje. Explicado más detalladamente, lo que se hace es demandar al usuario un eje de simetría, ya sea el “X” o “Y”, y representar la imagen repetida, pero mostrando la original y su simétrica respecto a cualquiera de los ejes dados.

Para conseguir este efecto, se usó la función ya comentada “remap”, calculando las coordenadas que debía tener la imagen inicial en la final, e introduciendo sus valores en la correspondiente matriz. Cabe destacar que, si se quería representar la imagen tomada por la cámara dos veces (original y simétrica), había que redimensionarla (función “resize” para que cupiera en la final, obteniendo los valores de redimensión de los cálculos “numFilas/2” y “numColumnas/2” según fuera un eje u otro).

Así pues, se recorrieron todas las posiciones de la matriz introduciendo la coordenada en el eje “X” e “Y” que debían tomar en la imagen final, y llamando posteriormente a la función “remap” con las matrices con el cambio de coordenadas, la imagen inicial y matriz donde guardar la final, y un último parámetro que indica el tipo de interpolación para los píxeles no enteros (se usó el valor por defecto).

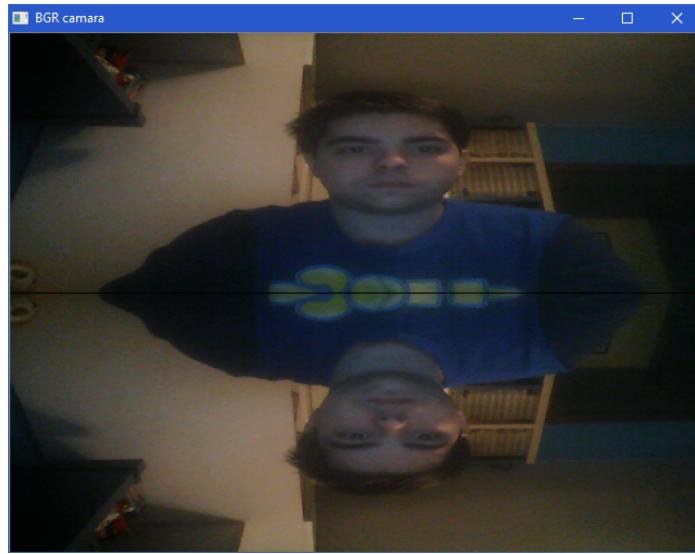


Ilustración 14.- Imagen simétrica en el eje x.

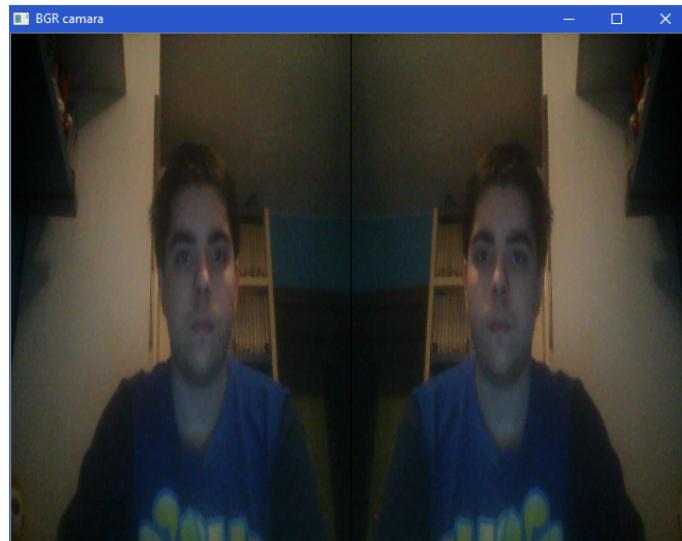


Ilustración 15.- Imagen simétrica en el eje y.

Imagen binaria

El efecto de pasar la imagen en color a una imagen binaria, como su propio nombre indica, consiste en conseguir una salida basada simplemente en dos colores: blanco y negro. La característica principal de este efecto es que divide los valores de cada píxel para asignarlos a totalmente blancos o totalmente negros. La operación requerida para conseguir este efecto se denomina “threshold” y está ya implementada en “opencv”.

Esta operación, igual que en la de mejora de contraste, se puede ver como una aplicación sobre el histograma. Los valores de cada píxel quedan reflejados ahí, y dado un valor límite o umbral, todos los píxeles con menos valor que el dado se caracterizan como negros, y los mayores como blanco. Por ello, este valor se pide al usuario para que lo introduzca por el teclado, consiguiendo de este modo que sea la persona la que decida sobre qué valor umbralizar para pasarla a binaria.

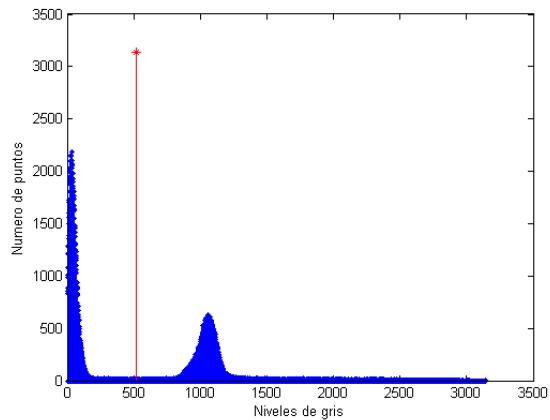


Ilustración 16.- Histograma con threshold o umbral

La función “threshold” de “opencv” se encarga de realizar todo el proceso, ya que sólo hace falta pasarle la imagen inicial, la matriz en la que guardar la imagen resultado, el valor umbral o “threshold” y el valor máximo sobre 255 que queremos utilizar (permite que en vez de blanco y negro se pueda cambiar el negro por algún otro valor en la escala de grises).

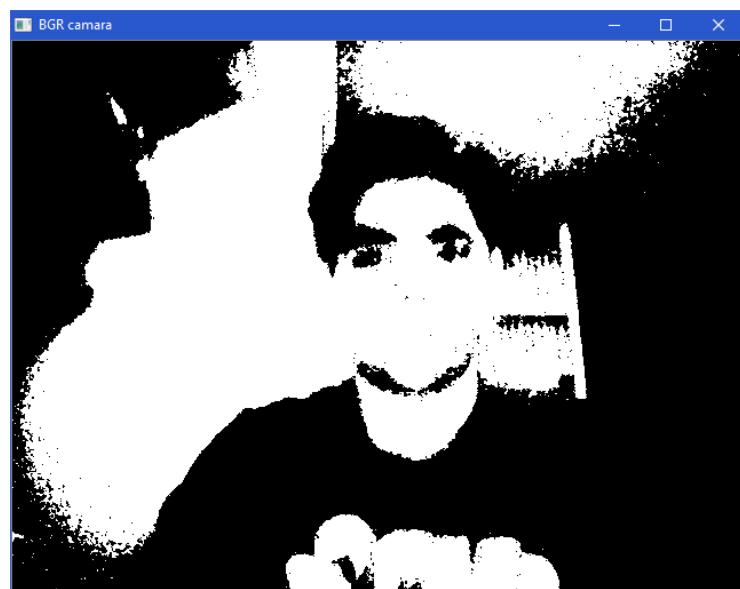


Ilustración 16.- Imagen binaria (threshold 50).

REFERENCIAS

- Moodle de “Visión por computador”:
 - <https://moodle2.unizar.es/add/course/view.php?id=17541>
- Tutorial “opencv”:
 - <http://docs.opencv.org/2.4/doc/tutorials/tutorials.html>
- HSV en Wikipedia:
 - https://es.wikipedia.org/wiki/Modelo_de_color_HSV
- Fórmula y código para distorsiones:
 - <https://github.com/machukas/openCV/blob/master/P1/Distorsion.cpp>