



# Universidad Zaragoza

## **VISIÓN POR COMPUTADOR**

### **PRÁCTICA 1**

[unizar.es](http://unizar.es)

Autores:

Sánchez Ballabriga, Víctor

Cepero Chicote, Juan Antonio

CURSO 2014 - 2015

Profesor:

Tardós, Domingo



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza

## Efecto “Alien”

Para implementar el tintado de la piel se ha desarrollado en primer lugar un detector de piel humana mediante el cambio de la imagen desde RGB a YcrCb y HSV, con estas nuevas imagenes se emplea un filtro para guardar una imagen binaria donde los pixeles blancos son aquellos que entran en el rango de color que se corresponde con el de la piel. Una vez obtenida esta imagen binaria, se usan las coordenadas de estos pixeles blancos para saber las coordenadas donde hay que aplicar el tintado en la imagen original RGB.

Se han implementado los tres colores pedidos en el guión de la práctica y además los colores primarios de impresora, los usados para la coloración por sustracción, que son amarillo, cián y magenta.

Se puede cambiar de color aplicado a la piel en tiempo de ejecución pulsando la tecla correspondiente a la inicial del color (G, R, B, M, Y, C).

## Efecto Poster

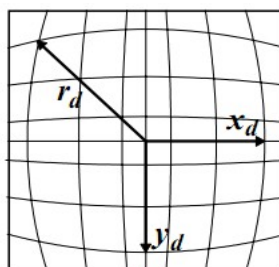
Para implementar este efecto se intentó hacer en un primer momento sobre el formato HSV de la imagen para evitar “quemazos” al simplificar la gama de colores debido a brillos, pero se acabo consiguiendo un efecto más parecido al de un poster trabajando directamente sobre la imagen RGB. Para hacerlo la función tiene un parámetro de entrada relativo al número de saltos de color en cada canal, con él se calcula el color que representa cada salto, y así se puede calcular al salto al que pertenecerá cada canal de color de cada pixel de la imagen con una simple división.

## Distorsión de Barril y de Cojín

Para implementar ambas distorsiones en tiempo real se ha usado la siguiente formula pixel por pixel de la imagen para calcular la nueva posición de dicho pixel:

$$r_u = r_d(1 + kr_d^2)$$

Siendo  $r_d$  la distancia al pixel central de la imagen en una determinada dimensión ( $x$  |  $y$ ), y  $r_u$  la nueva distancia a dicho pixel,  $k$  es una constante, con la cual se consigue el efecto de distorsión de barril cuando es positiva y de cojín cuando es negativa.



Se puede ajustar el nivel de distorsión de barril con las teclas “m” y “l”.

## **Contraste**

Se ha implementado el ajuste de contraste variable en tiempo real. Se emplean las teclas “m” y “l” para aumentar y reducir el nivel de contraste respectivamente. Para cada frame capturado por la cámara, se multiplica cada pixel por un valor alpha que va variando en función de las veces que se pulse cada tecla. Se controla además que el valor no exceda el máximo con la función `satúrate_cast(...)` de openCV. A continuación, aparecen algunos ejemplos de los resultados obtenidos al aplicar este efecto:

## **Ecualización del Histograma**

Se ha implementado tanto la ecualización del histograma de cada frame en color, como de cada frame en blanco y negro. Para el caso de blanco y negro, se convierte cada frame capturado por la cámara a un esquema de color en blanco y negro empleando la función `cvtColor(..., CV_BGR2GRAY)`, para a continuación ecualizar el histograma con la función `equalizeHist()` de openCV. Para el caso de frames en color, en primer lugar, se pasa cada frame a un esquema de color `CV_BGR2YCrCb`, a continuación, se separan los canales y se ecualiza únicamente el primer canal, ya que es la componente que encierra la luminosidad. Una vez hecho esto se vuelven a juntar los canales y se muestra la imagen de salida. A continuación, aparecen algunos ejemplos de los resultados obtenidos al aplicar este efecto:

## **Ruido de Sal y Pimienta**

Se ha implementado como apartado opcional. Para cada frame capturado por la cámara, se genera este tipo de ruido en base a un parámetro `k` que se puede modificar en tiempo real (con las teclas “m” y “l”). Para cada frame se seleccionan aleatoriamente un número de pixels en base al parámetro `k`. Estos pixels se colorean de blanco o de negro al 50% y se insertan en la imagen de salida. A continuación, aparecen algunos ejemplos de los resultados obtenidos al aplicar este efecto: