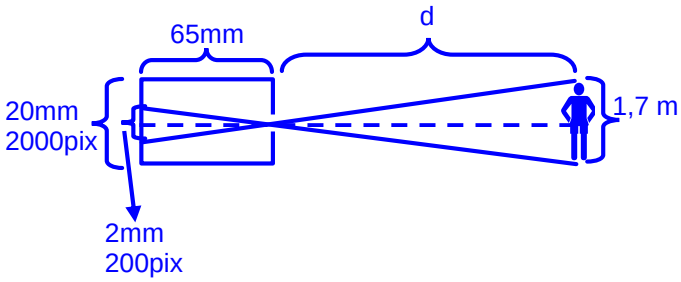
Examen de Procesamiento Audiovisual, 11/6/2010

Nombre: Ginés García Mateos

Titulación (II/ITIS): TIPO **X**

- En los ejercicios que sean de *programar*, suponer que tenemos una librería de procesamiento de imágenes con operaciones al estilo de las OpenCV. Utilizar siempre un pseudocódigo de alto nivel.
- Limitarse al espacio disponible para cada pregunta. Si crees que necesitas más espacio, es que probablemente no lo estás planteando bien.

1. (1,2 puntos) Tenemos una cámara de fotos digital con un CCD de 3000x2000 píxeles. El CCD mide 30x20 milímetros. Hemos tomado una fotografía con un objetivo de 65 mm de distancia focal. En el centro de la foto sale nuestro amigo Pepito, que mide 1,70 metros, con una resolución de 200 píxeles de alto. ¿A qué distancia se encontraba Pepito cuando tomamos la foto? Representa gráficamente la situación del problema y calcula la distancia que se pide.



Por simple analogía de triángulos, tenemos que:
2 mm / 65 mm = 1,7 m / d m

Por lo tanto, la distancia es:
d = 1,7 * 65 / 2 = 55,25 metros

2. (1 punto) Necesitamos detectar cierto color en una escena, por ejemplo, los tonos anaranjados. Queremos modelar un tono de color independientemente de la luminosidad (es decir, admitir tonos claros y oscuros de ese color). Para ello, decidimos usar un **histograma de dos canales** en algún espacio de color adecuado. Tenemos las siguientes posibilidades, para cada una de ellas indicar si es adecuada o no para este problema concreto; si es adecuada, indicar de qué dos canales se debería calcular el histograma.

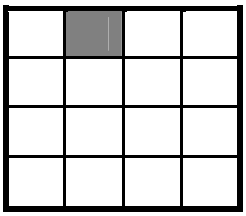
Modelo de color	¿Es adecuado?	Calcular el histograma de los canales…
RGB	No	
YCrCb	Sí	Canales Cr y Cb (se descarta Y).
HLS	Sí	Canales H y S (se descarta L).
CMYK	No	
YUV	Sí	Canales U y V (se descarta (Y).

3. (1,2 puntos) En los siguientes apartados, puede haber una o varias opciones correctas (siempre hay por lo menos una). Indica todas las que sean ciertas.

Opciones correctas	Afirmaciones
a b c	Una imagen contiene una campana gausiana. Calculamos el espectro de esa imagen (la magnitud de su transformada de Fourier). Podemos afirmar que el espectro de la imagen no cambiará si aplicamos las operaciones: a) Invertir los valores de brillo de la imagen. b) Rotar la imagen 65º a la derecha. c) Desplazar la imagen unos pocos píxeles a la izquierda. d) Aplicando un suavizado gausiano a la imagen.

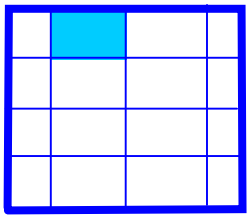
b d e	Las operaciones de convolución incluyen: a) La ecualización local del histograma. b) El suavizado gausiano y de media. c) Las transformaciones bilineales. d) Las derivadas en X y en Y. e) Los filtros de perfilado. f) Las operaciones de morfología matemática.
c	Tenemos una imagen que está manchada con una especie de pelos pequeños (líneas oscuras y muy finas, de un píxel de ancho). Lo más adecuado para eliminar los pelos de la imagen es: a) Aplicar un filtro de suavizado de media, con una máscara alrededor de 3x3. b) Aplicar un suavizado gausiano unidireccional. c) Aplicar un suavizado de mediana, por ejemplo, una mediana de 3x3. a) Aplicar un perfilado, es decir la Laplaciana más la imagen original.
c	Cierta máscara de convolución de tamaño 3x3 tiene: la primera fila todo 1, la segunda fila todo 0, y la tercera todo -1. a) Se puede considerar como un filtro de suavizado. b) Se puede considerar como un filtro derivativo, en concreto, derivada en X. c) Se puede considerar como un filtro derivativo, en concreto, derivada en Y. d) Se puede considerar como un filtro Laplaciano de segunda derivada.

4. (1,2 puntos) Suponer la imagen que tenemos debajo.

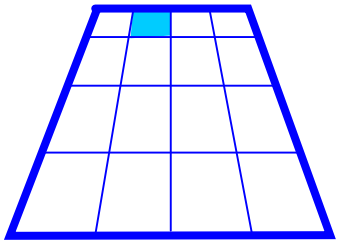


Sobre dicha imagen queremos aplicar las siguientes transformaciones geométricas. Representa, de manera aproximada, el resultado que se obtendría con cada una de ellas.

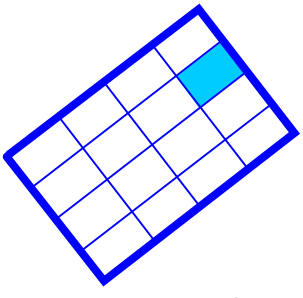
a) Transformación cilíndrica en sentido horizontal



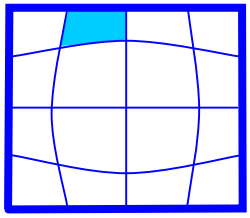
b) Transformación perspectiva (una arbitraria y que no sea una afín)



c) Transformación afín cualquiera



d) Transformación de estirar



5. (1,2 puntos) En relación a los miniproyectos presentados en clase, las siguientes afirmaciones contienen una parte que es cierta y otra parte que es falsa. Indicar la parte que es falsa y justificar brevemente por qué.

Afirmación	Justificación
En procesamiento de sonido digital, el efecto *flanger* es parecido a un eco, pero con retardos muy grandes de más de 2 segundos y poca atenuación de la onda copiada	Es cierto que el efecto *flanger* es parecido a un eco con poca atenuación, pero los retardos son muy pequeños, de duraciones próximas a los 10 ms.
En la reconstrucción 3D con mapas de alturas se usa un haz de luz blanco, y lo ideal es que la cámara y la fuente de luz estén muy cerca uno del otro	La fuente de luz y la cámara deben estar alejadas cierta distancia, ya que de lo contrario las líneas se verían casi rectas y no se podría realizar una estimación precisa de la profundidad.
Los parámetros más importantes que definen un sonido digital son la frecuencia de muestreo, el bit rate, el efecto aliasing y el número de canales	El aliasing no es un parámetro del sonido digital, sino que es un efecto indeseable que se produce con frecuencias de muestreo inadecuadas.
En la reconstrucción 3D, la distancia de la línea detectada a la línea base indica la profundidad del punto; cuanto más brillante es la línea más cercano está el punto	El brillo de los píxeles de la línea no está relacionado (en principio) con la profundidad del punto. La profundidad está en función de la distancia a la línea base.

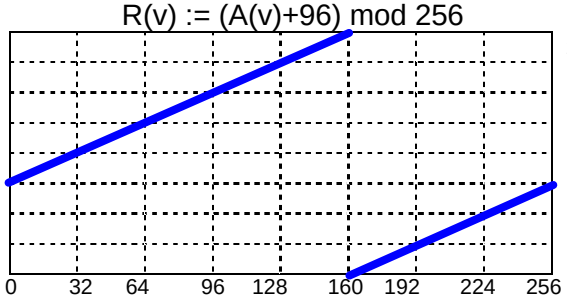
6. (1,5 puntos) Dada una imagen **img**, queremos cambiar todos los píxeles rojos por azules. En concreto, todos los píxeles que tengan un valor RGB entre (200, 0, 0) y (255, 0, 0) inclusive, deben ser sustituidos por el valor RGB (0, 0, 255).

   La anterior operación se podría hacer fácilmente recorriendo todos los píxeles de la imagen, y comprobando para cada uno si se cumple la condición o no. No obstante, eso sería demasiado lento, por lo que queremos implementarlo usando únicamente **operaciones globales** sobre las imágenes. Escribir una secuencia de operaciones para obtener el resultado deseado (cambiar los píxeles rojos por azules).

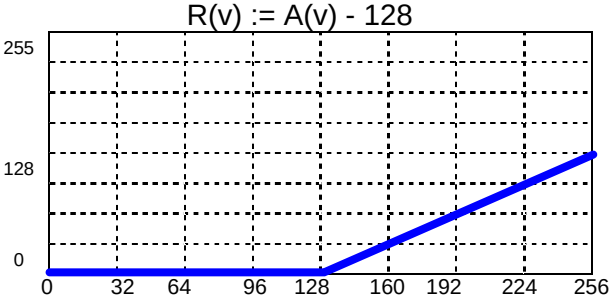
```
// Se puede hacer de muchas maneras diferentes. La forma más sencilla,
// usando operaciones que tienen su equivalente en OpenCV, puede hacerse
// con sólo dos instrucciones (suponemos que la imagen de entrada es A):

Mascara= EnRango(A, RGB(200,0,0), RGB(255,0,0))  // Función cvInRange
Set(A, RGB(0,0,255), Mascara)          // Función cvSet usando una máscara
```

7. (1,2 puntos) Representa gráficamente la forma que tienen las curvas tonales correspondientes a las siguientes transformaciones globales.

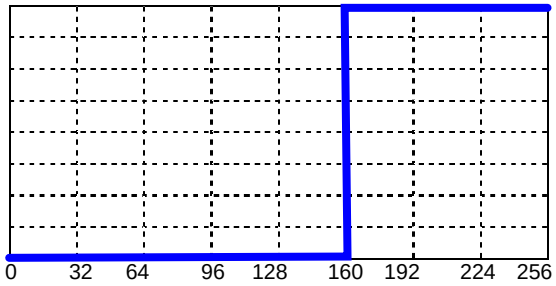
a) Sumar una constante (96) módulo 256
R(v) := (A(v)+96) mod 256



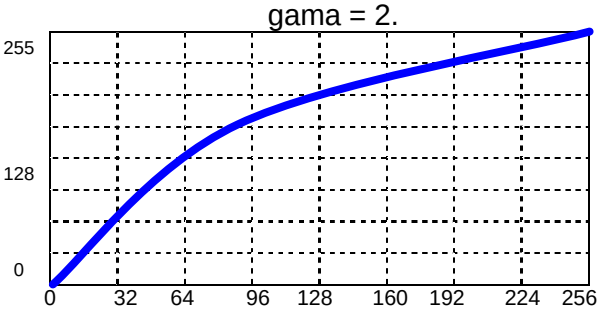
b) Restar una constante (128)
R(v) := A(v) - 128



c) Umbralizar (binarizar) una imagen con el valor de umbral 160



d) Transformación de gama, usando gama = 2.



8. (1,5 puntos) Tenemos una secuencia de vídeo de cierta escena grabada a 30 fps y con la cámara fija. De manera inesperada y aleatoria, la imagen aparece invertida (*flipada*) en algunos frames. El problema se puede resolver aplicando cvFlip a las imágenes invertidas. Pero el inconveniente es que no sabemos a priori qué imágenes están invertidas y cuáles no.

Frame 0 Frame 1 Frame 2 Frame 3 Frame 4…

Escribe un programa que dada una secuencia de *N* frames como la anterior, devuelva la secuencia rectificada (es decir, aplicando cvFlip a los frames que correspondan). Se supondrá que el frame 0 siempre está bien.

```
// Suponemos que la secuencia de vídeo es: frame[0], frame[1], …, frame[N-1]
// El resultado se guarda en las mismas posiciones.

para i= 1 hasta N-1 hacer
      double dif1= Media(DiferenciaAbs(frame[i], frame[i-1]))
      double dif2= Media(DiferenciaAbs(Flipar(frame[i]), frame[i-1]))
      si dif2 < dif1 entonces frame[i]= Flipar(frame[i])
finpara

// La idea es sencilla: entre la imagen original y la flipada, nos quedamos con la que
// más se parezca a la anterior. Media = cvAvg. DiferenciaAbs = cvAbsDiff
```