

Herramientas Software para Tratamiento de Imágenes
Javier Sáez Alonso

=====

INSTRUCCIONES EJERCICIOS OCTAVE/MATLAB

=====

Ejercicio 1 - Blur detection

Solo funciona con imágenes jpg
Ejercicio con **Octave 4.0.2** bajo Linux.

Emborronamiento de imágenes: `blurrer('./directorio/')`
Detección de blur: `blur detection('./directorio/')`

Ejercicio 2 - Color tracking

Solo funciona con imagenes jpg con nombre frameX.jpg (frame1.jpg, ...frame211.jpg)
Ejercicio con **Octave 4.0.2** bajo Linux.

Generación de filtro: `range color selector('./directorio/')`

Seguir las instrucciones en consola. Cuando dice "Pulse una tecla.." debe hacerse estando la ventana de la imagen seleccionada.

```
-----  
h-HUE, s-SATURATION, v-VALUE  
o/p - decrementar/incrementar limite superior  
k/l - decrementar/incrementar limite inferior  
n - siguiente imagen. q - salir  
-----  
Hmax: 1.000000 - Hmin 0.000000  
Smax: 1.000000 - Smin 0.000000  
Vmax: 1.000000 - Vmin 0.000000  
-----  
Pulse una tecla sobre la imagen para continuar
```

Salida generada: `[hmin,hmax,smin,smax,vmin,vmax]`

Seguimiento por color:
`color_tracking([0.320,0.400,0.200,1.000,0.180,1.000], './frames/')`

Filtro video rotulador verde:
- `[0.320,0.450,0.200,1.000,0.170,0.650];`

Frames alumno:
<https://dl.dropboxusercontent.com/u/64814516/master/HSW/frames-isa.zip>

Filtros video alumno:
- amarillo `[0.050,0.200,0.400,1.000,0.40,1.000]`
- verde `[0.320,0.400,0.200,1.000,0.180,1.000]`
- azul `[0.500,0.900,0.200,1.000,0.300,1.000]`

Ejercicio 3 - barCodeDetector

Ejercicio realizado con Matlab R2015A bajo Linux.

Detección de código: `barCodeDetector('./barcodeimages/barcode2.jpg')`