

# Práctica 10

Tercera de seguimiento de vídeo

SIFT – emparejamiento - homografía

# Índice

- ROI + máscara
- Puntos característicos y sus descriptores
- Emparejamiento de descriptores entre *frames*
- Cálculo homografía
- Cálculo del baricentro de la máscara y fusión del banner con transparencia
- Enunciado de entregables finales

# *Region Of Interest - máscara*

```
%% Leer un archivo de vídeo, modificar, escribir en otro
videoHandleIn = VideoReader('ejemplo.mp4'); % de donde leeremos
videoHandleOut = VideoWriter('salida.avi'); % a donde escribiremos
videoHandleOut.Quality=95;
videoHandleOut.FrameRate = videoHandleIn.FrameRate; % heredamos el frame rate

open(videoHandleOut); % creamos realmente el fichero de salida

%%
frameBefore = readFrame(videoHandleIn);
[BW, xind, yind] = roipoly(frameBefore); % se pide al usuario que dibuje un polígono alrededor de la zona de interés
R = round(sqrt((max(xind)-min(xind))/2 + (max(yind)-min(yind))/2)); % calculamos automáticamente una gaussiana adecuada
D = 2*R+1; % calculamos el diámetro como doble del radio y nos aseguramos de que sea impar
g = fspecial('gaussian', D, R/3);
BWfiltered = imfilter(double(BW), g); % la ventana suavizada en los bordes aún en blanco y negro
frameBeforeROI = im2uint8(im2double(frameBefore).*BWfiltered); % Ahora aplicada como máscara. Constituye el frame "antes de"

roi = images.roi.Polygon; % creamos el objeto ROI que luego necesitaremos para actualizar máscara y posición de la ROI
roi.Position = [xind, yind];
```

# Harris + SIFT, no el propio: con funciones Matlab

```
%%
```

```
pointsBefore = %% Llamar a Harris limitándolo a que analice la imagen en la ROI, esto es, frameBeforeROI  
[featuresBefore,validPointsBefore] = % extraer las características
```

# Emparejamiento de características entre frames

%%

```
while hasFrame(videoHandleIn) % mientras hayan frames que procesar
    frameNext = readFrame(videoHandleIn); % sacamos el frame actual como imagen
    frameNextROI = im2uint8(im2double(frameNext).*BWfiltered);
    pointsNext = % igual que linea "pointsBefore=" pero para el nuevo frame
    [featuresNext,validPointsNext] = % similarmente, pero para el nuevo frame

    indexPairs = % llamar a emparejamiento de características
    matchedPointsB = % qué puntos de valid points before fueron emparejados
    matchedPointsN = % qué puntos de valid points next fueron emparejados
    if matchedPointsN.Count < ¿cuanto? % si no tenemos suficientes para montar el modelo
        abortamos
            break;
    end

    ... % continua siguiente transparencia
```

# Cálculo homografía

```
showMatchedFeatures(frameBeforeROI,frameNextROI, matchedPointsB, matchedPointsN);
imageCapture = getframe(gcf);
legend('frame anterior','frame siguiente'); pause(1/videoHandleIn.FrameRate);

tform = % help estimateGeometricTransform
[xind,yind] = % help transformPoints ¿forward o inverse?

roi.Position = [xind, yind]; % actualizamos valores para...
BW = roi.createMask(rgb2gray(frameActual)); % comenzar un nuevo ciclo...
BWfiltered = imfilter(double(BW), g); % donde el frame actual pasar a ...
frameBeforeROI = im2uint8(im2double(frameNext).*BWfiltered); % ser el frame anterior
pointsBefore = pointsNext;
featuresBefore = featuresNext;
validPointsBefore = validPointsNext;

writeVideo(videoHandleOut,imageCapture.cdata); % lo escribimos
end

close(videoHandleOut); % cerramos el vídeo que estamos creando
```

# Cálculo del baricentro de la máscara y fusión del banner con transparencia

- Pueden encontrar las explicaciones al respecto en las transparencias vistas ayer en clase de teoría.
- Quitar la visualización de “traza” (`showMatchedFeatures`, `getframe` y `legend`) y superponer en cada frame el texto con antialiasing moviéndose a una distancia fija del centro de la ROI
- Para recuperar el canal de transparencia del banner tienen que usar `imread` con dos parámetros de salida

# Enunciado entregables finales

## 1) Suban un informe en PDF donde comenten:

- la práctica de *template matching* con las fotos antiguas de Prokudin-Gorski
- la práctica en la que mostrábamos los puntos de Harris en el vídeo de la lanzadera espacial (solo comenten el código e incluyan las capturas oportunas, pero NO suban ese vídeo).
- comentarios y fragmentos de código que estimen oportunos acerca de esta práctica final de seguimiento de características en vídeo. Especialmente si realizan partes opcionales.

## 2) Suban también el **video** resultante, recuerden:

- deben usar como punto de partida **un video diferente** al de ejemplo que les he suministrado.
- se admite que la imagen o animación que acompañe a una región de su vídeo no se deformé, solo se traslade. Es **OPCIONAL** que apliquen una transformación proyectiva a su imagen. Pero **OBLIGATORIAMENTE** su imagen o animación de acompañamiento **debe tener transparencia**.
- se admite que sea una imagen, o un texto, estático. **OPCIONALMENTE** pueden insertar una animación, pero **NO** hemos tratado en clase cómo conseguir una: es una parte creativa... solo para quien tenga tiempo, y conocimientos-habilidades previos.
- basta con unos pocos segundos, entre 5 y 15. Asegúrense que no se supera un tamaño de 50 Megabytes (entre informe y vídeo). Usen un *encoder* habitual, como h.264 o h.265 en contenedor mp4, avi, mpeg o webm. Si su fichero excede el límite de tamaño, prueben a usar cualquier herramienta de recodificación de vídeo, para rebajar el tamaño ya sea sustrayendo calidad o resolución. Puede usar herramientas similares, para rebajar también el tamaño del informe, como la página [https://www.ilovepdf.com/compress\\_pdf](https://www.ilovepdf.com/compress_pdf)