

Práctica 2

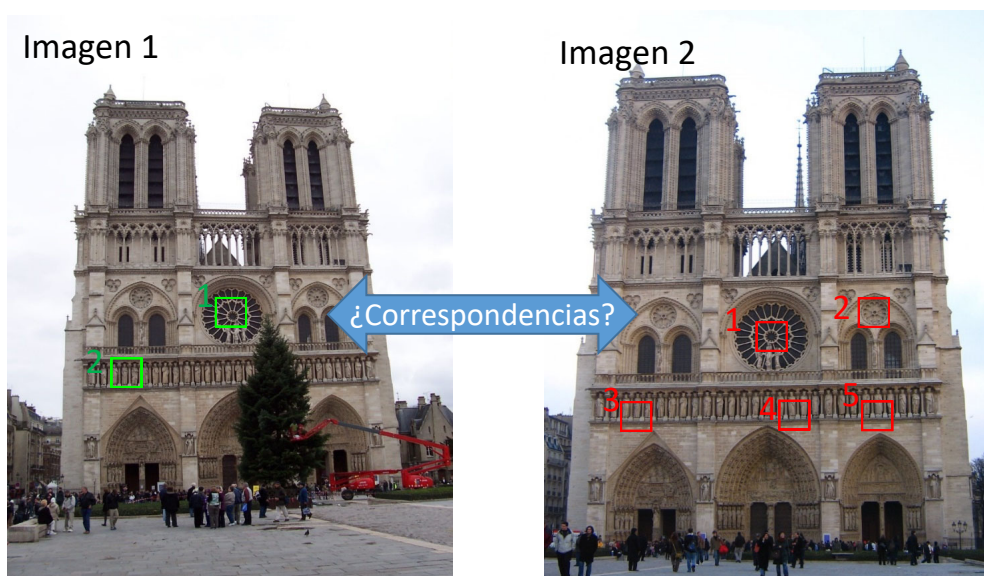
Extracción, descripción y similitud de características locales



OBJETIVO



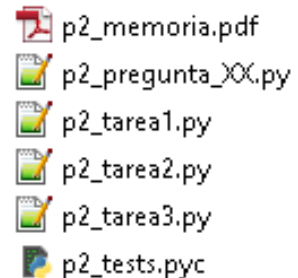
- Extracción, descripción y correspondencia de puntos de interés



Descriptores de los
puntos de interés

$$\begin{matrix} D_1^1 = [x_1^{(1)}, \dots, x_d^{(1)}] \\ D_2^1 = [x_1^{(2)}, \dots, x_d^{(2)}] \end{matrix} \begin{matrix} \text{¿?} \end{matrix} \begin{matrix} D_1^2 = [x_1^{(1)}, \dots, x_d^{(1)}] \\ D_2^2 = [x_1^{(2)}, \dots, x_d^{(2)}] \end{matrix} \begin{matrix} D_3^2 = [x_1^{(3)}, \dots, x_d^{(3)}] \\ D_4^2 = [x_1^{(4)}, \dots, x_d^{(4)}] \end{matrix} \begin{matrix} D_5^2 = [x_1^{(5)}, \dots, x_d^{(5)}] \end{matrix}$$

- Descargue los ficheros de código Python disponibles en Moodle
 - Fichero `p2_caracteristicas_code.zip`
- Complete las tareas utilizando los ficheros *.py contenidos en el ZIP
- La entrega se **realiza por parejas** y constará de:
 - Memoria en PDF (1 fichero)
 - + código Python adicional
 - Ficheros código Python de las tareas realizadas (hasta 3 ficheros)
 - Fichero `p2_tests.pyc`



No es necesario entregar todas las tareas

- Tarea 0 – Descargar material e incluir nombre en ficheros
- **Programación:**
 - **Tarea 1** - Implementar la **detección** de puntos de interés
 - **Tarea 2** - Implementar la **descripción** de puntos de interés
 - Tarea 2a - Histogramas de nivel de gris
 - Tarea 2b - Histogramas de orientación de gradientes
 - **Tarea 3** - Implementar la **similitud** de puntos de interés para determinar correspondencias mediante distancia Euclídea
 - Tarea 3a - Distancia mínima umbralizada
- **Razonamiento:**
 - **Tarea 4** - Responder a las preguntas en memoria

Aumento de complejidad con respecto a la práctica anterior.
Además no existe listado de operaciones a implementar
(se deben consultar los apuntes de teoría y artículos)

- Implementar detector de puntos de interés de Harris

- Función `detectar_puntos_interes_harris(imagen, sigma = 1, k = 0.05, threshold_rel = 0.2)` en el fichero `p2_tarea1.py`

- Consideraciones:

- Antes de operar convierta la imagen al formato `float` en el rango `[0,1]`
 - Considere una ventana $w(u,v)$ de tamaño `1x1` y aplique el procedimiento resumido visto en clase (utilice los filtros de *sobel* para el cálculo de derivadas en lugar de los descritos en el procedimiento)
 - Funciones recomendadas:
 - `scipy.signal.convolve2d` con `mode='same'`
 - `scipy.ndimage.filters.gaussian_filter` con `mode='constant'`
 - `skimage.feature.corner_peaks` con `min_distance = 5`
 - `skimage.feature.peak_local_max()` con `min_distance = 5`

- Funciones prohibidas:

- Cualquier función que no sea del paquete `numpy` o `scipy`
 - No se pueden utilizar las funciones `img_as_float64`, `img_as_float32`, `img_as_float` para normalización de imágenes. Se debe realizar la operación manualmente.
 - Por ejemplo: `skimage.filters.*`, `skimage.feature.corner_harris`

- Documentación

- <https://scikit-image.org/docs/dev/api/skimage.feature.html>
 - <https://docs.scipy.org/doc/scipy/reference/signal.html>
 - <https://docs.scipy.org/doc/scipy/reference/ndimage.html>

- Implementar la descripción de puntos de interés con Histogramas de niveles de gris alrededor del punto

- Fichero `p2_tarea2.py`

- Función `descripcion_puntos_interes(imagen, coords_esquinas, vtam = 8, nbins = 16, tipoDesc='hist')`:

- Consideraciones:

- Identificar el descriptor con `tipoDesc='hist'`
 - Antes de operar convierta la imagen al formato `float` en el rango `[0,1]`
 - El descriptor considera un vecindario `vtam+1` x `vtam+1` centrado en la coordenada de la esquina para la cual se extrae el descriptor
 - Cálculo histograma
 - Debe estar normalizado (i.e. que la suma de todos sus elementos sea 1)
 - Considere `nbins` niveles/intervalos de cuantificación homogéneos
 - En cada nivel de cuantificación, los intervalos serán del tipo `[a,b)`

- Funciones

- **Recomendadas:** `numpy.histogram` y `numpy.flatten` con valores por defecto
 - **Prohibidas:** Cualquiera que no sea del paquete `numpy` o `scipy` (e.g `skimage.feature.*`)
No se pueden utilizar las funciones `img_as_float64`, `img_as_float32`, `img_as_float` para normalización de imágenes. Se debe realizar la operación manualmente.

- **Sugerencia:** cree dos listas con los descriptores/esquinas que vaya calculando y posteriormente transforme las listas en arrays de `numpy` con `np.asarray`

- Implementar la descripción de puntos de interés con Histogramas de Gradientes Orientados alrededor del punto

– Fichero `p2_tarea2.py`

– Función `descripcion_puntos_interes`

(`imagen`, `coords_esquinas`, `vtam` = 8, `nbins` = 16, `tipoDesc`='hist'):

– Consideraciones:

- Identificar el descriptor con `tipoDesc='mag-ori'`
- Antes de operar convierta la imagen al formato *float* en el rango [0,1]
- El descriptor considera un vecindario `vtam+1` x `vtam+1` centrado en la coordenada de la esquina para la cual se extrae el descriptor
- Pasos para obtención del histograma:
 - Calcule el gradiente como visto en clase
 - Considere `nbins` niveles de cuantificación homogéneos en el rango [0, 360)
 - En cada nivel de cuantificación, los intervalos serán del tipo [a,b)
 - Ejemplo: 16 nbins → niveles = [0. 22.5 45. 67.5 90. 112.5 135. 157.5 180. 202.5 225. 247.5 270. 292.5 315. 337.5 360.]
(ojo que hacen falta 17 valores para definir 16 bins)

– Funciones

- Recomendadas: `scipy.ndimage.sobel`, `numpy.rad2deg`, `numpy.arctan2`, `numpy.linspace`, `numpy.digitize` con valores por defecto
- Prohibidas: Cualquiera que no sea de `numpy` ó `scipy` (e.g `skimage.feature.*`) . No se pueden utilizar las funciones `img_as_float64`, `img_as_float32`, `img_as_float` para normalización imágenes.

- Implementar la correspondencia de puntos de interés con distancia mínima umbralizada alrededor del punto

– Fichero `p2_tarea3.py`

– Función `correspondencias_puntos_interes` (`descriptores_imagen1`, `descriptores_imagen2`, `tipoCorr`='mindist', `max_distancia`=25)

– Consideraciones:

- Las correspondencias se determinan para cada descriptor de la imagen 1 (e.g. es decir, se busca el mejor descriptor en la imagen 2)
- La distancia utilizada siempre será euclídea
- Identificar el método con `tipoDesc='mindist'`
- Funciones recomendadas:
 - `numpy.linalg.norm` con valores por defecto
 - Bucles con `enumerate`

– Funciones prohibidas:

- Cualquier función que no sea del paquete de `numpy` y `scipy`

- Memoria

- Responda a las preguntas del fichero “preguntas_P2.docx”

- No necesita realizar funciones nuevas, el trabajo se basa en realizar experimentos con la funcionalidad que ha generado
 - Puede necesitar repasar algunos conceptos teóricos
 - Utilice figuras y tablas para resumir experimentos

- Ficheros a generar

- Fichero “preguntas_P2.pdf” en formato PDF con las respuestas
 - Genere tantos ficheros Python como necesite con el formato `p2_pregunta_XX.py` donde `XX` es el número de pregunta

- Criterios generales de evaluación: texto conciso y claro; relación con teoría; numeración de diagramas, figuras y tablas; referencias a figuras y tablas en texto; referencias a fuentes externas (si procede); errores ortográficos.

- Memoria : preguntas (4)

P2.1 Aplicando los descriptores de tipo ‘hist’ de la tarea 2 sobre la imagen `camera()` de Skimage (paquete `skimage.data`, <https://bit.ly/2ZsXfN7>), analice como cambian los valores del descriptor calculado si utiliza:

- Tamaño de vecindario con valores 8 y 16
- Número de bins con valores 16 y 32

(0.5 puntos)

P2.2 Analice las correspondencias obtenidas en la tarea 3 con los descriptores ‘hist’ y ‘mag-ori’ sobre la primera imagen de test . Visualice y razone porque los cambios que observe en los experimentos que realice. **(0.5 puntos)**.

P2.3 Aplique las funciones de las tareas 1, 2 y 3 sobre los pares de imágenes proporcionados en la carpeta ‘img’. Para cada par de imágenes, discuta las diferencias en las esquinas devueltas (si existieran) y qué fenómenos pueden explicar las diferencias observadas. Incluya ejemplos visuales de los experimentos que realice. **(0.75 puntos)**

P2.4 Extienda la función desarrollada en la tarea 3 para considerar la correspondencia de puntos de interés con distancia mínima umbralizada alrededor del punto de interés (i.e. tarea 3) junto con el criterio *Nearest Neighborg Distance Ratio*. Considere lo siguiente:

- La distancia utilizada siempre será euclídea
- Identificar el método con `tipoDesc='nndr'`
- Utilice un umbral con valor 0.75

Incluya ejemplos visuales de los resultados de la nueva funcionalidad desarrollada.

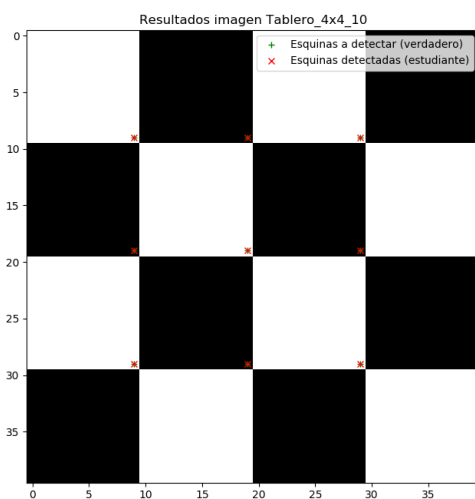
(1.25 puntos)

- ¿Cómo comprobar si la implementación es correcta?
 - Se proporciona funciones que se pueden ejecutar cuando se desee
 - **test_p2_tarea1(...)** en el fichero `p2_tarea1.py`
 - **test_p2_tarea2(...)** en el fichero `p2_tarea2.py`
 - **test_p2_tarea3(...)** en el fichero `p2_tarea3.py`
 - Parámetros comunes de las funciones
 - **disptime=-1** → muestra resultados visuales y espera '*disptime*' secs para avanzar al siguiente test (si igual 0, se para en cada ejecución)
 - **stop_at_error=True** → se para cuando se encuentra un error en la ejecución del código del estudiante
 - **debug=True** → se muestra información extendida por consola con el resultado correcto y los datos de entrada
 - **imgIdx=-1** → número/identificador de imagen sobre la que realizar el test (=-1, indica todas las imágenes del test)
 - Ejemplos de uso en cada fichero python

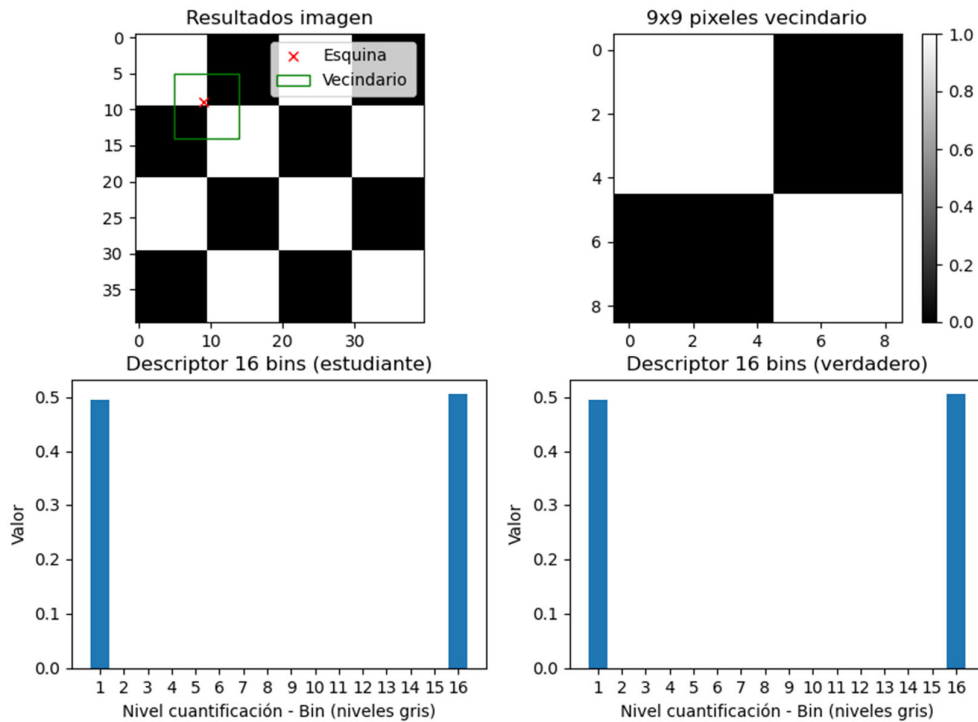
- Auto-evaluación **test_p2_tarea1(...)**

Practica 2 - Parte 1 - Comparativa puntos de interés con Harris

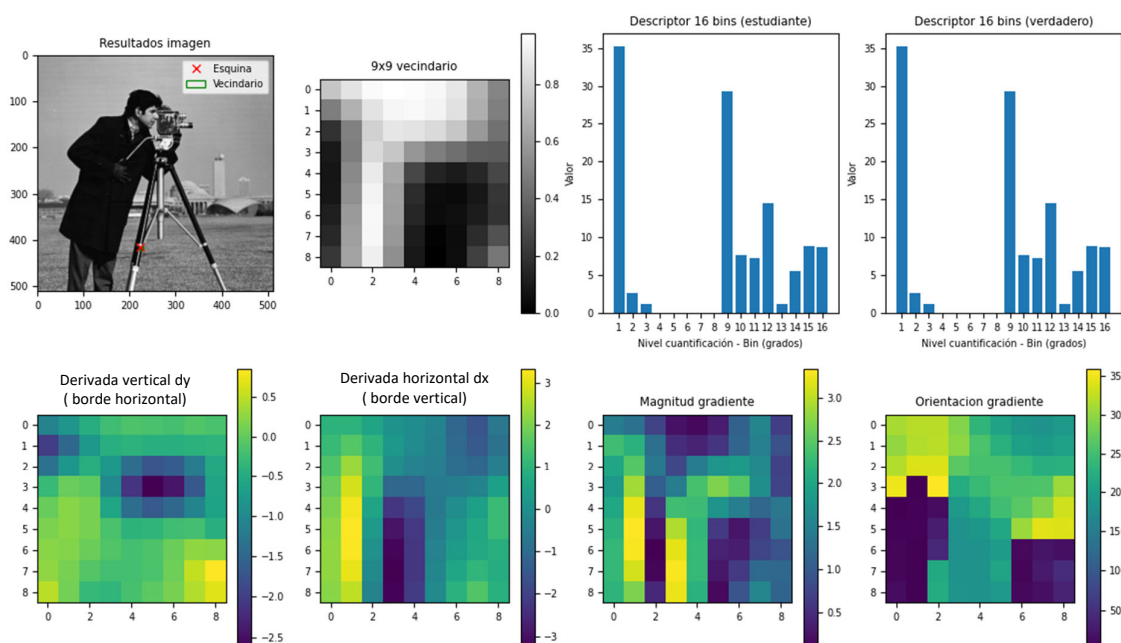
– □ ×



- Auto-evaluación `test_p2_tarea2(tipoDesc='hist'...)`

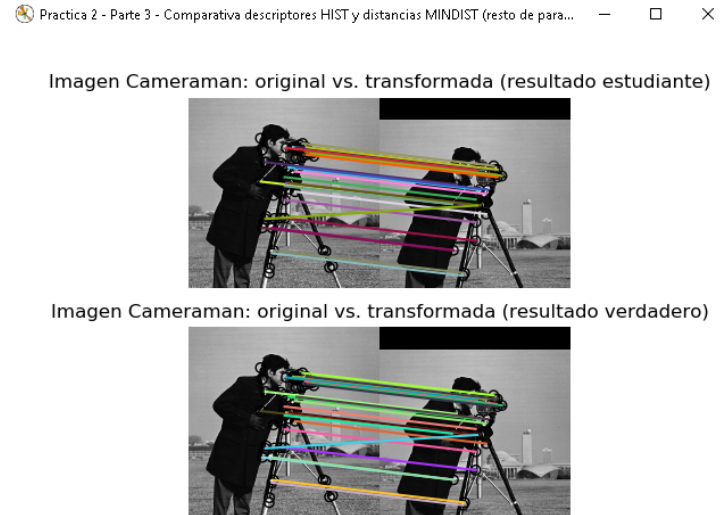
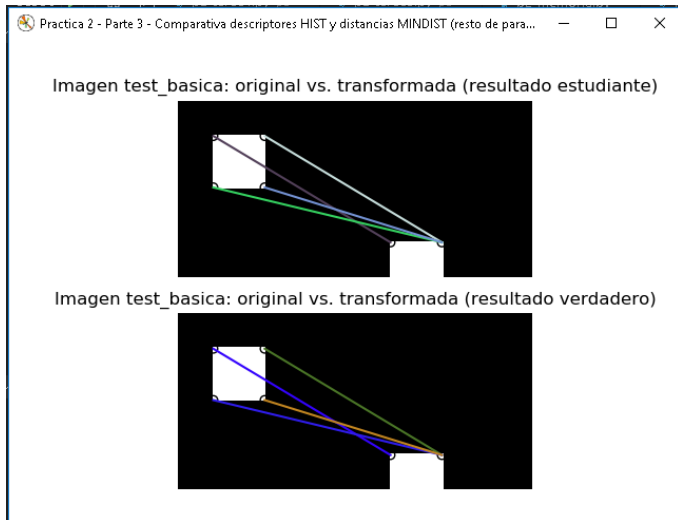


- Auto-evaluación `test_p2_tarea2(tipoDesc='mag-ori'...)`



- Auto-evaluación

```
test_p2_tarea3(tipoDesc='hist', tipoCorr='minDist')
test_p2_tarea3(tipoDesc='mag-ori', tipoCorr='minDist')
```



- 2 sesiones de prácticas (16 horas)
 - 4h presenciales (2h obligatorias + 2h opcionales)
 - 12h no presenciales

TAREA	Horas totales estimadas de trabajo
Explicación práctica	0.5 h
Tarea 1	1.5 h
Tarea 2a	1.5 h
Tarea 3a	1 h
Tarea 2b	5.5 h
Tarea 4 - Memoria	6 h
TOTAL	16h

- Evaluación de la práctica sobre 10 puntos

TAREA	Max nota	Criterio evaluado (ver rúbrica en Moodle)
Tarea 1	1.5	Código: Ejecución (60%) Código: Diseño & estilo (40%)
Tarea 2a	1.5	
Tarea 2b	2.5	
Tarea 3a	1.5	
Memoria	3	Memoria: Claridad, exactitud de la respuesta y experimentos realizados por cada pregunta (100%)
TOTAL	10	

- Penalizaciones:

- Por entrega de ficheros no acorde a las especificaciones: -0.5 puntos
- Por entrega de memoria no acorde a las especificaciones: -0.5 puntos
- Por uso de funciones prohibidas: -50% tarea
- Por entrega tardía (tras considerar los 4 días disponibles para cada pareja):
 - -25% (un día), -50% (dos días), -75% (tres días), -100%(>= días)

- Wikipedia https://en.wikipedia.org/wiki/Harris_Corner_Detector
- **Detector de Harris:**
 - Chris Harris and Mike Stephens (1988). "A Combined Corner and Edge Detector". Alvey Vision Conference
- **Descriptor Histograma de niveles de intensidad**
 - Manjunath, B. S., et al . (2001). Color and texture descriptors. IEEE Trans. on circuits and systems for video technology, 11(6), 703-715.
 - Alreshidi, E., et al (2021). A Comparative Study of Image Descriptors in Recognizing Human Faces Supported by Distributed Platforms. Electronics, 10(8), 915
- **Descriptor Histograma de Gradientes Orientados**
 - Richard Szeliski, "Computer Vision: Algorithms and Applications" 2020, **Sección 7.1.2.** <http://szeliski.org/Book/2ndEdition.html>
 - Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91-110
- **Metodos de búsqueda de correspondencias**
 - Richard Szeliski, "Computer Vision: Algorithms and Applications" 2020, **Sección 7.1.3.** <http://szeliski.org/Book/2ndEdition.html>