

# E4-Descriptores

---

## Instrucciones

En respuesta a las preguntas que se plantean a continuación, se debe entregar en la plataforma de enseñanza virtual (por separado, no en un archivo comprimido)

- un archivo con extensión .ipynb creado con notebook Jupyter de Anaconda que contenga, tanto las respuestas teóricas, como el código programado;
- el archivo anterior en .pdf (File->Print preview->guardar como pdf);
- una declaración de autoría firmada por el alumno/a según la plantilla proporcionada;
- las imágenes/vídeos usados.

La resolución de todos los ejercicios (tanto teóricos como prácticos) debe ser **razonada** y el código desarrollado debe ser **explicado en detalle, justificando** todos los parámetros escogidos y referenciando las fuentes.

## Objetivo

El objetivo de este entregable es el de la extracción de descriptores globales y locales y cómo pueden usarse para clasificación y otras aplicaciones.

### Ejercicio 1. Clasificación de texturas [0,6 pts]

Considera la base de datos de texturas de la Universidad del Sur de California, que puedes encontrar aquí: <https://sipi.usc.edu/database/database.php?volume=textures&image=58#top>

Escoge dos texturas distintas y varias imágenes de cada una (al menos 3 de cada).

- a) Cárgalas en escala de grises y calcula dos descriptores estadísticos que no dependan de la iluminación de la imagen (en el sentido de que si se aumenta el brillo de toda la imagen por igual, el descriptor siga dando el mismo resultado). Pueden ser descriptores explicados en clase u otros, explicándolos e incluyendo las referencias apropiadas.
- b) En base a los resultados, determina cuáles de estos descriptores podrían usarse para clasificar las texturas. Razona el porqué del éxito o el fracaso de cada uno.
- c) Suponiendo que al menos uno de los descriptores funciona bien ¿cómo podría diseñarse un método sencillo de clasificación supervisada para, dada una nueva imagen, determinar a qué clase pertenece? Nota: no hay que implementarlo, sólo explicarlo.

### Ejercicio 2. Descriptores geométricos [0,6 pts]

Toma 3 imágenes con una pelota sobre un fondo homogéneo y otras 3 de un cubo de Rubik sobre fondo homogéneo.

- a) Diseña, **razonadamente**, usando OpenCV, dos descriptores geométricos, independientes de la escala, que distingan los dos objetos por su forma. Calcúlalos para las 3 imágenes de cada tipo, para corroborar que los descriptores elegidos son una buena elección.
- b) ¿Hay algún descriptor topológico que pueda ayudar a distinguir estos objetos? Razónalo.

Ayuda: [https://docs.opencv.org/4.1.0/d3/d05/tutorial\\_py\\_table\\_of\\_contents\\_contours.html](https://docs.opencv.org/4.1.0/d3/d05/tutorial_py_table_of_contents_contours.html)

Las características geométricas que se explican aquí, podrían ser de utilidad:

[https://docs.opencv.org/4.1.0/dd/d49/tutorial\\_py\\_contour\\_features.html](https://docs.opencv.org/4.1.0/dd/d49/tutorial_py_contour_features.html)

### Ejercicio 3. Descriptores locales. Harris corners, SIFT y ORB [1,3 ptos]

En la documentación de OpenCV se puede encontrar información sobre cómo calcular Harris Corners: [https://docs.opencv.org/4.x/dc/d0d/tutorial\\_py\\_features\\_harris.html](https://docs.opencv.org/4.x/dc/d0d/tutorial_py_features_harris.html)

SIFT es otra opción para calcular puntos de interés en una imagen: [https://docs.opencv.org/4.x/da/df5/tutorial\\_py\\_sift\\_intro.html](https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html)

Debido a que este último método estaba patentado, surgió otra opción para el cálculo de descriptores locales o puntos de interés: ORB = Oriented FAST and Rotated BRIEF. [https://docs.opencv.org/4.x/d1/d89/tutorial\\_py\\_orb.html](https://docs.opencv.org/4.x/d1/d89/tutorial_py_orb.html)

- a) Explica, de forma concisa y esquemática, con tus propias palabras, en qué consiste cada uno de los tres métodos. Estas descripciones deben incluir:
  - 1. Las referencias completas de los artículos de investigación originales en los que se publicaron estos algoritmos (recuerda las indicaciones de la primera semana para acceder a artículos de investigación).
  - 2. Entrada de cada algoritmo.
  - 3. Los pasos fundamentales de cada método.
  - 4. Forma de la salida.
  - 5. Tipo de invarianza de los puntos calculados (a traslación, a rotación, a escala).

Se valorará la claridad de la explicación y que sea lo más didáctica posible (que no sea simplemente traducir partes de los artículos ni copiar explicaciones del tutorial).

- b) Haz un ejemplo práctico, ayudándote del tutorial de OpenCV y explicando el código usado, para calcular los tres tipos de puntos característicos a una misma imagen. ¿Qué ocurre si se gira la imagen?