

Laboratorio

Introducción a OpenCV y Espacios de Color

Visión por Ordenador I
Ingeniería Matemática

Laboratorio

Introducción a OpenCV y Espacios de Color

Profesor:

Erik Velasco
Lionel Güitta
Daniel Pinilla
Luis Arias
José María Bengochea
Mario Triviño

Email

evelasco@icai.comillas.edu
lglopez@icai.comillas.edu
dpinilla@icai.comillas.edu
learias@icai.comillas.edu
jmbengochea@icai.comillas.edu
mtrivino@comillas.edu

Cover:

Erosion, tectonic uplift, and a human-built dam have all helped shape the Upper Lake Powell area in Utah. Image Credit: NASA

Contents

| | | |
|---------------------|--|----------|
| 1 | Introducción al laboratorio | 1 |
| 1.1 | Requisitos | 1 |
| 1.2 | Metodología | 1 |
| 1.3 | Materiales | 1 |
| 1.4 | Entregas | 2 |
| 1.5 | Evaluaciones | 2 |
| 1.6 | Faltas al laboratorio | 2 |
| 2 | Sesión 1: Introducción a OpenCV y Espacios de color | 3 |
| 2.1 | Materiales | 3 |
| 2.2 | Apartados de la práctica | 3 |
| 2.3 | Observaciones | 3 |
| 2.4 | Qué va a aprender | 3 |
| 2.5 | Evaluación | 4 |
| 3 | Apartado A | 5 |
| Tarea A.1: | Carga de una imagen | 5 |
| Tarea B.2: | Detección de esquinas de las imágenes | 5 |
| Tarea A.3: | Conversion de imagen de color a escala de grises | 5 |
| Tarea A.4: | Cambio de tamaño de una imagen | 5 |
| Tarea A.5: | Recorte de una imagen | 5 |
| A.6: | Rotación de una imagen | 5 |
| A.7: | Escritura en disco de una imagen | 6 |
| 4 | Apartado B: Segmentación por color | 7 |
| Tarea A.1: | Carga de imágenes | 7 |
| Tarea A.2: | Visualización y guardado de imágenes | 7 |
| Tarea A.3: | Cambio de espacio de color | 7 |
| Tarea B.4: | Segmentación de las partes naranjas | 7 |
| Tarea B.5: | Segmentación de las partes blancas | 8 |
| Tarea B.6: | Combinación de máscaras | 8 |
| Tarea B.7: | Guardado de imágenes | 9 |
| Preguntas | | 9 |

Introducción al laboratorio

El laboratorio de *Visión por Ordenador I* es una oportunidad para poner en práctica los conocimientos teóricos de la asignatura. En este laboratorio se trabajará con técnicas de visión por ordenador clásica y se abarcarán los siguientes bloques:

- Sesión 1: Introducción a OpenCV y Espacios de Color (2 horas).
- Sesión 2: Calibración de cámaras (2 horas).
- Sesión 3: Procesamiento de imagen y Extracción de características (2 horas).
- Sesión 4: Bolsa de palabras visuales y Seguimiento de objetos (2 horas).

Además, las últimas 3 sesiones de laboratorio se dedicarán a la elaboración de un proyecto. Este contendrá los módulos trabajados durante el curso. El proyecto le ayudará a reforzar los conceptos estudiados en el aula y las sesiones de laboratorio. Puede encontrar inspiración en proyectos de cursos anteriores¹.

1.1. Requisitos

Es fundamental que, antes de iniciar la sesión, cada persona disponga del lenguaje de programación Python instalado en su equipo. Además, debe tener disponible un IDE para trabajar de forma eficiente². Si tiene problemas a la hora de realizar las instalaciones, por favor, comuníquese a uno de los profesores.

1.2. Metodología

El laboratorio se realiza en parejas. Se recomienda que ambas personas realicen la práctica (de forma colaborativa pero en ordenadores diferentes). Tenga en cuenta que los exámenes siempre contienen apartados dedicados a la elaboración de código, por lo que es importante que tenga soltura programando y consultando la documentación de las librerías.

1.3. Materiales

En algunas sesiones se trabajará con un Jupyter Notebooks (.ipynb), mientras que en otras se trabajará con scripts de Python (.py). En las prácticas finales se introducirá el uso de cámara para obtener y procesar sus propias imágenes.

Los materiales relacionados con software podrá encontrarlos en la sección de Moodle dedicada al laboratorio. Se facilitarán los materiales antes de cada práctica.

¹Proyecto Tangible UI en Github: <https://github.com/winoo19/wandavision>

²Un IDE muy extendido y recomendado es: VScode: <https://code.visualstudio.com/>

1.4. Entregas

Las entregas se realizarán a través de la sección habilitada en Moodle para cada sesión de laboratorio. La entrega deberá realizarse, como tarde, una semana después de la sesión trabajada. Se deberá entregar el código generado (ficheros .ipynb o .py completos) y un reporte en pdf.

1.5. Evaluaciones

En el guion de cada práctica encontrará una sección donde se especifique el peso de cada apartado. Así podrá saber cómo se ha evaluado su trabajo.

1.6. Faltas al laboratorio

La persona que se ausente de una sesión de laboratorio de forma injustificada obtendrá un 0 como calificación de esa sesión.

2

Sesión 1: Introducción a OpenCV y Espacios de color

2.1. Materiales

En esta práctica se trabajará con los siguientes recursos (puede encontrarlos en la sección de Moodle *Laboratorio/Sesión 1*):

- **lab_1.ipynb**: notebook con el código que deberá completar.
- **utils.py**: fichero con utilidades.
- **data**: carpeta con imágenes para trabajar durante la práctica.

2.2. Apartados de la práctica

La Sesión 1 del laboratorio está dividida en los siguientes apartados:

- Instalaciones: Instalación de librerías necesarias.
- Librerías: Importación de las librerías que se utilizan en la sesión. Se recomienda realizar la importación en una celda inicial para mantener la organización del Notebook.
- Apartado A: Introducción a OpenCV.
- Apartado B: Segmentación de imágenes por color.

2.3. Observaciones

Aunque el guion de la práctica y los comentarios en Markdown del Notebook estén escritos en español, observe que todo aquello que aparece en las celdas de código está escrito en inglés. Es una buena práctica que todo su código esté escrito en inglés.

Aquellas partes del código que deberá completar están marcadas con la etiqueta **TODO**.

Es muy importante que trabaje consultando la documentación de OpenCV¹ y *skimage*² para familiarizarse de cara al examen. Tenga en cuenta que en los exámenes no podrá utilizar herramientas de ayuda como Copilot.

2.4. Qué va a aprender

Al finalizar esta práctica, sabrá cómo trabajar con OpenCV y usar diferentes espacios de color para la segmentación de imágenes.

¹Documentación de OpenCV: <https://docs.opencv.org/4.x/index.html>

²Documentación de skimage: <https://scikit-image.org/docs/stable/>

2.5. Evaluación

La nota que obtenga en esta sesión de laboratorio será la misma que obtenga su pareja. Los apartados de la práctica serán evaluados como refleja la Tabla 2.1.

| Tarea | Valor | Resultado |
|--------------|-------------|-----------|
| Apartado A | 3.0 | |
| Apartado B | 3.0 | |
| Pregunta B.1 | 3.0 | |
| Pregunta B.2 | 1.0 | |
| Total | 10.0 | |

Tabla 2.1: Valoración de los apartados de la práctica.

3

Apartado A

En este apartado se trabajará en la utilización de funciones básicas de OpenCV. Posteriormente, en el apartado B, se utilizarán estas funciones para construir métodos que le serán de utilidad en el futuro.

Para este ejercicio, las imágenes se encuentran en la carpeta *data*. De nuevo, se recomienda consultar la documentación de OpenCV para ver como usar las funciones utilizadas y sus argumentos: [OpenCV Documentation](#)¹.

Tarea A.1: Carga de una imagen

Utilice la función `cv2.imread()` para cargar una imagen de la carpeta *data*.

Tarea A.2: Visualizar una imagen

Utilice los métodos `cv2.imshow()`, `cv2.waitKey()` y `cv2.destroyAllWindows()` para visualizar la imagen por pantalla.

Tarea A.3: Conversión de imagen de color a escala de grises

Utilice el método `cv2.cvtColor()` para realizar la conversión de BGR a escala de grises. Para obtener el tamaño de la imagen (alto y ancho) y el número de canales, utilice `.shape`.

De nuevo, represente la imagen para verificar que se ha realizado correctamente la conversión a escala de grises.

Tarea A.4: Cambio de tamaño de una imagen

Utilice la función `cv2.resize()` para cambiar el tamaño de la imagen a 200x200 píxeles. Como en la tarea anterior obtenga el tamaño de la imagen resultante para verificar que se ha realizado correctamente el cambio de tamaño. Por último, visualice la nueva imagen obtenida con el nuevo tamaño.

Tarea A.5: Recorte de una imagen

Para recortar el centro de una imagen a un tamaño de 120x160 píxeles, primero es necesario conocer las dimensiones originales de la imagen (alto y ancho). A partir de ellas se calcula el punto central y, desde ahí, se determinan las coordenadas de inicio y fin del recorte. El proceso consiste en aplicar un corte (slicing) sobre la matriz de la imagen para obtener únicamente la región central con el tamaño deseado.

Tarea A.6: Rotación de una imagen

Para rotar una imagen en OpenCV se utiliza la combinación de dos funciones: `cv2.getRotationMatrix2D()` y `cv2.warpAffine()`. Con `cv2.getRotationMatrix2D()` se genera la matriz de transformación de

¹Doc. OpenCV <https://docs.opencv.org/4.8.0/index.html>

rotación, indicando el punto central de la imagen, el ángulo de rotación y un factor de escala (normalmente 1.0 si no se quiere redimensionar). Luego, con `cv2.warpAffine()` se aplica dicha transformación a la imagen original, produciendo la versión rotada. Esta función necesita la matriz de rotación y el tamaño de salida (ancho y alto de la imagen).

Tarea A.7: Escritura en disco de una imagen

Utilice el método `cv2.imwrite()` para realizar la escritura de la imagen en un fichero en disco.

4

Apartado B: Segmentación por color

En este apartado deberá segmentar los colores blancos y naranjas que aparecen en las imágenes disponibles en la carpeta data. Segmentar por colores es una técnica sencilla y útil en aplicaciones en las que se tiene un gran control sobre las condiciones de contorno: iluminación, tipo de objeto que se espera encontrar, color del fondo, etc. A continuación se añaden algunos comentarios para resolver paso a paso la tarea:

Tarea B.1: Carga de imágenes

Para trabajar de forma eficiente, antes de empezar a procesar imágenes deberá definir algunos métodos auxiliares. El primero de ellos es un método que debe cargar imágenes de la carpeta data.

Tarea B.2: Visualización y guardado de imágenes

De nuevo, estos métodos ya los utilizó en la 1^a sesión de laboratorio, así que aproveche este paso para afianzar conceptos. Defina los métodos para visualizar imágenes y guardarlas. Respete y utilice todos los argumentos que se proporcionan.

Tarea B.3: Cambio de espacio de color

Cambie el espacio de color de las imágenes (BGR) a uno donde la crominancia e intensidad estén separados (HSV). Puede interpretar este paso como un cambio de sistema de referencia (cartesiano a polar). De esta forma, le será más fácil discriminar los colores de interés. La Figura 4.1 ¹ muestra los espacios de colores con los que trabajará.

Para cambiar el espacio de color de sus imágenes, puede utilizar el método `cv2.cvtColor()` de OpenCV. Tenga en cuenta que OpenCV cuenta con una gran cantidad de espacios de color, visite la documentación.²

Tarea B.4: Segmentación de las partes naranjas

Genere una máscara para cada imagen y realice una segmentación con ella. Una máscara es una imagen binaria (blanca y/o negro) donde los píxeles de interés aparecen en blanco, mientras que los píxeles que no son de interés aparecen en negro. Segmentar implica multiplicar la imagen original con la máscara. Así se obtienen los píxeles de interés de la imagen original.

Para obtener las máscaras y segmentar la imagen, puede utilizar respectivamente los métodos `cv2.inRange()` y `cv2.bitwise_and()` de OpenCV. Por otro lado, en este paso se facilitan los rangos de color naranja:

¹Imagen modificada de VerbaGleb, CC BY-SA 4.0 https://commons.wikimedia.org/wiki/File:RGB_2_HSV_conversion_with_grid.ogg

²Espacios de color de OpenCV https://docs.opencv.org/3.4/d8/d01/group__imgproc__color__conversions.html

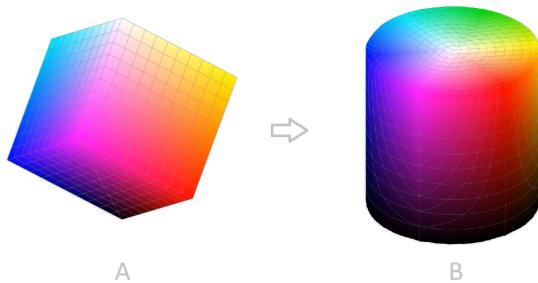


Figure 4.1: Espacios de color utilizados. Gráfico A: BGR (atención con el orden de canales en OpenCV) y Gráfico B: HSV.

```
light_orange = (1, 190, 200)
dark_orange = (255, 255, 255)
```

Para confirmar que está obteniendo los resultados adecuados, muestre una de las imágenes iniciales, su máscara y la imagen segmentada.

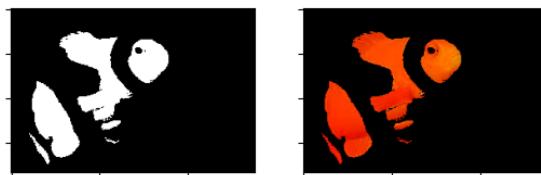


Figure 4.2: Máscara y segmentación de colores naranjas.

Tarea B.5: Segmentación de las partes blancas

En este paso deberá repetir un proceso similar al anterior, pero centrándose en los tonos blancos. Explore el rango de valores necesarios para segmentar los colores blancos de las imágenes. Para ello, juegue con los valores de H, S y V en la interfaz que se le proporciona.

De nuevo, para confirmar que está obteniendo los resultados adecuados, muestre la máscara y la imagen segmentada.

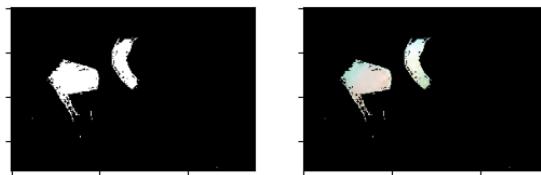


Figure 4.3: Máscara y segmentación de colores blancos.

Tarea B.6: Combinación de máscaras

Combine las máscaras de los colores naranja y blanco para cada imagen. Posteriormente, segmente cada imagen con la máscara que ha generado. Como resultado, conseguirá segmentar el cuerpo de los peces en el conjunto imágene. Debería obtener resultados parecidos al ejemplo de la Figura 4.4.

Confirme que está obteniendo los resultados adecuados mostrando la máscara y la imagen segmentada.

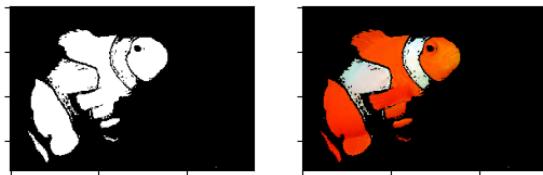


Figure 4.4: Máscara completa y segmentación del pez como combinación de colores blancos y naranjas.

Tarea B.7: Guardado de imágenes

Guarde las imágenes de interés como resultado de esta parte de la práctica.

Preguntas

Pregunta B.1

Segmenta por color el escudo de su equipo deportivo favorito: descompóngalo en al menos 2 colores. De igual forma que ha realizado en el ejercicio de clase, proporcione las máscaras binarias de cada color. Además, multiplique la máscara con la imagen original para visualizar solamente el color de interés. Por último indique el porcentaje que representa cada color sobre el total del escudo.

Pregunta B.2

¿Qué ocurre cuando carga las imágenes con la librería imageio pero las visualiza con la librería OpenCV? Investigue qué puede estar pasando y cómo podría solucionarlo. Pista: cv2.cvtColor().