

# Práctico: Medición de objetos usando CV

## Visión por Computadoras

Ingeniería en Electrónica

24 de abril de 2019

El objetivo de este práctico es diseñar un programa que permita realizar la medición en la escala real de objetos que se encuentran sobre un plano conocido.

La elección del método a utilizar queda a criterio del estudiante, aunque les sugerimos algunas opciones posibles.

- La primera opción puede ser utilizar un objeto de referencia de dimensiones conocidas y medir estableciendo relaciones de tamaño entre el objeto a medir y el objeto de referencia. Para usar este método el eje focal de la cámara debe estar perpendicular al plano donde se encuentran los objetos. Ver figura 1.
- La segunda opción es usando un patrón de calibración como se muestra en la figura 2. En este caso no es necesaria la condición de perpendicularidad entre el eje focal y el plano de la escena. Notar que para este caso la posición de la cámara respecto del plano de la escena puede obtenerse siempre que el patrón de calibración se encuentre en la imagen, esto permite que la cámara pueda moverse mientras se realizan las mediciones.
- La tercera opción, similar a la anterior, es calibrar el plano usando un patrón, pero luego quitándolo para realizar las mediciones. Esto implica que la posición de la cámara respecto del plano de medición no puede alterarse. Este método es el más usual en un entorno industrial, donde el sistema una vez calibrado no se modifica al realizar las mediciones.



Figura 1: Ejemplo usando un objeto de dimensiones conocidas.

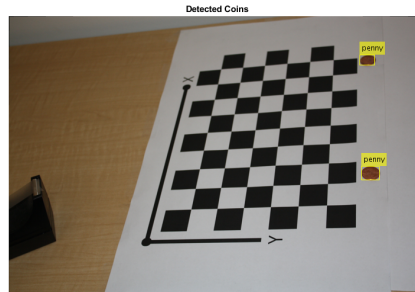


Figura 2: Ejemplo usando un patrón de calibración.

Además de la elección del método de medición, debemos resolver el problema de la detección de los objetos, tema que aún no desarrollamos en el curso y que debe ser investigado. Para esto existen varias opciones, algunas de ellas pueden ser:

- Detección de contornos:  
[https://docs.opencv.org/3.3.1/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html).
- Detección de blobs:  
<https://www.learnopencv.com/blob-detection-using-opencv-python-c/>,  
<https://www.makehardware.com/2016/05/19/blob-detection-with-python-and-opencv/>.
- Detección de formas:  
<https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/>.