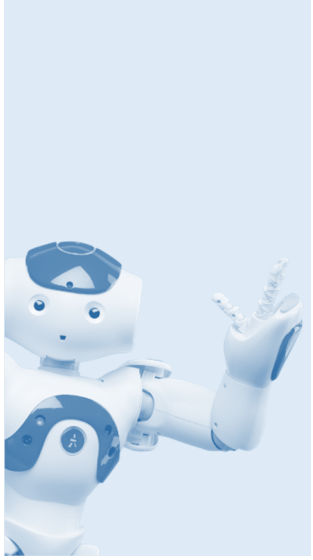


PRÁCTICA 1
Reconocimiento de gestos

Universitat d'Alacant
Universidad de Alicante

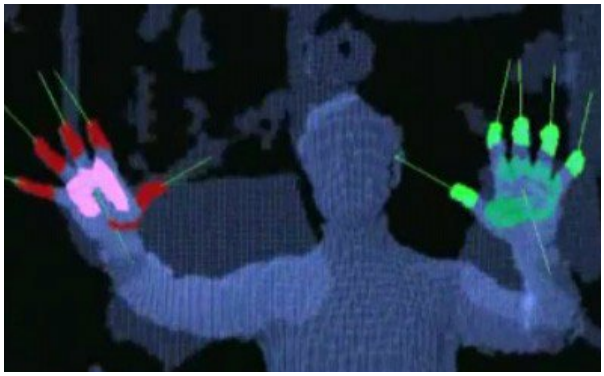
Objetivos de la práctica

- Conocer las características de tratamiento de imágenes en tiempo real de la plataforma de código abierto Bonsai
- Estudiar el funcionamiento de funciones básicas de tratamiento de imágenes 2D para reconocimiento de objetos
- Saber diseñar distintos esquemas de reconocimiento de elementos en un tiempo real y conocer como procesar, guardar y enviar los valores obtenidos con Bonsai

Profesor: Andrés Úbeda Castellanos

Introducción

El **reconocimiento de gestos** permite generar interfaces que interpretan gestos humanos como comandos. En otras palabras, es la habilidad de un ordenador para interpretar gestos generados por un operador humano y ejecutar comandos basándose en ellos. En esta práctica se van a estudiar y analizar herramientas de análisis de visión por computador a través de la herramienta de código abierto **Bonsai** que, además de otras muchas funciones, permite procesar imágenes de vídeo en tiempo real para extraer características de las mismas y obtener información de los objetos en la escena.



Software de código abierto Bonsai

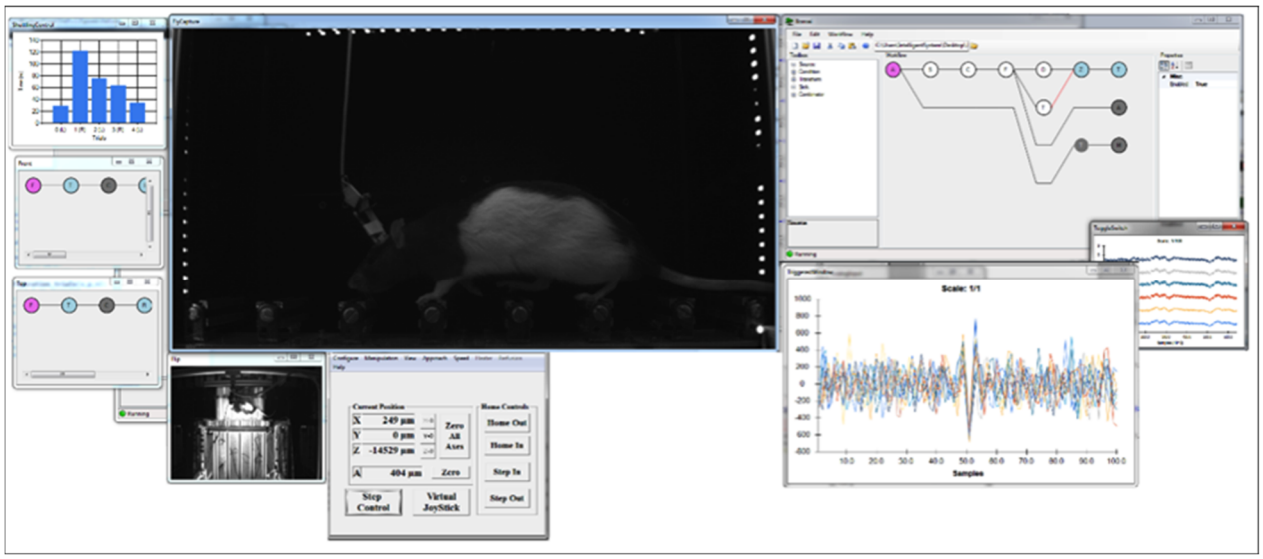
Bonsai es un software de código abierto para procesar flujos de datos heterogéneos (http://www.open-ephys.org/bonsai). Es una herramienta ideal para el análisis de vídeo en tiempo real, pero también permite controlar simultáneamente un experimento y/o adquirir datos diversos a través de una tarjeta de adquisición *Open Ephys*. Bonsai ha sido desarrollado por **Gonçalo Lopes** en el Champalimaud Centre for the Unknown, en Lisboa.

BONSAI

La configuración de Bonsai es tan sencilla como manipular elementos gráficos para generar el flujo de procesamiento deseado. Cada elemento o nodo representa una operación modular en un flujo de datos. Los nodos pueden conectarse para realizar tareas más complejas, como por ejemplo hacer un seguimiento de un objeto en un área determinada.

Algunas de las principales características de Bonsai son las siguientes:

- Desarrollado totalmente en C#
- Usa Microsoft Reactive Extensions (Rx) de forma interna.
- El número de operaciones simultáneas sólo está limitado por el sistema y el hardware disponible.
- Interfaz intuitiva y fácil de entender.
- Numerosos algoritmos de procesamiento de imagen y de señal incluidos por defecto.
- Cada paso de procesamiento se puede visualizar mientras el proceso global está ejecutándose.



Instalación

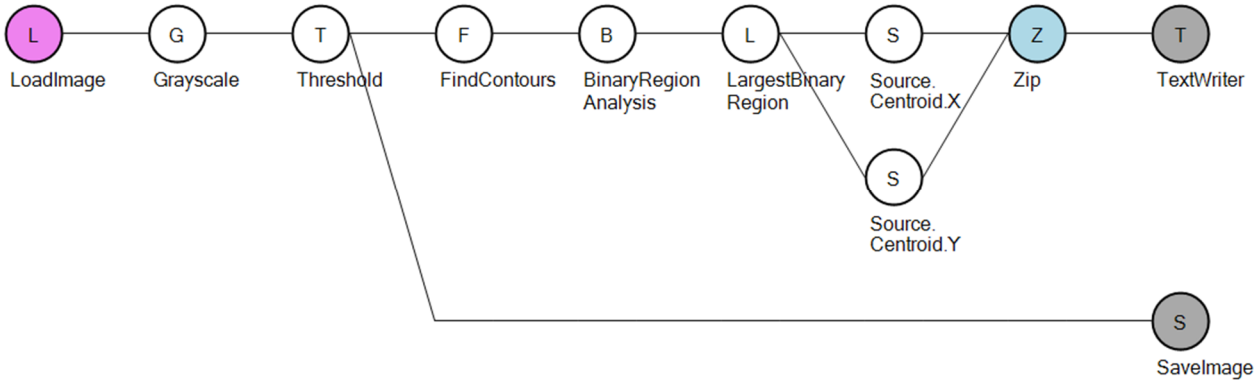
Para poder realizar procesado de vídeo en tiempo real se debe instalar el programa y el paquete Vision que contiene funciones de procesamiento de imagen. Para ello se deben seguir estos pasos:

- Instalar la versión 2.3 de Bonsai que se puede descargar de la siguiente página: https://bitbucket.org/horizongir/bonsai/downloads/
- Ejecutar el programa
- En el menú Tools -> Manage Packages, instalar el paquete Bonsai – Vision Library.
- Cuando el paquete esté instalado, aparecerán las funciones de visión entre los nodos seleccionables en el menú contextual de la izquierda

Un primer ejemplo con Bonsai

En este primer ejemplo con Bonsai se muestra cómo podemos binarizar una imagen y extraer de ella la posición de un objeto en la escena. Para ello, se proporciona una imagen de ejemplo llamada "pingpong_ball.jpg".

Paso 1. Crear un nuevo proyecto en Bonsai y construir el siguiente diagrama de flujo



Lista de nodos y ubicación:

- Source.Vision: LoadImage
- Transform.Vision: Grayscale, Threshold, FindContours, BinaryRegionAnalysis, LargestBinaryRegion
- Combinator.Reactive: Zip
- Sink.IO: TextWriter
- Sink.Vision: SaveImage

Los nodos Source.Centroid.X y Source.Centroid.Y forman parte del output del nodo LargestBinaryRegion. Para generarlos hay que pinchar con el botón derecho en el nodo en cuestión y seleccionarlos en la lista de outputs.

Paso 2. Modificar los siguientes nodos

- *LoadImage*: modificar la ruta para que lea la imagen "pingpong_ball.jpg".
- *SaveImage*: modificar la ruta para que se guarde la imagen "pingpong_bal_bin.jpg".
- *Threshold*: modificar el umbral hasta que se separe bien la pelota del fondo. Puedes comprobar el resultado obtenido en la imagen generada por el nodo SaveImage.
- *TextWriter*: modificar el nodo para que se guarde la información de la posición de la pelota en el archivo "coordenadas.txt".

Paso 3. Ejecutar el programa y comprobar el resultado

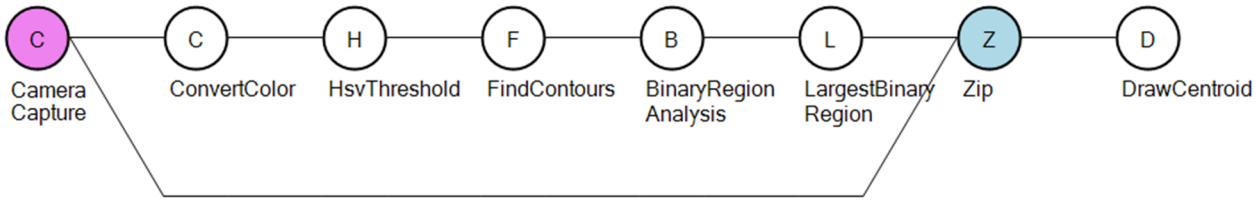
Paso 4. Utilizar otras imágenes y jugar con los umbrales para familiarizarse con el flujo de procesamiento de Bonsai.

Procesamiento en tiempo real con Bonsai

En este segundo ejemplo se van a explorar las características en tiempo real que tiene Bonsai. Para ello, se va a implementar un diagrama que permita detectar objetos de un color determinado en la escena.

Nota: *para ejecutar este ejemplo es necesario disponer de webcam. Si no se dispone de webcam se puede introducir el vídeo pregrabado "Axis.avi" como entrada usando el nodo Vision.FileCapture*

Paso 1. Crear un nuevo proyecto en Bonsai y construir el siguiente diagrama de flujo

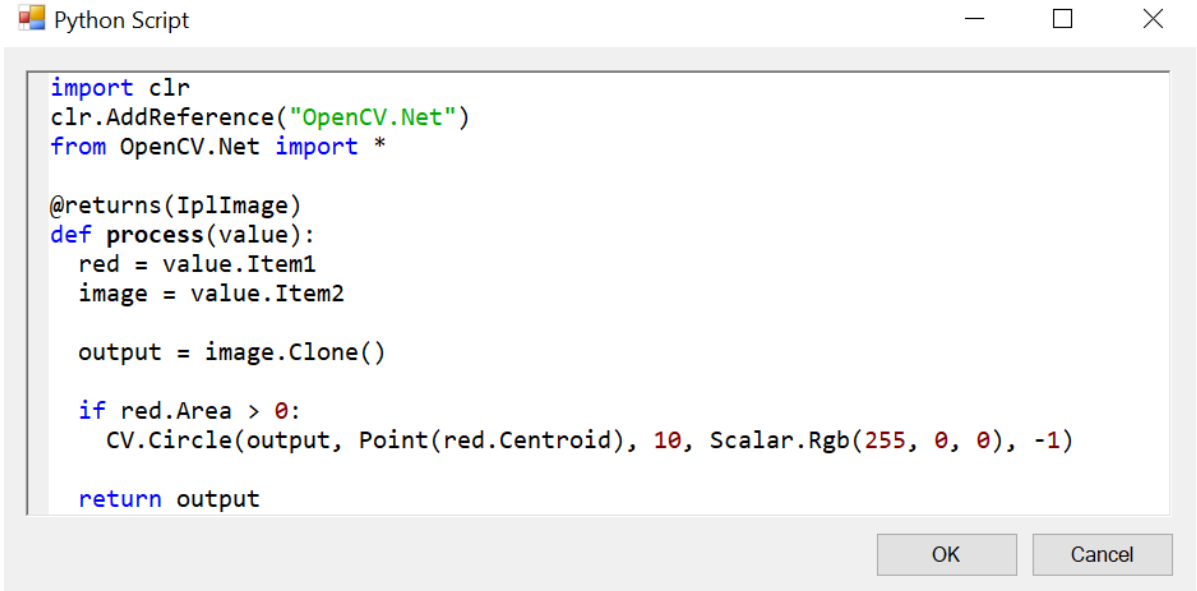


Lista de nodos y ubicación

- Source.Vision: CameraCapture
- Transform.Vision: ConvertColor, HsvThreshold, FindContours, BinaryRegionAnalysis, LargestBinaryRegion
- Combinator.Reactive: Zip
- Transform.Scripting: PythonTransform (cambiar nombre al nodo por DrawCentroid)

Paso 2. Modificar los siguientes nodos

- *HsvThreshold:* Lower (150, 160, 45, 0), Upper (179, 255, 255, 255)
- Introducir el siguiente script en el nodo DrawCentroid. El script recoge los datos de posición del centroide del objeto rojo obtenido en LargestBinaryRegion y dibuja un círculo rojo sobre el objeto en la imagen de vídeo en tiempo real.



Paso 3. Ejecutar el programa y comprobar el resultado

Paso 4. Modificar el programa para que detecte un objeto verde y muestre su posición en pantalla.

Paso 5. Modificar el programa para que detecte un objeto rojo y un objeto verde y muestre la posición de ambos en pantalla.

Escritura de datos en Bonsai

Como hemos visto, Bonsai es una herramienta que permite el procesamiento de imágenes en tiempo real mediante nodos predeterminados y también permite la introducción de código propio en Python para realizar procesamientos específicos. No obstante, puede ser necesario enviar datos en tiempo real a programas o hardware externo que realicen procesamientos más avanzados o que utilicen los datos procesados para realizar ciertas acciones, por ejemplo, dar información a un robot sobre elementos de la escena o datos de posición para mover sus articulaciones.

Una forma de hacerlo es usar las funciones de la librería *Sink.IO*

- *TextWriter:* este nodo se ha visto ya en el ejemplo 1 y permite escribir en un *txt* datos obtenidos en el flujo de Bonsai. Un programa externo puede ir leyendo ese *txt* generado para realizar sus propias acciones.
- *CsvWriter:* del mismo modo, este nodo puede escribir datos en un archivo *csv* que después puede ser leído por un programa externo.
- *SerialStringWrite:* otra forma de hacerlo es escribir datos por el puerto serie. En las propiedades del bloque se seleccionará el COM por donde se quieren enviar los datos y el programa o hardware externo se conectará a ese puerto serie para leer los datos.

Actividad

Realizar un diagrama de flujo en Bonsai que detecte la posición del dedo índice y del dedo pulgar en una escena en que el usuario apunta con el dedo índice con un ángulo de 90º entre ambos dedos. Una vez detectados ambos dedos, realizar un script que permita evaluar la dirección en la que apunta el dedo índice del usuario en grados.

Una solución para discriminar un dedo respecto del otro es el uso de patrones de color o con formas determinadas. También se pueden implementar algoritmos más complejos que detecten la forma de la mano y su dirección como una región específica en la escena.

Añadir todas las mejoras de visualización y procesado que se deseen.