

PROYECTO PROE

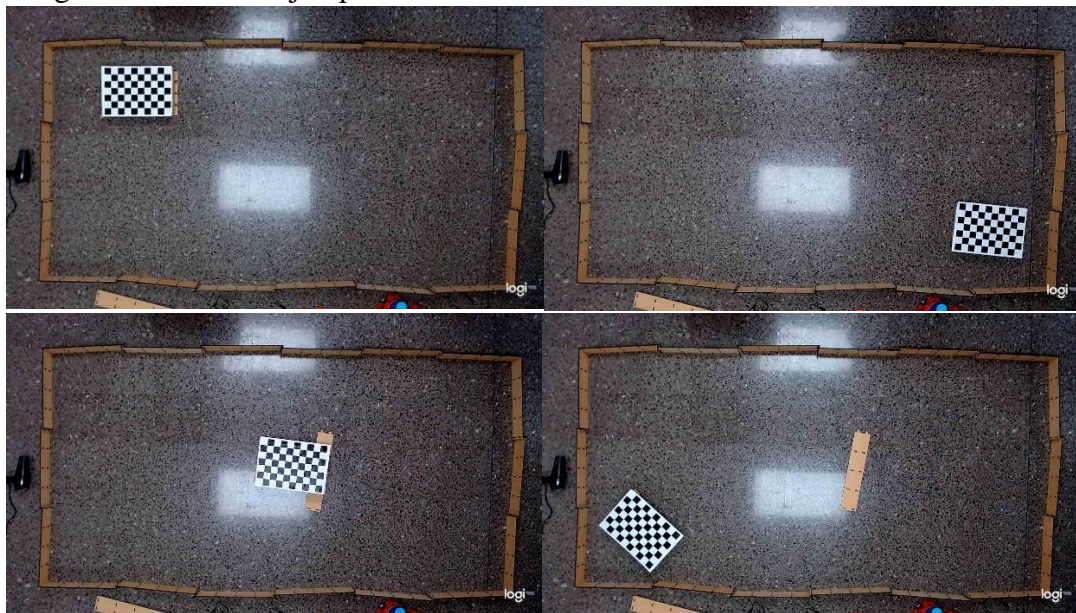
Manual de uso del Sistema de Visión (última actualización: 05/02/2023)

CONSIDERACIONES GENERALES

Para ejecutar los códigos se debe tener un ambiente programación como Jupyter Notebooks o Google Colab. Alternativamente, se pueden descargar todas las librerías requeridas y utilizar Python de forma local. Además, se debe contar con el software Logitech Capture para utilizar la webcam.

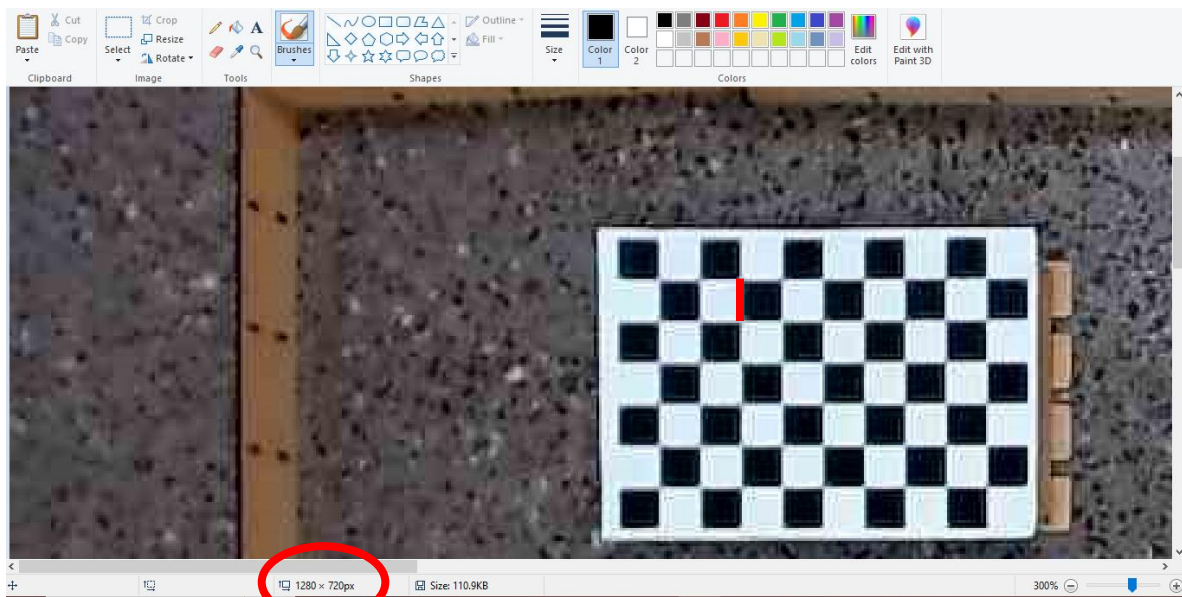
CALIBRACIÓN

1. Abrir el repositorio de PROE y buscar en la sección Herramientas/Cámara/VALIDACION_PYTHON/Calibracion Python.
2. Se debe disponer de al menos 10 imágenes de calibración utilizando el patrón correspondiente y colocándolo en diferentes posiciones. Las mismas deben ser tomadas con la webcam en la posición definitiva que se vaya a usar. Se adjuntan 4 imágenes a modo de ejemplo:



3. Agregar las imágenes de calibración al folder llamado "Imagenes_Calibracion_Nueva_Sala".
4. Ejecutar el código llamado "Calibracion_Camara"
5. Se van a generar 2 archivos .txt en el folder, estos corresponden a las matrices de calibración necesarias.

6. Copiar ambos .txt y pegarlos dentro del folder Herramientas/Cámara/VALIDACION_PYTHON/ Validación Python.
7. Al finalizar la calibración, se debe ajustar el ratio de mm/píxeles dentro del código de Validacion. Esto se puede realizar con una de las imágenes de calibración tomadas en el punto 2. Se debe abrir la imagen en Paint, en el mismo se pueden visualizar los píxeles en forma de coordenadas (x,y) en la parte inferior, tal como se muestra en la imagen. Calcular la longitud del lado de un cuadrado del patrón (en píxeles) utilizando la fórmula para la distancia entre 2 puntos. Se sabe de antemano que el lado mide 40 mm.



$$\text{ratio [mm/píxeles]} = \frac{40 \text{ mm}}{\text{Longitud del lado en píxeles}}$$

8. Por último, actualizar valor en el código para la variable mmPixel, en ambas líneas. Tal y como se muestra a continuación.

```
def calcular_desplazamiento(centrosC, centrosM, centrosV, centrosA, centrosN, numImagenes, lista_robots):
    #resolución espacial para imagen con zoom [mm/pixel]
    mmPixelZ=40/54
    #resolución espacial para imagen sin zoom [mm/pixel]
    mmPixel=40/23
```

```

#Dirección de imágenes prueba Bryan
#dir = 'Imagenes/'
#Dirección de imágenes de prueba Rebeca
dir = nombreCarpetaOut
#Guarda Los nombres de las imágenes que estaban en la carpeta en la lista imgList
imgList = os.listdir(dir)
#Inicializa la lista que contendrá a las imágenes que se utilizarán
imagenes = []
imagenesHSV = []
#Ordena las imágenes por número de menor a mayor
imgList.sort(key = lambda x: int(x.replace("frame", "").split('.')[0]))
#Inicializa contadores
n = 0
cont = 0
ind = 0
#resolución espacial para imagen con zoom [mm/pixel]
mmPixelZ=40/54
#resolución espacial para imagen sin zoom [mm/pixel]
mmPixel=40/23

```

9. El sistema de visión está listo para ser usado. Se debe agregar un video a la carpeta Herramientas/Cámara/VALIDACION_PYTHON/Validación Python. El nombre se debe actualizar en la variable “nombreVideoIn”. Tener en cuenta las líneas señaladas en azul, dado que cada vez que se ejecuta un nuevo video el código lo descompone en imágenes y las guarda en la carpeta “ImagenesRbk”. Se deben descomentar esta líneas únicamente cuando se cambie de video y adicional a esto, se deben eliminar todas las imágenes de la carpeta “ImagenesRbk” antes de ejecutar el código con un nuevo video.

```

return n

#----- Lectura de imágenes-----
#Nombre video en la carpeta de Segmentación por color #REBECA
nombreVideoIn='2022-10-19_16-42-34.mp4'
nombreCarpetaOut='ImagenesRbk/'

#-----PARA PROBAR OTRO VÍDEO
#Obtener Frames de un video: Solo se ejecuta si es la primera vez con este video
#numFrames=obtenerFrames(nombreVideoIn, nombreCarpetaOut)
#-----

```