



**Neil Otniel Moreno Rivera**

**Visión por Computadora**

## **Tarea 1: Multiumbralización de 3 objetos**

### **Introducción**

El objetivo de este trabajo fue desarrollar un programa que capture video en tiempo real desde una cámara Intel RealSense y detecte tres objetos de colores distintos dentro del campo de visión, mostrando un bounding box que los delimite (Con un archivo de prueba con un video previamente tomado). Los objetos están relacionados funcionalmente: una hoja de papel (roja), una regla (blanca) y un marcador (negro). La tarea simulada podría ser dibujar sobre la hoja con el marcador una línea paralela a la regla.

Para el desarrollo se exploraron 2 enfoques:

- Segmentación por color y umbralización simulada.
- Segmentación con preprocesamiento y umbralización multi-otsu.

### **Enfoque por color y umbralización simulada:**

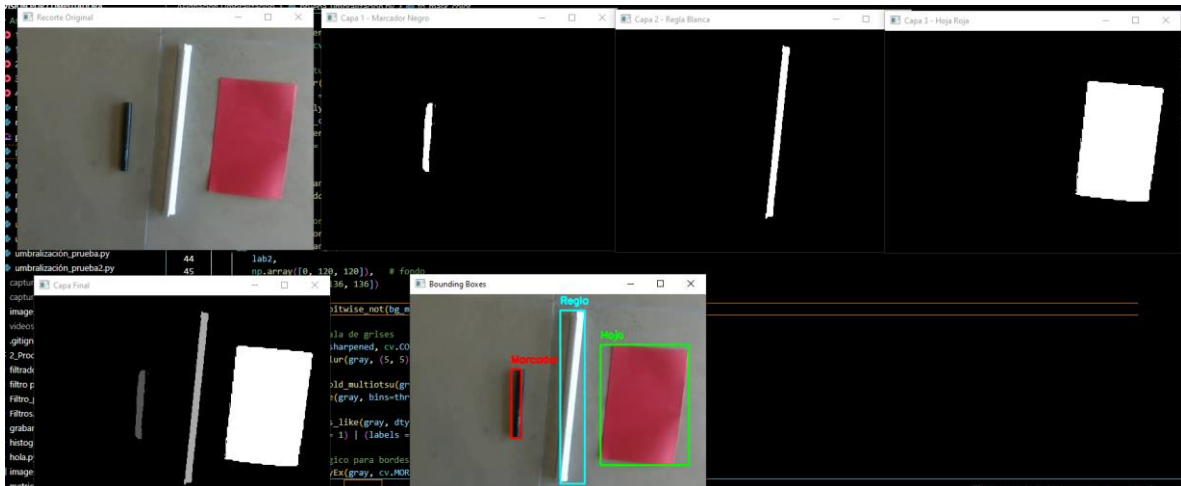
Archivos representativos:

- `real_sense_video_prueba.py` – Versión funcional con RealSense.
- `mejora_video.py` – Versión con video de archivo (base para adaptación).

Dentro de ambos scripts se siguen los siguientes pasos:

- Se captura el frame y se recorta la región central para eliminar bordes con posible ruido.
- Conversión a espacio HSV y definición de rangos para cada color:
  - Rojo (hoja): dos rangos (0–10 y 160–180) para cubrir toda la gama.
  - Blanco (regla): baja saturación y alto valor.
  - Negro (marcador): bajo valor.
- Se generan máscaras binarias para cada color.
- Se calculan contornos y se dibujan bounding boxes con etiquetas ("Hoja", "Regla", "Marcador").





*Imagen de salida de mejora\_video.py*

En la imagen anterior, se observa que la salida en un entorno controlado es buena, se pueden dividir los 3 objetos sin mayor problema, el mayor inconveniente que se puede esperar es la sensibilidad a los cambios de luz y sombras, por eso puede requerir calibración manual o una implementación para la calibración, su principal ventaja frente al otro, es la rapidez y facilidad de lectura.

### **Enfoque Complejo: Preprocesamiento + Multi-Otsu**

Archivos representativos:

- `real_sense_otsu_tiempo_real.py` – Versión completa con RealSense y técnicas avanzadas.
- `prueba_umbralizacion.py` – Versión con video de archivo (base de pruebas).

Dentro de ambos scripts se siguen los siguientes pasos:

#### **1. Preprocesamiento:**

- Eliminación de sombras mediante separación del canal L en LAB, sustracción del fondo suavizado y normalización.
- Incremento de saturación en HSV (factor 1.4) para realzar colores.
- Filtro de sharpen (unsharp mask) para mejorar bordes.

#### **2. Segmentación:**

- Eliminación de fondo por color (en LAB) usando un rango fijo (ajustado al color del entorno).



- Umbralización multi-Otsu sobre la imagen en escala de grises (3 clases, 4 divisores) para separar objetos del fondo.
- Detección de objetos negros mediante gradiente morfológico.

### 3. Post-procesamiento:

- Unión de máscaras (fondo eliminado + Otsu + gradiente).
- Relleno de huecos con floodFill.
- Operaciones morfológicas (cierre y apertura) para unir regiones y eliminar ruido.

### 4. Bounding boxes:

- Cálculo de contornos sobre la máscara final.
- Filtrado por área mínima (1200 píxeles) y dibujo de rectángulos genéricos ("Objeto + número).

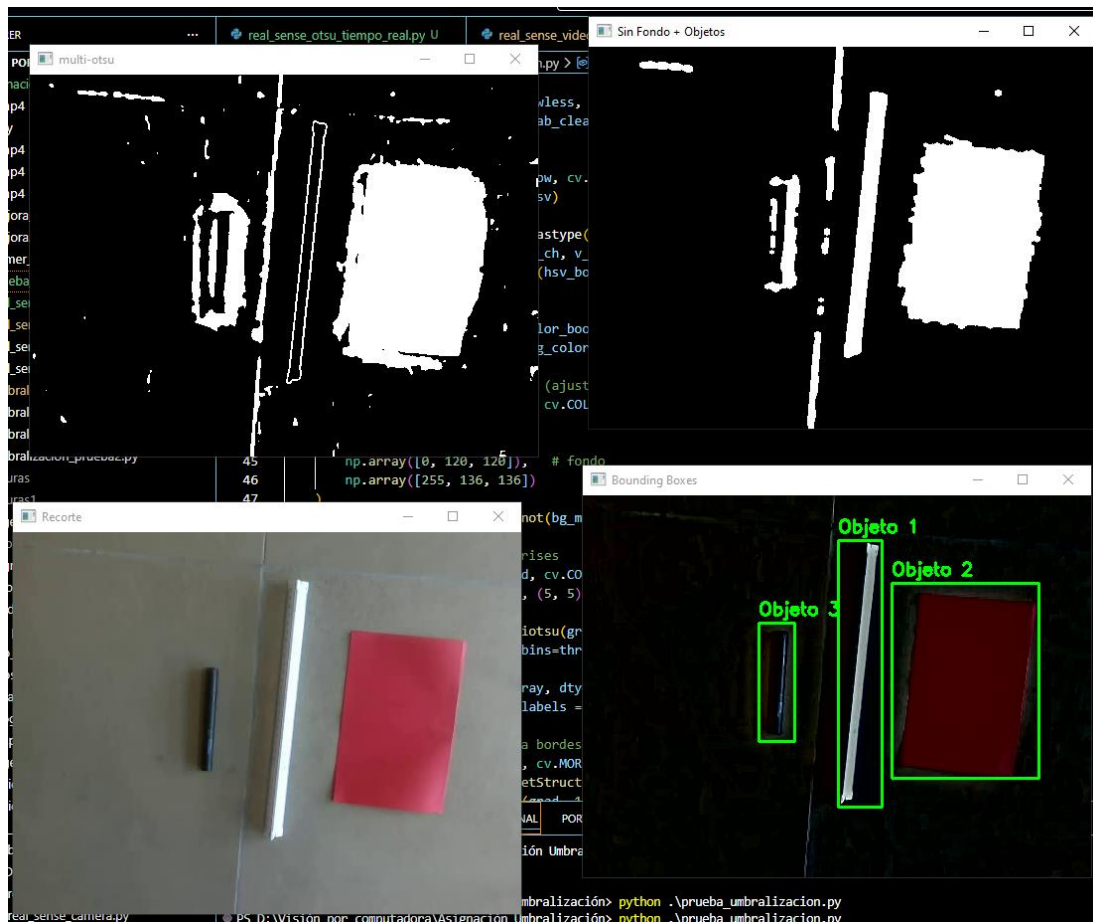


Imagen de salida de prueba\_umbralizacion.py



En la imagen anterior, se muestra la salida del programa prueba\_umbralizacion.py, en el cual se ve una correcta segmentación y etiquetado genérico, una ventaja que se gana al desarrollarlo de esta manera es que se aumento la resistencia a las sombras o variaciones de luz, tambien es que no depende de valores fijos a diferencia de la implementación anterior, la mayor desventaja es su costo computacional debido a los filtros y al otsu, en las pruebas el otsu con más de 4 divisores(3 clases) congela el programa, por lo que no es recomendable usarlo.

### Análisis comparativo

Criterio	Enfoque por color y umbralización simulada	Enfoque Complejo: Preprocesamiento + Multi-Otsu
Cumple con los criterios descritos		
Robustez en la iluminación	Baja	Alta(teórico)
Etiquetas	Por color, específico ["Hoja","Regla","Marcador"]	Generico("Objeto + número")
Velocidad	Alta	Media
Complejidad	Baja (Solo máscaras)	Alta (Punto crítico de uso)
Adaptabilidad	Requiere ajuste manual o automatización	Requiere ajuste de umbrales y color de fondo.

### Resultados Obtenidos

Enfoque HSV: Los bounding boxes aparecen correctamente delimitando los objetos cuando la iluminación es uniforme y los colores están dentro de los rangos predefinidos. En condiciones de sombra, la detección puede falla.

Enfoque avanzado: Los objetos se detectan de forma consistente incluso con sombras suaves o cambios de brillo. Los bounding boxes engloban completamente los objetos, aunque sin distinción de tipo, solamente mostrando una capa de objetos.

### Conclusiones

Segmentación por color y umbralización simulada:

Logra una detección rápida y con etiquetado específico de cada objeto (hoja, regla, marcador) cuando las condiciones de iluminación son estables y los rangos de color están bien calibrados.



Sin embargo, su rendimiento se degrada notablemente ante variaciones de iluminación, sombras o cambios sutiles en el tono de los objetos, lo que limita su aplicabilidad en entornos no controlados.

Enfoque Complejo: Preprocesamiento + Multi-Otsu

Incorpora etapas de eliminación de sombras, realce de color y sharpen, lo que lo hace altamente robusto frente a condiciones de iluminación adversas.

La combinación de máscaras (fondo por color, multi-Otsu y gradiente morfológico) permite segmentar los objetos de forma fiable, generando bounding boxes que los delimitan correctamente.

La principal limitación es la falta de discriminación entre tipos de objeto (todos se etiquetan como "Objeto + número"), aunque la máscara resultante es de gran calidad y puede servir como base para una clasificación posterior, los objetos solamente se colocan en una sola capa.

