



## **PROCESAMIENTO DE IMAGEN Y SEÑALES**

### **INFORME TÉCNICO EJERCICIO PRÁCTICO CLASE 3**

#### **CRITERIOS DE FILTROS, SELECCIÓN DE BORDES Y MORFOLOGÍA MATEMÁTICA.**

INTEGRANTES GRUPO 5:

CESAR NELSON ABARCA ARAUJO

JHONNY NICOLAS RAMIREZ BARAHONA

TEÓFILO MANUEL CHÓEZ ARTEAGA

## **1. Introducción**

El presente trabajo aborda el problema de la segmentación automática de áreas de interés, específicamente la zona de soldadura, en imágenes infrarrojas (IR) en formato .tif. Las imágenes térmicas presentan desafíos inherentes como el ruido del sensor y variaciones de intensidad no lineales, lo que dificulta la extracción de características mediante métodos simples. El objetivo principal no fue estético, sino la construcción de un criterio sólido de segmentación que permitiera aislar la información relevante (zona caliente) del fondo térmico y el ruido, preparando los datos para etapas posteriores en la utilización para modelado.

## 2. Metodología

Se implementó un flujo de trabajo en Python utilizando las librerías OpenCV y NumPy. El conjunto de datos se dividió en entrenamiento (train) y prueba (test), asegurando que los parámetros se ajustaran únicamente con el primer grupo. Se diseñaron y evaluaron cuatro enfoques técnicos distintos:

- ***Técnica 1: Umbralización Adaptativa y Morfología:*** Se utilizó el método de Otsu para calcular automáticamente el umbral óptimo de binarización, seguido de operaciones morfológicas de apertura (opening) para eliminar ruido puntual y cierre (closing) para rellenar huecos dentro de la región de interés.
- ***Técnica 2: Detección de Bordes y Morfología:*** Se aplicó un suavizado Gaussiano seguido del detector de bordes Canny. Posteriormente, se intentó conectar los bordes fragmentados mediante dilatación y rellenar los contornos cerrados.
- ***Técnica 3: Combinación Secuencial:*** Un enfoque más complejo que integró filtro bilateral para preservar bordes mientras se reduce el ruido, realce de contraste mediante CLAHE (Contrast Limited Adaptive Histogram Equalization), umbralización adaptativa y selección del componente conectado más grande.
- ***Técnica 4: Algoritmo Watershed:*** Se implementó una segmentación basada en cuencas (Watershed), utilizando la transformada de distancia para definir el "fondo seguro" (sure background) y el "primer plano seguro" (sure foreground) antes de aplicar el algoritmo de inundación

### 3. Comparativa Visual

Durante la fase de experimentación con las imágenes de muestra, se observaron diferencias significativas en la calidad de las máscaras generadas:

- La Técnica 1 logró aislar la zona de mayor intensidad térmica de manera consistente, generando una máscara limpia y coherente con la forma esperada de la soldadura.
- La Técnica 2 falló en la generación de la máscara, produciendo una imagen vacía. Esto sugiere que el ruido térmico fragmentó los bordes hasta el punto en que las operaciones morfológicas no pudieron reconstruir un contorno cerrado.
- La Técnica 3, a pesar de su complejidad, resultó ser demasiado restrictiva, detectando áreas minúsculas que no representaban la totalidad de la zona de interés.
- La Técnica 4 (Watershed) ofreció una segmentación razonable, muy cercana a la Técnica 1, aunque con una cobertura ligeramente menor

### 4. Estadísticas Cuantitativas

- El análisis cuantitativo sobre la imagen de muestra arrojó los siguientes resultados de cobertura (píxeles segmentados respecto al total de la imagen):
  - Técnica 1 (Umbral + Morfología): 9,851 píxeles (3.21% de la imagen).
  - Técnica 2 (Bordes): 0 píxeles (0.00% de la imagen).
  - Técnica 3 (Secuencial): 76 píxeles (0.02% de la imagen).
  - Técnica 4 (Watershed): 7,380 píxeles (2.40% de la imagen).
- Estos datos confirmaron que la Técnica 1 ofrecía la mayor recuperación de información relevante sin introducir excesivo ruido de fondo.

## 5. Resultados Obtenidos

Basado en el balance entre precisión, estabilidad y costo computacional, se seleccionó la **Técnica 1 (Umbralización + Morfología)** como la metodología recomendada. Este método demostró ser eficiente para escenarios donde el área de interés presenta un contraste térmico diferenciado, permitiendo un procesamiento rápido ideal para aplicaciones en tiempo real. La máscara resultante se aplicó a las imágenes originales mediante una operación *bitwise AND*, asignando valor cero a todo píxel fuera del área relevante

## 6. Validación Visual del Test Set

La técnica seleccionada se aplicó al conjunto de prueba (70 imágenes) para validar su generalización. El procesamiento fue exitoso, guardándose las imágenes resultantes en el directorio de resultados. Las estadísticas del conjunto de prueba mostraron una consistencia en la detección, con áreas segmentadas que oscilaron generalmente entre el 0.38% y el 2.23% del área total de las imágenes, lo que indica que el método se adaptó adecuadamente a las variaciones dentro del dataset sin fallos catastróficos (como máscaras vacías o completas)

## 7. Conclusiones

El estudio concluye que, para este conjunto de datos de termografía de soldadura, la complejidad no garantizó mejores resultados. Mientras que enfoques avanzados como la detección de bordes o el filtrado secuencial agresivo fallaron debido a la naturaleza difusa de los bordes térmicos o la pérdida de información sutil, el método de **Umbralización de Otsu combinado con limpieza morfológica** resultó ser la solución más robusta. Esta técnica logró estabilizar la información y reducir efectivamente el ruido del sensor, cumpliendo con el objetivo de preparar los datos para análisis posteriores al reducir la dimensionalidad irrelevante de la imagen.

## 8. Referencias al Código

Las implementaciones clave utilizadas en este informe se encuentran en las siguientes secciones del código fuente:

- **Carga y Preprocesamiento:** Uso de `cv2.imread` y `os.listdir` para gestión de datos.
- **Técnica 1 (Seleccionada):** Implementación de `cv2.threshold` con `cv2.THRESH_OTSU` y `cv2.morphologyEx`.
- **Técnica 2:** Uso de `cv2.Canny` y `cv2.findContours`.
- **Técnica 3:** Aplicación de `cv2.bilateralFilter` y `cv2.createCLAHE`.
- **Técnica 4:** Implementación de `cv2.distanceTransform` y `cv2.watershed`.
- **Validación:** Bucle de procesamiento sobre `TEST_DIR` y cálculo de estadísticas de área