

**UNIVERSIDAD  
DE ANTIQQUIA**  
1803



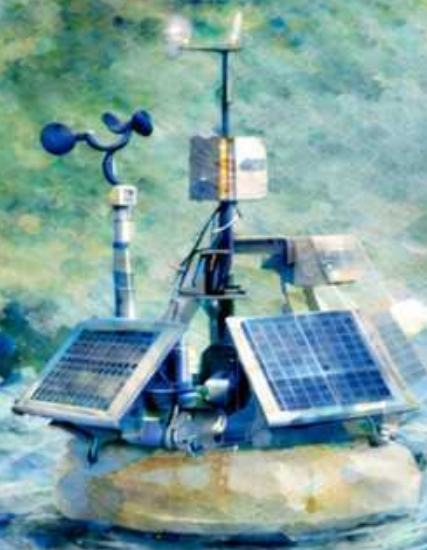
**GeoLimna**



# *Seminario GeoLimna–GEPAR*

*— Volumen 1 —*

*Investigaciones 2019–2021*



**Coordinación**

**Juan Pablo Serna | David Fernández**  
Coordinadores de Geolimna y GEPAR

**Colección Seminario GeoLimna–GEPAR**

**Volumen 1. Investigaciones 2019–2021**

© 2026 Universidad de Antioquia

© 2026 Facultad de Ingeniería, Universidad de Antioquia

© 2026 Grupos de investigación GeoLimna y GEPAR

**ISBN Digital:** *(por asignar)*

Queda hecho el depósito que ordena la Ley

(Ley 44 de 1993 / Decreto 460 de 1995)

**Dirección editorial:** Juan Pablo Serna y David Fernández

**Dirección de colección:** *Juan Pablo Serna y David Fernández*

**Diseño de cubierta y diagramación:** *Juan Pablo Serna y David Fernández*

**Universidad de Antioquia – Facultad de Ingeniería**

Calle 67 # 53–108

Medellín, Colombia

Código Postal: 050010

Correo electrónico: *jpablo.serna@udea.edu.co; david.fernandez@udea.edu.co*

Sitio web: *https://tinyurl.com/mvmcnxad*

No está permitida la reproducción total o parcial de este libro ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito de los titulares de derechos.

# Contenido

Prefacio . . . . .	III
1 Determinación automática de macroinvertebrados con IA . . . . .	1
2 Máquina seleccionadora de esquejes: arquitectura, modelo y operación . . . . .	12
3 Conteo automático de colonias virales en pozos de agar . . . . .	23
4 Inventario automático de áreas de líquenes cortícolas . . . . .	33
5 Extracción automática de características geométricas de microperforaciones en placas GEM . . . . .	46
6 Estabilización digital de videos de <i>Daphnia pulex</i> para análisis fisiológico <i>in vivo</i> . . . . .	57
7 Estimación de la frecuencia cardiaca de <i>Daphnia pulex</i> mediante fotocardiograma . . . . .	71
8 Sistema inalámbrico para la estimación de la velocidad de flujo de agua . . . . .	83
9 SICOP Panorámica: construcción de una imagen panorámica de camas de plántulas . . . . .	96
10 SICOP Analítica: Conteo y caracterización morfométrica de plántulas sembradas . . . . .	108
11 Conteo automático y análisis morfométrico de <i>Hyalella</i> mediante visión artificial	121
12 Sistema de cuantificación y aforo de nubes a partir de imágenes de cielo . . . . .	134
13 Aplicación móvil para medición de variables ambientales en ecosistemas acuáticos . . . . .	145
14 Identificación automática de irregularidades en pavimento . . . . .	156

# Comité Académico

Facultad de Ingeniería - Universidad de Antioquia

## Coordinadores

### Juan Pablo Serna López

Profesor asociado, Departamento de Ingeniería Ambiental y Sanitaria.

`jpablo.serna@udea.edu.co`

### David Fernández Mc Cann

Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones.

`david.fernandez@udea.edu.co`

## Integrantes

### Néstor Jaime Aguirre Ramírez

Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria.

`nestor.aguirre@udea.edu.co`

### Fabio de Jesús Vélez Macías

Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. `fabio.velez@udea.edu.co`

### Elkin Libardo Ríos Ortiz

Profesor titular, Departamento de Ingeniería Industrial. `elkin.rios@udea.edu.co`

### Juan Pablo Urrea Duque

Profesor asociado, Departamento de Ingeniería Electrónica y Telecomunicaciones.

`juan.urrea@udea.edu.co`

### Yarin Tatiana Puerta Quintana

Profesora asistente, Departamento de Ingeniería Ambiental y Sanitaria.

`tatiana.puerta@udea.edu.co`

# Prefacio

Este libro es el resultado de una continuidad silenciosa: la de un seminario que, desde 2019, ha servido como punto de encuentro entre dos maneras de mirar el mundo que no compiten, sino que se completan. Por un lado, el interés ambiental, ecológico y territorial que impulsa al grupo Geolimna; por el otro, la capacidad técnica, instrumental y metodológica del grupo GEPAR para convertir preguntas complejas en sistemas medibles, algoritmos verificables y herramientas utilizables. En ese cruce —que en la práctica se vuelve conversación, prototipo, prueba de campo, iteración y mejora— se fue formando una colección de desarrollos que merecía un formato más estable que la presentación oral y el video: una memoria estructurada.

El seminario ha funcionado como un laboratorio de síntesis. Cada exposición recoge una necesidad real, la traduce en un problema abordable, y obliga a definir un camino técnico con criterios de rigurosidad. A veces el punto de partida fue una inquietud ambiental: ¿cómo medir y entender un sistema acuático? ¿cómo observar organismos bioindicadores con menos dependencia de la mirada experta? ¿cómo registrar señales débiles y variables en ambientes cambiantes? Otras veces, el origen fue una tecnología disponible —sensores, cámaras, comunicaciones, visión por computador— buscando un problema donde demostrar impacto. El valor del seminario ha sido obligar a que ambos impulsos se encuentren: que el propósito ambiental no se quede en discurso y que la solución técnica no sea un artefacto aislado.

Por eso, este libro no se plantea como una recopilación ornamental. Cada capítulo está escrito como un reporte técnico de investigación y desarrollo: define un problema, formaliza un método, presenta resultados y discute alcances y limitaciones. Su fuente primaria son las presentaciones del seminario y sus videos, complementadas con texto adicional, manuales, registros y publicaciones asociadas. Se privilegia un lenguaje que permita trazabilidad: que el lector pueda reconstruir la lógica del desarrollo y, si lo desea, replicar, mejorar o adaptar el trabajo.

Conviene subrayar, sin embargo, la naturaleza de origen de estos textos: los capítulos son, ante todo, la cristalización escrita de exposiciones. En varios casos no constituyen

---

un “reporte original” en el sentido estricto de la publicación científica especializada; por el contrario, representan una etapa intermedia o paralela del proceso de investigación, donde el desarrollo se expone, se discute, se valida y se documenta. Por ello, algunos trabajos aquí consignados pueden haber derivado posteriormente en artículos, ponencias o reportes formales en ámbitos disciplinares específicos. Esta compilación no compite con esos productos; los complementa. Su valor reside en preservar el hilo técnico completo que suele quedar fragmentado: la motivación, la decisión metodológica, el prototipo, la implementación, el ajuste iterativo y el aprendizaje colectivo que ocurre en el seminario.

En el trasfondo hay una idea que atraviesa todos los capítulos: lo ambiental necesita ingeniería, y la ingeniería necesita un horizonte ambiental. Medir calidad de agua, reconocer organismos, automatizar procesos de monitoreo, diseñar instrumentación o construir herramientas de análisis no es solo “aplicar tecnología”; es desarrollar capacidades concretas para sostener decisiones informadas sobre sistemas vivos. Esta perspectiva no romantiza el dato: lo sitúa. Cada variable medida, cada imagen procesada, cada clasificador entrenado y cada interfaz desplegada existe porque hay un fenómeno ambiental que exige atención, y porque hay una comunidad que necesita herramientas para entenderlo con mayor precisión y menor fricción.

La colaboración Geolimna–GEPAR ha demostrado que la interdisciplina no es una etiqueta, sino una práctica. Se construye cuando se acuerdan protocolos de captura, cuando se discuten criterios de calidad, cuando se aceptan iteraciones, cuando se reconoce la incertidumbre y se diseñan mecanismos para reducirla. Se construye también cuando el resultado final no es solo un informe, sino un software usable, un manual técnico, un registro, un método y un conjunto de evidencias que resisten la revisión.

Finalmente, este libro pretende cumplir una función doble. Hacia adentro, es memoria y plataforma: organiza lo logrado entre 2019 y 2021 y facilita que nuevas cohortes no comiencen desde cero. Hacia afuera, es invitación: muestra que es posible convertir problemas ambientales en desarrollos técnicos con impacto, sin perder rigor ni sensibilidad por el contexto. Si el lector encuentra aquí ideas transferibles, metodologías replicables o, simplemente, una manera más clara de narrar el vínculo entre ambiente e ingeniería, entonces esta compilación habrá cumplido su propósito.

*Grupos Geolimna y GEPAR*  
Universidad de Antioquia  
Volumen 1, Investigaciones 2019-2021

# Capítulo 1

## Determinación automática de macroinvertebrados con IA

### Autores

Juan Pablo Serna López<sup>1</sup>; Néstor Jaime Aguirre Ramírez<sup>2</sup>; Fabio Vélez Macías<sup>3</sup>; David Fernández Mc Cann<sup>4</sup>; Juan Sebastián Lobo Ramírez<sup>5</sup>.

Grupos Geolimna y Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

Los macroinvertebrados acuáticos (MA) son bioindicadores ampliamente utilizados para caracterizar la calidad y estabilidad ecológica de sistemas lóticos. Sin embargo, los flujos de trabajo tradicionales para su determinación demandan alta experticia taxonómica, tiempo de laboratorio y, en general, implican colecta y preservación con sacrificio de organismos. En este capítulo se presenta, con formulación formal, el desarrollo *MACRODEX*: un aplicativo computacional soportado en MATLAB que reconoce e identifica MA a partir de fotografías digitales de laboratorio para ríos del departamento de Antioquia (quebrada La Nitrera, ríos San Juan y Cauca) [1], [2], [3]. El sistema implementa un pipeline

---

<sup>1</sup>Profesor Asociado, Departamento de Ingeniería Ambiental y Sanitaria, jpablo.serna@udea.edu.co

<sup>2</sup>Profesor Titular, Departamento de Ingeniería Ambiental y Sanitaria, nestor.aguirre@udea.edu.co

<sup>3</sup>Profesor Titular, Departamento de Ingeniería Ambiental y Sanitaria, fabio.velez@udea.edu.co

<sup>4</sup>Profesor Titular, Departamento de Ingeniería Electrónica y Telecomunicaciones, david.fernandez@udea.edu.co

<sup>5</sup>Estudiante, Ingeniería de Sistemas, juan.lobo@udea.edu.co

de captura estandarizada, preprocesamiento para aislar el organismo, extracción de características locales, cuantización mediante  $k$ -means, representación *Bag of Visual Words* (BoVW) y clasificación multiclase con máquinas de soporte vectorial (SVM). El modelo fue entrenado para cuatro géneros (*Thraulodes*, *Traverella*, *Anacroneuria* y *Smicridea*) [3], [4] y alcanza un porcentaje de acierto reportado de 97.1% en la mejor configuración [4]. Se presenta además la formulación de métricas, matriz de confusión y consideraciones para escalamiento.

**Palabras clave:** macroinvertebrados acuáticos; biomonitorio; procesamiento digital de imágenes; Bag of Visual Words; SURF; SVM; MATLAB App Designer; MACRODEX.

### 1.1. Introducción

La comunidad de macroinvertebrados acuáticos refleja el estado ecológico de un río y ha sido utilizada como proxy de calidad ambiental en ecosistemas lóticos [3], [5]. En Antioquia, perturbaciones naturales y antrópicas alteran la estructura y ensamblaje de dichas comunidades, motivando estrategias de monitoreo más eficientes [3], [5].

Desde la ingeniería, el reto operacional se centra en transformar evidencia biológica (organismos preservados y observados en laboratorio) en información confiable y repetible, con menor dependencia de expertos. El desarrollo MACRODEX se enmarca en este reto: proveer un mecanismo computacional que, bajo condiciones de captura controladas, determine el género del organismo a partir de su morfología en imágenes [1], [2], [4].

Además del aporte técnico, este enfoque tiene una implicación ambiental directa: al trasladar parte del proceso de determinación hacia evidencia visual estandarizada, se abre la posibilidad de reducir la colecta destructiva asociada al biomonitorio tradicional. En la práctica clásica, la identificación suele requerir captura, preservación y observación detallada en laboratorio; ello implica impacto sobre la comunidad muestreada y, a escala de campañas repetidas, una presión no despreciable sobre poblaciones locales. En cambio, un método computacional como MACRODEX sugiere una ruta de *monitoreo menos invasivo*: cuando la adquisición de imágenes puede realizarse con protocolos reproducibles (y, en escenarios futuros, incluso *in situ*), la determinación puede apoyarse en el análisis digital sin depender necesariamente del sacrificio sistemático de organismos. Esta transición no elimina la necesidad de validación taxonómica experta, pero sí permite que la evidencia se capture, conserve y revise sin incrementar el costo ecológico del muestreo, alineando la instrumentación y la IA con un principio de mínima perturbación del ambiente.



Figura 1.1: Ejemplos de macroinvertebrados acuáticos utilizados como clases de reconocimiento en MACRODEX. De izquierda a derecha: *Thraulodes*, *Traverella*, *Smicridea* y *Anacroneuria*. Estas imágenes ilustran la variabilidad morfológica que el sistema debe capturar para discriminar géneros a partir de evidencia visual de laboratorio.

## 1.2. Planteamiento del problema (formulación matemática)

### 1.2.1. Problema de determinación como decisión estadística

Sea  $\Omega = \{\omega_1, \dots, \omega_K\}$  el conjunto de clases taxonómicas (géneros) modeladas. En MACRODEX,  $K = 4$  y  $\Omega = \{\text{Thraulodes}, \text{Traverella}, \text{Anacroneuria}, \text{Smicridea}\}$  [3], [4] (ver ejemplos en la Fig. 1.1).

Sea  $\mathcal{I}$  el espacio de imágenes y  $x \in \mathcal{I}$  una fotografía digital capturada bajo un protocolo de laboratorio (microscopio estereoscópico + cámara) [1], [2]. El problema consiste en diseñar una regla de decisión

$$\delta : \mathcal{I} \rightarrow \Omega$$

tal que minimice el riesgo esperado de clasificación bajo una función de pérdida  $\ell(\cdot, \cdot)$ :

$$\delta^* \in \arg \min_{\delta} \mathbb{E}[\ell(Y, \delta(X))],$$

donde  $(X, Y)$  es una variable aleatoria con  $Y \in \Omega$ .

### 1.2.2. Modelo de imagen y segmentación del objeto de interés

Una imagen digital RGB se representa como una función discreta

$$f : \mathbb{N}^2 \rightarrow \mathbb{N}^3, \quad f(x, y) = (r, g, b), \quad (r, g, b) \in [0, 255]^3,$$

con  $1 \leq x \leq n$ ,  $1 \leq y \leq m$  [4].

Sea  $S$  un operador de segmentación que asigna etiquetas a regiones conectadas:

$$S\{f\} \Rightarrow \{(\text{Obj}_i, e_i)\}_{i=1}^{N_o},$$

donde  $\text{Obj}_i$  es una región (objeto) y  $e_i$  su etiqueta. La regla operativa del pipeline consiste en conservar el objeto de mayor área (organismo) y eliminar el resto:

$$i^* = \arg \max_i \text{Area}(\text{Obj}_i), \quad f' = \mathcal{P}(f) = \text{keep}(\text{Obj}_{i^*}),$$

obteniendo una imagen depurada  $f'$  donde solo permanece el macroinvertebrado, sobre fondo homogenizado (negro) [3], [4].

### 1.2.3. Representación discriminativa

El objetivo es construir un mapeo

$$\phi : \mathcal{I} \rightarrow \mathbb{R}^d$$

tal que un clasificador  $D : \mathbb{R}^d \rightarrow \Omega$  determine la clase:

$$\hat{y} = \delta(x) = D(\phi(x)).$$

En el método documentado,  $\phi$  se obtiene mediante BoVW a partir de descriptores locales tipo SURF y cuantización por  $k$ -means [4].

## 1.3. Metodología (formulación formal y pipeline)

### 1.3.1. Protocolo de captura

La calidad del modelo depende de estandarizar el proceso de captura, incluyendo resolución, encuadre y enfoque. En la documentación técnica se especifica un procedimiento de captura digital con estereomicroscopio/cámara, ubicación del organismo en caja de Petri

con alcohol al 70 %, centrado, aumento aproximado 80 % y enfoque macrométrico antes de capturar y etiquetar la imagen [1], [2]. Un ejemplo del montaje de captura empleado se muestra en la Fig. 1.2.



Figura 1.2: Montaje de captura de imágenes en laboratorio: microscopio Zeiss y soporte para smartphone (iPhone 12) para adquisición fotográfica estandarizada. La estabilización mecánica y la repetibilidad del encuadre reducen variabilidad intra-clase no deseada (iluminación, escala y enfoque), lo que favorece la generalización del clasificador.

### 1.3.2. Base de datos y partición

Sea  $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$  la base de datos etiquetada. Se reporta una base de  $N = 587$  imágenes para cuatro clases y una partición típica 70/30 para entrenamiento/prueba [3], [4].

Para evitar desbalance, se aplican técnicas de aumento hasta aproximar 500 muestras por clase (según se documenta en el método) [4].

### 1.3.3. Extracción de características locales y cuantización

Dada una imagen depurada  $f'$ , se extrae un conjunto de descriptores locales

$$\mathcal{Z} = \{z_j\}_{j=1}^M, \quad z_j \in \mathbb{R}^p,$$

mediante SURF [4].

Se aprende un diccionario visual (*codebook*) de tamaño  $V$  resolviendo  $k$ -means:

$$\{\mu_v\}_{v=1}^V \in \arg \min_{\{\mu_v\}} \sum_{j=1}^M \min_v \|z_j - \mu_v\|_2^2.$$

En el desarrollo se reporta una representación tipo “palabras” con longitud  $V = 500$  por imagen (tamaño del histograma/feature vector) [4].

Cada descriptor se cuantiza por vecino más cercano:

$$q(z_j) = \arg \min_v \|z_j - \mu_v\|_2.$$

### 1.3.4. Bag of Visual Words (histograma) como vector de características

El vector de características  $\chi \in \mathbb{R}^V$  se define como el histograma de ocurrencias:

$$\chi_v = \sum_{j=1}^M \mathbf{1}\{q(z_j) = v\}, \quad v = 1, \dots, V.$$

Este histograma se interpreta como la “huella” de la clase, y su longitud corresponde al número de palabras visuales [3], [4].

Opcionalmente, puede normalizarse:

$$\tilde{\chi} = \frac{\chi}{\|\chi\|_1}.$$

### 1.3.5. Clasificador SVM multiclase

Sea  $\tilde{\chi}_i$  el descriptor BoVW de  $x_i$ . El objetivo del clasificador es aprender una función discriminante. En el sistema se adopta SVM como clasificador de mejor rendimiento [4].

Para el caso binario, la SVM resuelve:

$$\min_{w,b,\xi} \frac{1}{2} \|w\|_2^2 + C \sum_i \xi_i \quad \text{s.a.} \quad y_i(w^\top \tilde{\chi}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

Para multiclase se emplea una descomposición One-vs-One u One-vs-Rest (implementación estándar en MATLAB) [1]. El resultado operacional del sistema produce *scores* por clase para soportar la identificación [1].

### 1.3.6. Implementación software (MACRODEX)

MACRODEX fue implementado modularmente en MATLAB App Designer, con dos funcionalidades: identificación por individuo y por grupo (lote) [6] [1], [2]. A nivel funcional, se documenta que el flujo central invoca un procesamiento que carga modelo, decodifica imagen, clasifica con SVM, obtiene *scores* y retorna identificación [1].

## 1.4. Resultados (formulación y evidencias)

### 1.4.1. Exactitud global reportada

En la mejor configuración, se reporta un porcentaje de acierto de 97.1 % [4].

### 1.4.2. Matriz de confusión y métricas derivadas

Sea  $C \in \mathbb{N}^{K \times K}$  la matriz de confusión, donde  $C_{ij}$  es el número de ejemplos de clase real  $\omega_i$  predichos como  $\omega_j$ . La exactitud global es

$$\text{Acc} = \frac{\sum_{k=1}^K C_{kk}}{\sum_{i=1}^K \sum_{j=1}^K C_{ij}}.$$

Para cada clase  $k$ :

$$\text{Prec}_k = \frac{C_{kk}}{\sum_j C_{jk}}, \quad \text{Rec}_k = \frac{C_{kk}}{\sum_j C_{kj}}, \quad \text{F1}_k = \frac{2 \text{Prec}_k \text{Rec}_k}{\text{Prec}_k + \text{Rec}_k}.$$

En la tesis/artículo se reportan aciertos por clase (ejemplo): para Thraulodes, 81 de 86; para Traverella, 167 de 168; para Anacroneuria, 50 de 51; y para Smicridea, 81 de 84

(según matriz de confusión reportada) [3], [4].

Tabla 1.1: Resumen de aciertos por clase (según reporte) [3], [4].

<b>Clase</b>	<b>Aciertos</b>	<b>Total prueba</b>
Thraulodes	81	86
Traverella	167	168
Anacroneuria	50	51
Smicridea	81	84

### 1.4.3. Distribución de la base de datos

La distribución inicial de imágenes por clase (y la ampliación) se reporta para un total de 587 imágenes, con predominio de Traverella [3], [4].

Tabla 1.2: Distribución de imágenes por clase reportada [3], [4].

<b>Género</b>	<b>Orden</b>	<b>Familia</b>	<b># inicial</b>
Thraulodes	Ephemeroptera	Leptophlebiidae	124
Traverella	Ephemeroptera	Leptophlebiidae	240
Anacroneuria	Plecoptera	Perlidae	73
Smicridea	Trichoptera	Hydropsychidae	150

### 1.4.4. Interfaz gráfica (evidencia de despliegue)

Se documenta una interfaz en App Designer que permite determinación individual y por lote [1], [2]. Un ejemplo de salida hacia el usuario final se muestra en la Fig. 1.3.

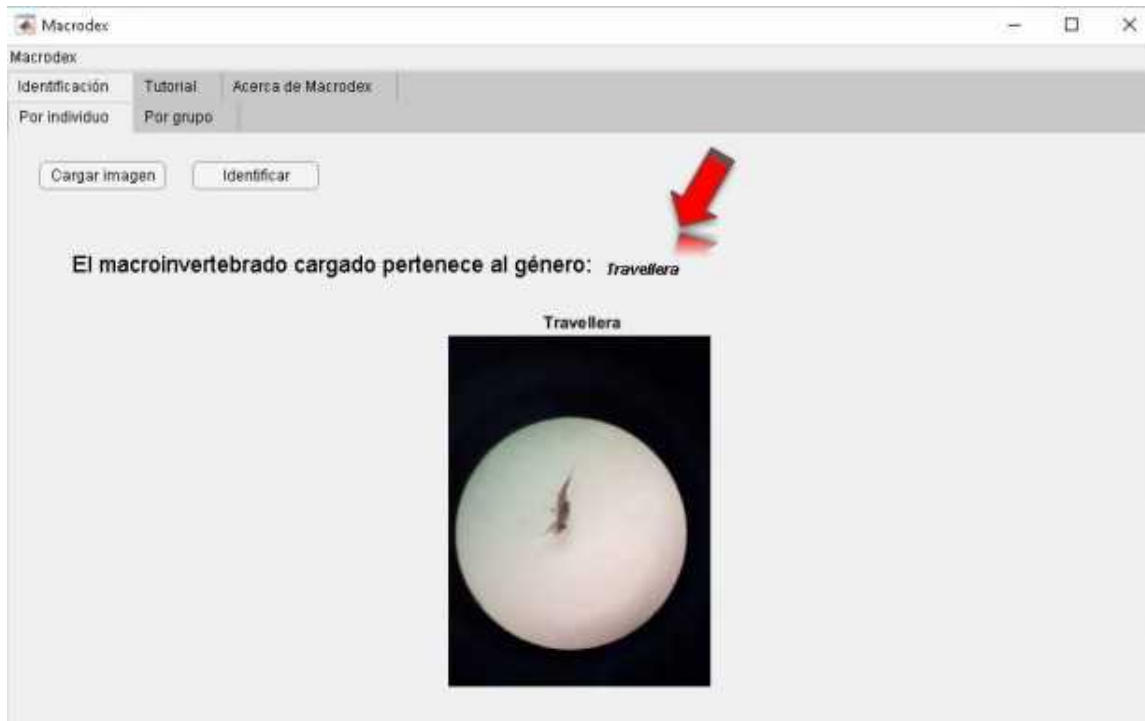


Figura 1.3: Interfaz gráfica de MACRODEX (MATLAB App Designer). La aplicación carga una imagen adquirida en el montaje de laboratorio, ejecuta el pipeline de preprocesamiento y clasificación, y presenta al usuario el género estimado (ej., *Traverella*) junto con la imagen analizada.

## 1.5. Análisis de resultados (formal)

### 1.5.1. Estabilidad del descriptor y separabilidad

La tubería BoVW induce un mapeo  $\phi(x) = \tilde{\chi}$  que convierte la imagen a un espacio discreto de frecuencias. La hipótesis operacional es que, bajo condiciones de captura controladas, las morfologías de los géneros inducen distribuciones suficientemente separables en  $\mathbb{R}^V$  para que una frontera SVM logre bajo error empírico [4].

### 1.5.2. Efecto del protocolo de captura

El sistema depende críticamente de un método de captura digital estandarizado [1], [2]. En términos de modelamiento, esto reduce la varianza intra-clase no deseada (cambios de iluminación, escala y enfoque), acercando el dominio de prueba al de entrenamiento y mejorando la generalización [4].

### 1.5.3. Interpretación de la matriz de confusión

La presencia de altas tasas de acierto por clase (p. ej., 167/168 en Traverella) sugiere que para ciertos morfotipos la señal morfológica capturada por SURF y cuantizada en BoVW es altamente consistente [3], [4]. Las discrepancias residuales (p. ej., errores en Thraulodes) pueden asociarse a variaciones de pose/oclusiones, a fallos del preprocesamiento  $\mathcal{P}$  al aislar el objeto mayor, o a similitud morfológica parcial entre clases en escalas particulares de captura [3].

### 1.5.4. Escalamiento: de cuatro géneros a inventarios ampliados

Si  $\Omega$  crece, la complejidad de decisión aumenta y la representación BoVW puede requerir: (i) mayor tamaño del diccionario  $V$ , (ii) datos adicionales por clase para estimación robusta y (iii) estrategias de calibración de *scores* [4]. El manual técnico contempla identificación por lote y exportación de resultados a `xlsx`, lo cual es coherente con escenarios de inventario y biomonitoreo [1].

## 1.6. Conclusiones

Se presentó el capítulo formal del desarrollo MACRODEX, incluyendo (i) formulación del problema como decisión estadística  $\delta : \mathcal{I} \rightarrow \Omega$ , (ii) pipeline matemático de segmentación  $\mathcal{P}$ , representación BoVW  $\phi(x) = \tilde{\chi}$ , y clasificación SVM  $D(\cdot)$ , (iii) evidencia de implementación en MATLAB App Designer con modos individual y por lote, y (iv) resultados reportados con porcentaje de acierto 97.1% y matriz de confusión por clase [1], [3], [4]. Como trabajo futuro, el escalamiento a más géneros/clases requiere expansión de base de datos, control estricto de variabilidad de captura y evaluación sistemática de modos de fallo [3].

## Agradecimientos

A los Grupos de Investigación Geolimna y GEPAR por el soporte académico e interdisciplinar que permite articular problemas de biomonitoreo con soluciones de biomonitoreo, visión artificial y automatización.

## Referencias

- [1] *Manual Técnico – Aplicación MACRODEX*, Documento técnico interno, Grupo GEPAR y GeoLimna, 2020.
- [2] *MACRODEX – Manual de Usuario*, Grupo GEPAR y GeoLimna, 2020.
- [3] J. P. Serna López, “Análisis de la estabilidad ecológica de un reoambiente a lo largo de un gradiente altitudinal a través de los macroinvertebrados acuáticos y el procesamiento digital de imágenes,” Tesis doctoral, Universidad de Antioquia, Medellín, Colombia, 2019.
- [4] J. P. Serna López, D. S. Fernández Mc Cann, F. d. J. Vélez Macías y N. J. Aguirre Ramírez, “An Image Processing Method for Recognition of Four Aquatic Macroinvertebrates Genera in Freshwater Environments in the Andean Region of Colombia,” *Environmental Monitoring and Assessment*, vol. 192, n.º 617, 2020. DOI: 10.1007/s10661-020-08545-2.
- [5] J. P. Serna, F. Vélez, D. Fernández y N. Aguirre, “Análisis de la estabilidad ecológica de tres reoambientes a través de los macroinvertebrados acuáticos y el procesamiento digital de imágenes,” en *AQUATROP: Ecosistemas acuáticos tropicales en el Antropoceno (Resumen de congreso)*, Resumen de congreso, Quito, Ecuador, 2018.
- [6] D. Fernández Mc Cann, N. J. Aguirre Ramírez, F. Vélez Macías, J. P. Serna López, J. S. Lobo Ramírez y D. Uribe Giraldo, *MACRODEX: Certificado de Registro de Soporte Lógico (Software), Libro–Tomo–Partida 13-73-482*, Dirección Nacional de Derecho de Autor (DNDA), Ministerio del Interior, Colombia, Fecha de registro: 22-may-2019. Titular del derecho patrimonial: Universidad de Antioquia, mayo de 2019.

# Capítulo 2

## Máquina seleccionadora de esquejes: arquitectura, modelo y operación

### Autores

Sebastián Guzmán Obando<sup>1</sup>; Santiago Ruiz González<sup>2</sup>; Maycol Esteban Zuluaga Montoya<sup>3</sup>; David Fernández Mc Cann<sup>4</sup>.

Grupos Geolimna y Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

La selección de esquejes en floricultura es una operación crítica por su impacto directo en la uniformidad del material vegetativo, el rendimiento del cultivo y la estabilidad del proceso de enraizamiento. Tradicionalmente, esta selección se realiza de forma manual, demandando tiempo, atención sostenida y alta repetibilidad bajo condiciones de producción. En este capítulo se presenta, con formulación matemática y lectura mecatrónica, una máquina seleccionadora de esquejes soportada en visión artificial y actuación neumática sincronizada, junto con su implementación software asociada (SOCAES) [1], [2], [3].

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. [sebastian.guzman@udea.edu.co](mailto:sebastian.guzman@udea.edu.co)

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. [santiago.ruiz@udea.edu.co](mailto:santiago.ruiz@udea.edu.co)

<sup>3</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. [maycol.zuluaga@udea.edu.co](mailto:maycol.zuluaga@udea.edu.co)

<sup>4</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. [david.fernandez@udea.edu.co](mailto:david.fernandez@udea.edu.co)

El sistema integra una banda transportadora, adquisición de imágenes bajo un protocolo operativo, estimación morfológica (longitud y condiciones basales), una regla de decisión por umbrales y un mecanismo de retardo discreto que sincroniza la decisión computacional con la expulsión neumática. Se discuten además límites de observabilidad derivados de la proyección 2D, métricas estándar de desempeño (precisión, *recall*) y aprendizajes de consolidación en campo (interferencias, presión/boquilla, guiado y tolvas). Finalmente, se contextualiza el desarrollo frente a sistemas industriales comparables y el marco institucional de registro de software ante la DNDA [4], [5], [6], [7].

**Palabras clave:** floricultura; selección de esquejes; visión artificial; medición morfológica; control discreto; sincronización; eyección neumática; clasificación óptica; SOCAES.

### 2.1. Introducción

La floricultura industrial exige consistencia en el material vegetativo para asegurar lotes homogéneos en enraizamiento y crecimiento. En este contexto, la selección de esquejes se convierte en un cuello de botella cuando depende exclusivamente de inspección manual: la variabilidad humana, la fatiga y la velocidad del flujo productivo incrementan la probabilidad de error y limitan la productividad. Desde la ingeniería, el problema puede leerse como un sistema de decisión y actuación sobre un flujo de objetos deformables en transporte continuo.

El proyecto abordó la automatización parcial/total de la selección mediante visión artificial y neumática, materializado en una máquina seleccionadora y el software SOCAES para adquisición, medición y decisión [1], [2]. El trabajo de grado asocia formaliza la implementación del algoritmo de clasificación y sincronización con el sistema embebido [3].

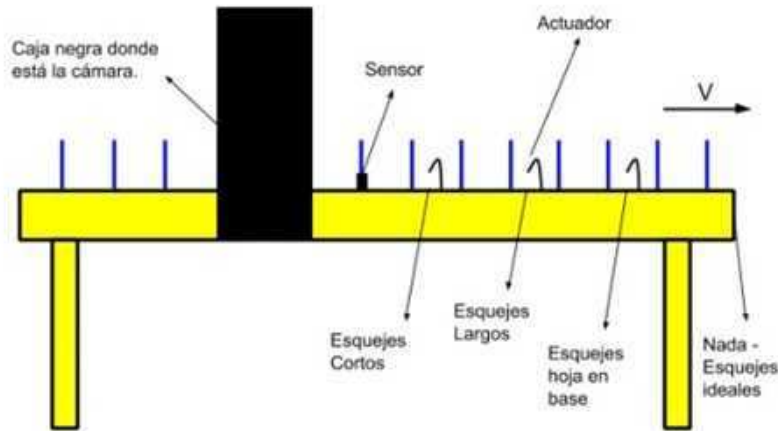


Figura 2.1: Arquitectura general de la máquina seleccionadora: alimentación sobre banda, adquisición con cámara, decisión de clasificación y actuación neumática sincronizada para desviar el esqueje hacia contenedores de salida.

## 2.2. Planteamiento del problema (formulación matemática)

### 2.2.1. Selección como decisión bajo incertidumbre

Sea un flujo de esquejes que transita sobre una banda. Para cada unidad inspeccionada se adquiere una imagen  $x \in \mathcal{I}$  y se requiere asignar una etiqueta de calidad o condición operativa. Definimos el conjunto de clases

$$\Omega = \{\omega_0, \omega_1, \omega_2, \omega_3, \omega_4\},$$

donde  $\omega_0$  representa ausencia de objeto procesable (fondo/lectura nula) y el resto corresponde a categorías operativas tales como *corto*, *largo*, *hoja en base* e *ideal* [1].

El objetivo es construir una regla de decisión

$$\delta : \mathcal{I} \rightarrow \Omega$$

que minimice el riesgo esperado

$$\delta^* \in \arg \min_{\delta} \mathbb{E}[\ell(Y, \delta(X))],$$

donde  $(X, Y)$  es una variable aleatoria con  $Y \in \Omega$  y  $\ell(\cdot, \cdot)$  es una función de pérdida que penaliza errores de selección (por ejemplo, dejar pasar material no conforme como *ideal*) [3].

### 2.2.2. Modelo de imagen, segmentación y medición

Sea la imagen digital (en escala de grises o canal equivalente para medición) definida como

$$I : \Omega_p \subset \mathbb{Z}^2 \rightarrow \{0, \dots, 255\}.$$

Tras preprocesado, se busca una máscara binaria del objeto (esqueje):

$$M(x, y) = \begin{cases} 1, & (x, y) \in \text{esqueje}, \\ 0, & \text{fondo}. \end{cases}$$

Definimos el conjunto de píxeles del esqueje

$$S = \{(x, y) \in \Omega_p : M(x, y) = 1\}.$$

Una estimación simple de longitud en píxeles puede definirse como el diámetro máximo:

$$L_{\text{px}} = \max_{p, q \in S} \|p - q\|_2.$$

Con un factor de calibración  $\alpha$  (cm/píxel), se obtiene la longitud física

$$L = \alpha L_{\text{px}} \quad [\text{cm}],$$

que se compara con umbrales operativos para decidir *corto* vs *largo* e identificar el rango *ideal* [1], [3].

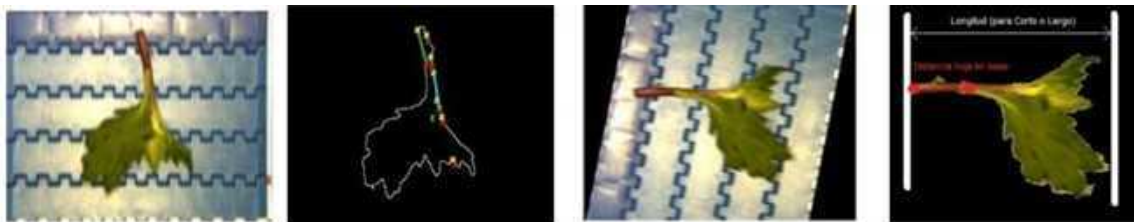


Figura 2.2: Ejemplo de salida de segmentación y medición: máscara binaria del esqueje, contorno estimado y cálculo de longitud  $L_{\text{px}}$  (y conversión a centímetros mediante calibración).

## 2.3. Metodología (formulación formal y pipeline)

### 2.3.1. Variables y vector de características

Sea el vector de características mínimo

$$\mathbf{x} = \begin{bmatrix} L \\ H \end{bmatrix},$$

donde  $L$  es la longitud estimada y  $H \in \{0, 1\}$  indica presencia de *hoja en base* detectada en una región de interés (ROI) en la zona basal del esqueje [1], [3].

### 2.3.2. Regla de decisión por umbrales

Se definen umbrales  $\tau_{\min}$  y  $\tau_{\max}$  que delimitan el rango *ideal*. Una regla operativa (interpretable) es:

$$\hat{c} = \begin{cases} \omega_0, & \text{si no se detecta objeto,} \\ \omega_3, & \text{si } H = 1 \quad (\text{hoja en base}), \\ \omega_1, & \text{si } L < \tau_{\min} \quad (\text{corto}), \\ \omega_2, & \text{si } L > \tau_{\max} \quad (\text{largo}), \\ \omega_4, & \text{si } \tau_{\min} \leq L \leq \tau_{\max} \text{ y } H = 0 \quad (\text{ideal}). \end{cases}$$

Esta formulación es coherente con una selección orientada por criterios geométricos y condiciones basales del material [1], [2].

### 2.3.3. Sincronización banda–cámara–actuación: retardo discreto

La decisión computacional ocurre en el instante  $k$  asociado a la captura, pero la expulsión debe ejecutarse cuando el objeto alcanza la zona neumática. Para modelar este retardo se utiliza un registro desplazable que representa posiciones discretas del transporte:

$$\mathbf{d}[k] = (d_0[k], d_1[k], \dots, d_{N-1}[k]),$$

donde  $d_i[k] \in \Omega$  almacena la etiqueta del eslabón/posición  $i$  en el tiempo discreto  $k$ .

La actualización por avance se modela como:

$$\mathbf{d}[k + 1] = \mathcal{S}(\mathbf{d}[k], \hat{c}[k]),$$

donde  $\mathcal{S}$  desplaza el registro e inserta  $\hat{c}[k]$  en la entrada. Definiendo posiciones de actuación  $\mathcal{P} \subset \{0, \dots, N - 1\}$ , la expulsión ocurre cuando:

$$\exists i \in \mathcal{P} : d_i[k] \in \mathcal{C}_{\text{expulsión}},$$

donde  $\mathcal{C}_{\text{expulsión}}$  es el conjunto de clases que requieren desvío hacia una salida específica [2], [3].

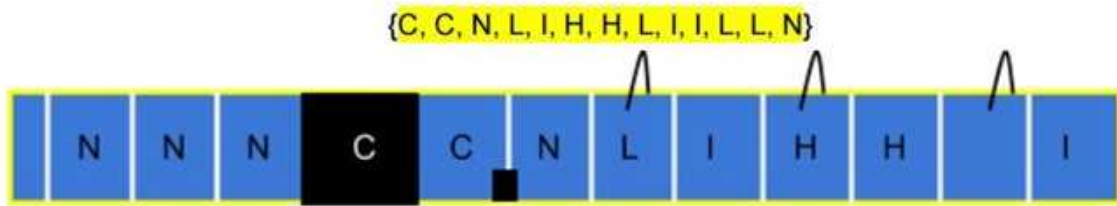


Figura 2.3: Modelo discreto de sincronización: la etiqueta insertada en el tiempo de captura se desplaza con los avances del transporte y se dispara la válvula cuando alcanza posiciones de actuación.  $\mathcal{P}$ . Clasificación = ‘0’ clase = ‘nada’, Clasificación = ‘1’ clase = ‘corto’, Clasificación = ‘2’ clase = ‘largo’, Clasificación = ‘3’ Clase = ‘hbase’, Clasificación = ‘4’ clase = ‘ideal’.

### 2.3.4. Capacidad de proceso: relación transporte–tiempo de cómputo

Sea  $v$  la velocidad de banda [m/s] y  $s$  la separación efectiva entre posiciones discretas [m]. La tasa de llegada es:

$$\lambda = \frac{v}{s} \quad [\text{esquejes/s}],$$

y el tiempo máximo de procesamiento por elemento (sin acumular cola) se aproxima por:

$$T_{\text{máx}} = \frac{1}{\lambda} = \frac{s}{v}.$$

Esta relación vincula la arquitectura de visión (tiempo por frame) con la productividad alcanzable [2], [3].

## 2.4. Resultados y métricas (formulación)

### 2.4.1. Métricas de clasificación

Para una clase objetivo (por ejemplo, *ideal*), sean  $TP$ ,  $FP$ ,  $FN$ ,  $TN$  los conteos estándar. Se definen:

$$\text{Prec} = \frac{TP}{TP + FP}, \quad \text{Rec} = \frac{TP}{TP + FN}, \quad \text{F1} = \frac{2 \text{Prec Rec}}{\text{Prec} + \text{Rec}}.$$

En inspección de esquejes, errores típicos derivan de oclusiones, curvaturas y plegados que afectan la medición  $L$  y la detección de condición basal, lo que se traduce en  $FP$  y  $FN$  operativamente relevantes [2].

### 2.4.2. Limitación estructural: observabilidad 2D y ambigüedad

Bajo adquisición 2D, la geometría real 3D puede proyectarse en medidas aparentemente similares. Por ejemplo, un esqueje curvado puede presentar una longitud proyectada menor que su longitud real; o una hoja basal puede quedar ocluida por pose/solapamiento. Formalmente, si  $g$  es la proyección 3D→2D inducida por la cámara, entonces dos estados físicos distintos  $u_1 \neq u_2$  pueden satisfacer:

$$g(u_1) \approx g(u_2),$$

creando ambigüedad en  $\delta(x)$ . Esto justifica el énfasis en acondicionamiento mecánico del material y diseño del guiado antes de la captura [2].

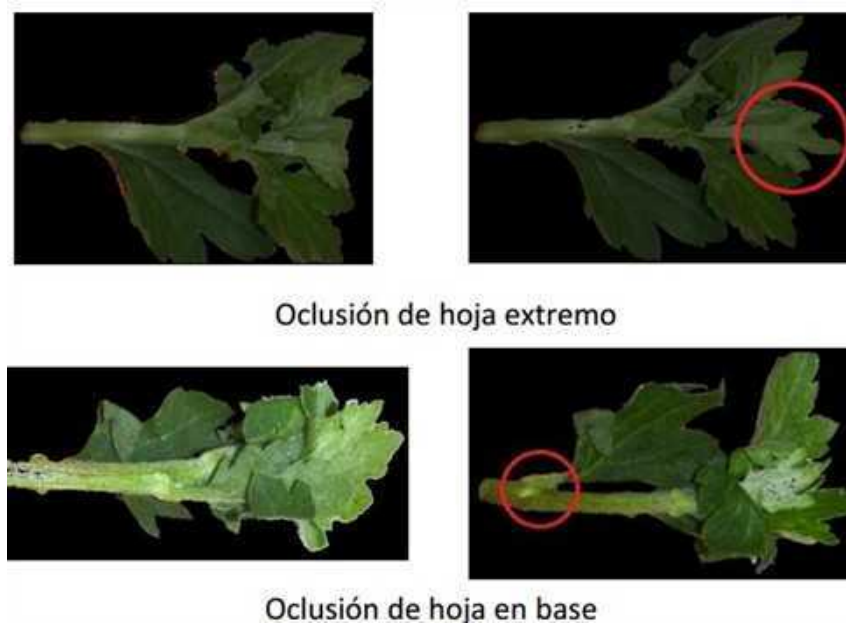


Figura 2.4: Ejemplos de escenarios que inducen errores por observabilidad limitada en 2D: curvaturas que alteran la longitud proyectada y oclusiones/plegados que enmascaran condiciones basales.

## 2.5. Consolidación en campo: ajustes electro–mecánicos y neumáticos

La etapa de consolidación evidenció que el desempeño depende tanto de la decisión computacional como de la repetibilidad física de la expulsión y la conducción hacia el contenedor. Se reportaron ajustes orientados a robustez electromagnética, sintonía neumática y rediseño de guiado y tolvas para estabilizar trayectorias del material expulsado [2]. Estos aprendizajes reafirman una lectura sistémica: la precisión de  $\delta(x)$  solo se traduce en productividad si la cadena *decisión–actuación–trayectoria* se mantiene estable.



Figura 2.5: Ajustes de consolidación: configuración neumática (presión/boquilla) y elementos mecánicos (deflectores/tolvas) para estabilizar la expulsión y conducción hacia la salida.

### 2.6. Referencia a sistemas similares y marco de registro de software

El enfoque de selección por visión sobre banda y eyección/actuación sincronizada es consistente con sistemas industriales de clasificación óptica, donde la cámara adquiere, el algoritmo decide y la expulsión se realiza mediante chorros de aire u otros actuadores en sincronía con el transporte [4], [5], [6]. En Colombia, el registro formal del software se gestiona ante la Dirección Nacional de Derecho de Autor (DNDA), como mecanismo de soporte jurídico para obras de software [7]. Para el caso particular del software SOCAES, el certificado o número de registro puede incorporarse como referencia específica cuando se disponga del dato (si ya existe en tus archivos, lo agregamos a `references.bib` con su identificador).

### 2.7. Conclusiones

Se presentó una formulación formal y mecatrónica de una máquina seleccionadora de esquejes basada en visión artificial y actuación neumática sincronizada. Se definió el problema como decisión  $\delta : \mathcal{I} \rightarrow \Omega$ , se describió la medición morfológica de longitud a partir de segmentación, se propuso una regla de decisión interpretable por umbrales y se modeló la sincronización mediante un retardo discreto implementable como registro

desplazable. Se discutieron métricas estándar (precisión, *recall*) y un límite estructural de observabilidad asociado a la proyección 2D. Finalmente, se reportó la lectura sistémica de consolidación en campo, donde la estabilidad electromecánica y la balística neumática condicionan el desempeño global [1], [2], [3].

## Agradecimientos

A los equipos de los grupos de investigación y a los aliados del sector floricultor que permitieron validar el sistema en condiciones de producción, aportando evidencias y aprendizajes de consolidación que orientan el escalamiento del desarrollo [2].

## Referencias

- [1] *SOCAES: Software de clasificación de esquejes. Manual de usuario*, Documento interno. Incluye descripción de modos de lectura, interfaz y operación del software., Grupo GEPAR, Universidad de Antioquia, Medellín, Colombia, s.f.
- [2] D. S. Fernández Mc Cann, S. Guzmán Obando y D. Jaramillo Grisales, “Informe técnico: Proceso de consolidación de sistema de clasificación de esquejes – Asocolores (Grupo GEPAR),” Universidad de Antioquia, Facultad de Ingeniería, Grupo de Investigación GEPAR, Medellín, Colombia, ene. de 2017, Informe interno de proyecto y consolidación en campo.
- [3] D. Jaramillo Grisales, “Implementación de algoritmo para clasificación y selección de esquejes,” Trabajo de grado (Ingeniería Electrónica), Universidad de Antioquia, Facultad de Ingeniería, Departamento de Ingeniería Electrónica y Telecomunicaciones, Medellín, Colombia, 2017.
- [4] Raytec Vision, *Sorting machines with camera and air-jet expulsion system (Discovery data sheet)*, Ficha técnica de máquina de clasificación: visión + cinta + expulsión por chorro de aire., 2018. dirección: <https://www.sealpac-uk.com/hubfs/Partner%20Page%20Images/Discovery-data-sheet.pdf>.
- [5] Visionerf. “Optical sorting: From image capture to ejection (Visionerf).” Descripción industrial del flujo visión–decisión–eyección en transportadores. Consultado en línea. dirección: <https://visionerf.com/en/food-sorting-and-counting/>.

- [6] D. R. Yana et al., “Low-Cost Machine Vision System for Sorting Green Lentils (Lens Culinaris) Based on Pneumatic Ejection and Deep Learning,” 2025, Preprint en arXiv (sistema comparable: banda + visión + válvulas/boquillas neumáticas sincronizadas). dirección: <https://arxiv.org/pdf/2507.20531>.
- [7] Dirección Nacional de Derecho de Autor (DNDA). “Registro de software (Dirección Nacional de Derecho de Autor – Colombia).” Guía oficial del trámite y requisitos de registro de software. Consultado en línea. dirección: <https://www.derechodeautor.gov.co/es/atencion-y-servicios-a-la-ciudadania/registro-de-obras/registro-de-software/registro-de-software>.

# Capítulo 3

## Conteo automático de colonias virales en pozos de agar

### Autores

Andrea Isabel Trujillo Correa<sup>1</sup>; Hugo Andrés Morales Hoyos<sup>2</sup>; Miguel Ángel Monsalve Londoño<sup>3</sup>; David Fernández Mc Cann<sup>4</sup>.

Grupos PECET y GEPAR, Facultad de Medicina e Ingeniería, Universidad de Antioquia

### Resumen

El conteo manual de colonias virales en pozos de agar es una tarea operativa frecuente que demanda tiempo y experiencia, y es susceptible a fatiga, variabilidad interobservador y errores acumulativos en experimentos extensos [1], [2]. Este capítulo presenta *APVirus*, un software desarrollado en MATLAB para estandarizar y automatizar el conteo, aprovechando el cambio de coloración del agar consumido por las colonias [1], [3]. El método se fundamenta en la hipótesis de crecimiento aproximadamente circular de las colonias [2], [4] y propone un pipeline reproducible: selección de región de interés, preprocesamiento, binarización por Otsu [5], [6], segmentación de manchas y extracción iterativa de

---

<sup>1</sup>Profesora asistente, Facultad de Medicina, Universidad de Antioquia. andrea.trujillo@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. hugo.morales@udea.edu.co

<sup>3</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. miguel.monsalve@udea.edu.co

<sup>4</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. david.fernandez@udea.edu.co

círculos máximos inscritos en regiones binarizadas mediante transformada de distancia y watershed [7], [8]. El sistema entrega el conteo estimado, radios extremos, áreas agregadas, un histograma de radios y un criterio configurable de “cobertura de área” (por defecto 75 %) para auditoría visual de resultados [2]. Se incluyen pautas de implementación, parámetros y una estructura de evaluación para su uso en el marco del seminario Geolimna–GEPAR.

**Palabras clave:** conteo de colonias; pozos de agar; Otsu; watershed; transformada de distancia; segmentación; MATLAB; APVirus.

## 3.1. Introducción

En el flujo manual, un operador dispone la placa a contraluz y realiza el conteo de zonas con ausencia de agar (o pérdida de intensidad) asociadas a colonias, registrando luego los resultados en formatos y hojas de cálculo. Esta práctica requiere atención sostenida y experiencia; a mayor número de pozos y ante condiciones de iluminación variables, mayor probabilidad de fatiga y error [1]. APVirus se propone como una respuesta técnica: formalizar criterios geométricos y operacionales para generar resultados consistentes, portables y auditables [1], [2].

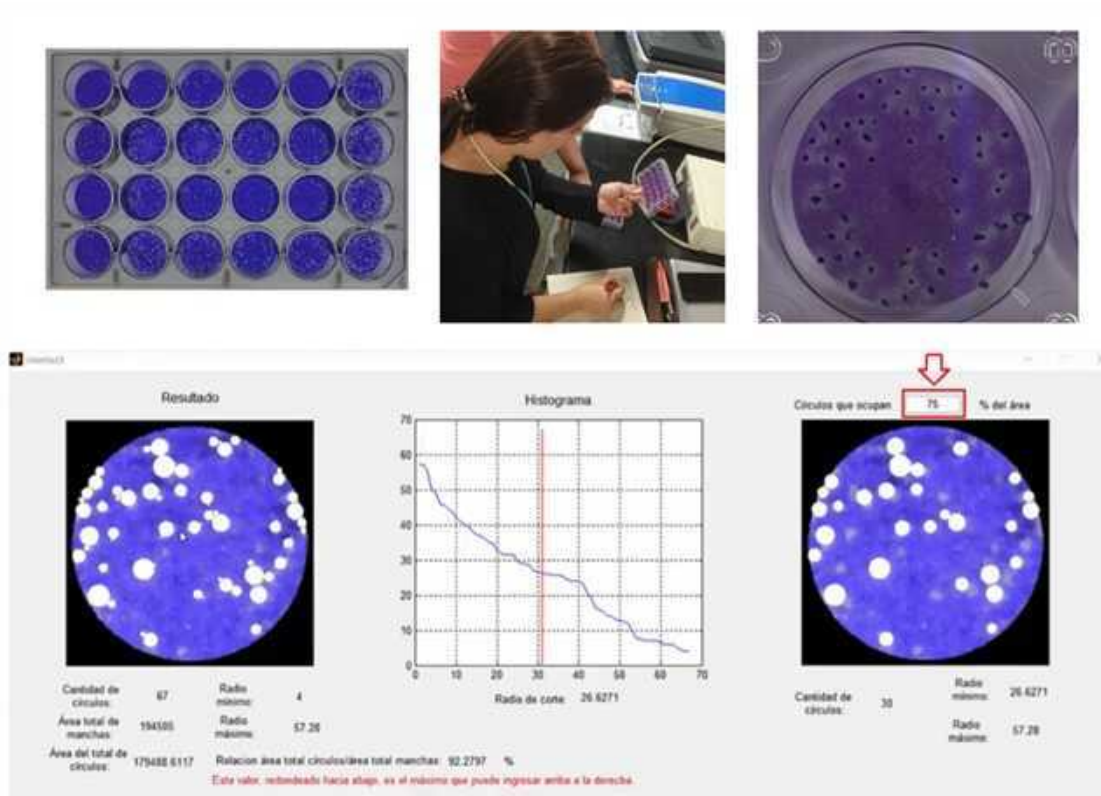


Figura 3.1: Comparación conceptual del flujo manual (observación, marcación e inventario) frente al flujo APVirus (ROI, filtrado, binarización, extracción iterativa de círculos y estadísticas).

## 3.2. Planteamiento del problema (formulación formal)

### 3.2.1. Definición del objetivo de conteo

Sea  $I : \Omega \subset \mathbb{Z}^2 \rightarrow \mathbb{R}$  la imagen (escaneada o capturada) de un pozo. El objetivo es estimar la cardinalidad  $N$  del conjunto de colonias presentes a partir de evidencia visual en el agar:

$$\hat{N} = \mathcal{A}(I; \theta),$$

donde  $\theta$  representa parámetros de operación (por ejemplo, un radio mínimo  $r_{\min}$  fijado por el usuario) [9].

### 3.2.2. Modelo geométrico de colonia y restricciones

A escala de imagen, una colonia se aproxima por un disco:

$$\mathcal{D}(c, r) = \{x \in \mathbb{R}^2 : \|x - c\|_2 \leq r\},$$

con centro  $c \in \mathbb{R}^2$  y radio  $r > 0$ . El problema se reduce a encontrar un conjunto  $\{(c_k, r_k)\}_{k=1}^{\hat{N}}$  tal que:

$$\mathcal{D}(c_k, r_k) \subseteq \mathcal{B}, \quad r_k \geq r_{\text{mín}},$$

donde  $\mathcal{B} \subseteq \Omega$  es la región binaria que contiene manchas candidatas (regiones conectadas) obtenidas tras el umbralado.

### 3.2.3. Objetivos de ingeniería

El diseño del sistema busca: (i) estandarizar el conteo, (ii) reducir tiempos operativos, (iii) entregar estadísticas útiles al investigador, y (iv) ofrecer un mecanismo de validación visual y exportación de resultados [1], [2].

## 3.3. Materiales y métodos

### 3.3.1. Requerimientos técnicos y entorno

APVirus fue desarrollado en MATLAB (R2018a) y utiliza herramientas de umbralado por color (Color Thresholder) [1]. El manual de usuario recomienda como mínimo MATLAB R2018a, 4 GB de RAM y 1 GB de almacenamiento disponible [2]; el manual técnico describe un entorno típico con Windows 10, procesador Intel Core i5, 8 GB RAM y GPU de 2 GB [1].

### 3.3.2. Adquisición y formato

El manual de usuario recomienda imágenes en formato TIFF para el análisis [2]. El pipeline asume una entrada que preserve contraste suficiente entre regiones consumidas y no consumidas del agar.

### 3.3.3. Selección de región de interés (ROI)

El usuario delimita la región de interés mediante una máscara ajustable (tamaño y desplazamiento) que debe cubrir el pozo sin tocar su borde [2]. Sea  $R \subset \Omega$  la ROI; se trabaja con  $I_R = I|_R$  para minimizar interferencias.

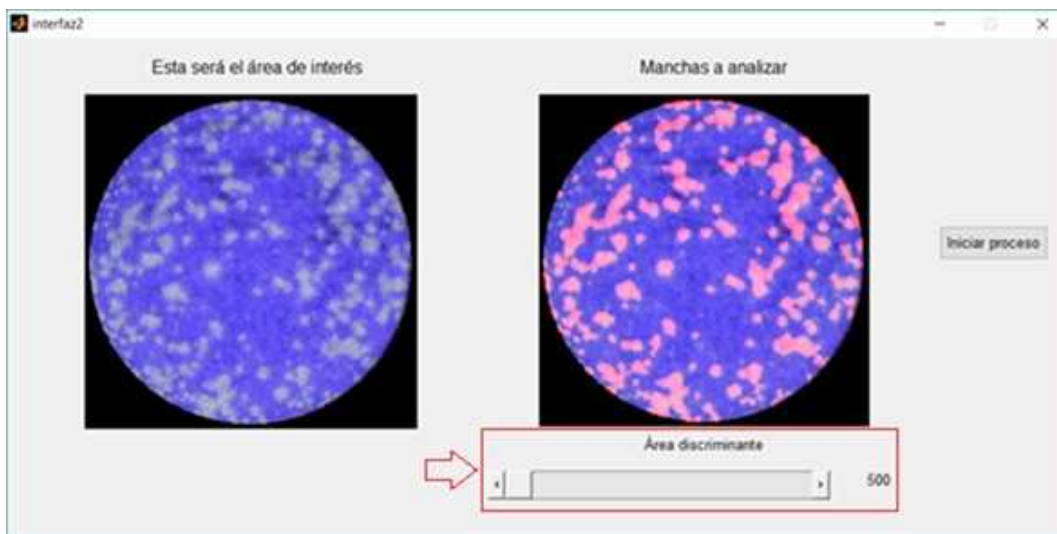


Figura 3.2: Selección de ROI mediante máscara ajustable (tamaño y centrado) previa al análisis.

### 3.3.4. Preprocesamiento

El método aplica filtros de homogenización para estabilizar la segmentación posterior [4], [9]. En forma general:

$$I_1 = \mathcal{F}_{\text{media}}(I_R), \quad I_2 = \mathcal{F}_{\text{promedio}}(I_1),$$

donde  $\mathcal{F}_{\text{media}}$  y  $\mathcal{F}_{\text{promedio}}$  son filtros espaciales con ventanas definidas según configuración del sistema.

### 3.3.5. Binarización por Otsu

Se determina un umbral global  $t^*$  mediante el criterio de Otsu (máxima separabilidad interclase) [5], [6]:

$$B(x) = \mathbf{1}\{I_2(x) \geq t^*\}.$$

Luego se extraen componentes conectadas  $\{S_j\}$  (“manchas”) candidatas a contener una o varias colonias.

### 3.3.6. Filtrado interactivo de manchas

Antes del conteo, el usuario puede descartar manchas pequeñas no relevantes mediante una barra deslizadora, eliminando artefactos y ruido [2].

### 3.3.7. Extracción iterativa de círculos máximos inscritos (núcleo del algoritmo)

Para cada mancha  $S_j$ , se estima iterativamente el círculo máximo inscrito y se sustrae de la región, repitiendo de mayor a menor hasta que el radio cae por debajo de un umbral  $r_{\text{mín}}$  (definido por el usuario) [4], [9]. Sea  $S_j^{(0)} = S_j$  y:

$$(c^{(i)}, r^{(i)}) = \arg \max_{c,r} \{r : \mathcal{D}(c, r) \subseteq S_j^{(i)}\}, \quad S_j^{(i+1)} = S_j^{(i)} \setminus \mathcal{D}(c^{(i)}, r^{(i)}).$$

El cálculo de  $(c^{(i)}, r^{(i)})$  se apoya en la transformada de distancia y en watershed para localizar el centro asociado al mayor radio permitido por la frontera de  $S_j^{(i)}$  [7], [8]. El proceso se detiene cuando  $r^{(i)} < r_{\text{mín}}$  [9].

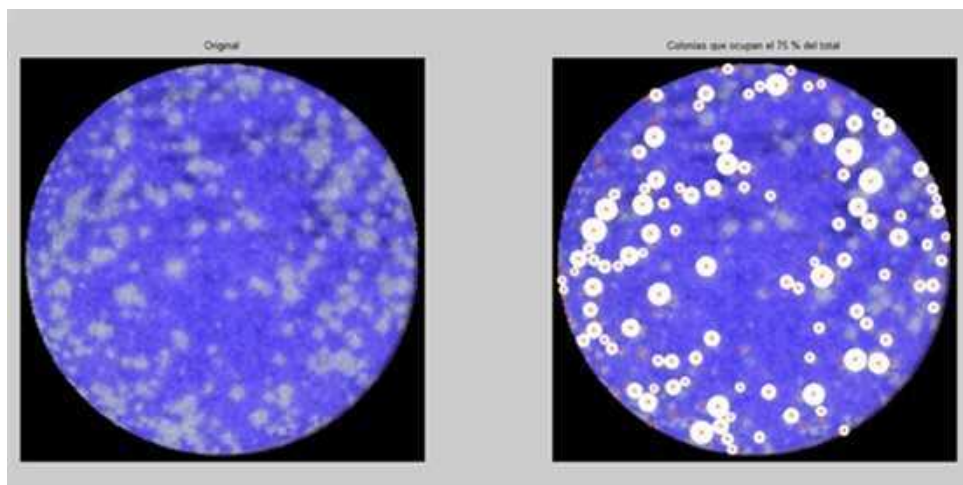


Figura 3.3: Núcleo del algoritmo: región binarizada  $\rightarrow$  transformada de distancia/watershed  $\rightarrow$  centro y radio del círculo máximo  $\rightarrow$  sustracción iterativa hasta  $r < r_{\text{mín}}$ .

### 3.3.8. Salida: conteo y estadísticas

Al finalizar, el software reporta: número de círculos (conteo), radio máximo y mínimo, área total de manchas, área total cubierta por círculos y la relación entre ambas áreas [2]. Además, muestra un histograma de radios ordenados de mayor a menor, con un punto de corte que indica el subconjunto de círculos que componen un porcentaje de referencia (por defecto 75 %) del área total de manchas; dicho porcentaje es ajustable, con la restricción de que no exceda la relación  $\frac{A_{\text{círculos}}}{A_{\text{manchas}}}$  [2].

Sea:

$$A_{\text{manchas}} = \sum_j |S_j|, \quad A_{\text{círculos}} = \sum_k \pi r_k^2, \quad \rho = \frac{A_{\text{círculos}}}{A_{\text{manchas}}}.$$

El porcentaje  $\alpha \in (0, \rho]$  define el conjunto de círculos ordenados por radio tal que su área acumulada alcanza  $\alpha A_{\text{manchas}}$ .

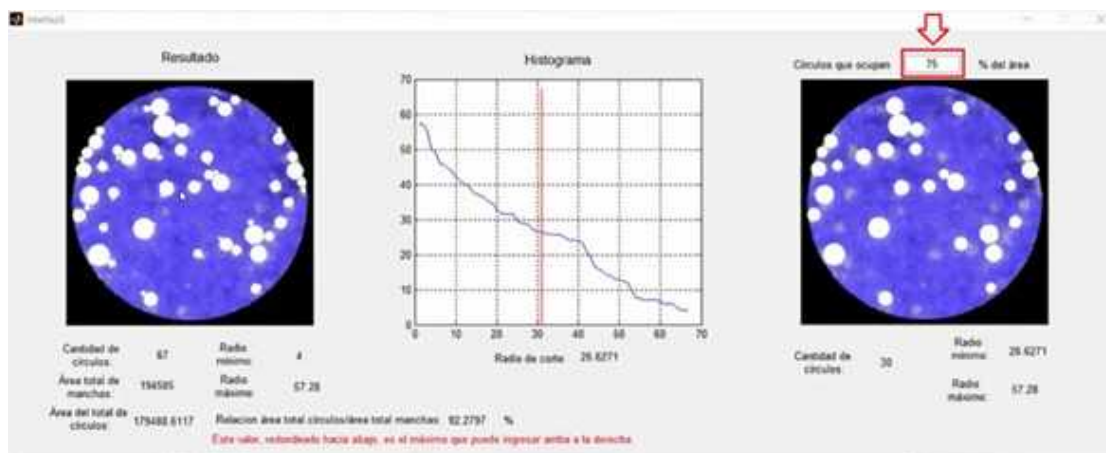


Figura 3.4: Interfaz de resultados: círculos detectados, centros marcados, métricas agregadas e histograma de radios con punto de corte del porcentaje de área.

## 3.4. Implementación software

El manual de usuario describe un flujo por interfaces: (i) carga de imagen, (ii) selección de ROI mediante máscara, (iii) filtrado de manchas y (iv) ejecución del proceso y visualización de resultados [2]. El manual técnico complementa con una visión orientada a mantenimiento, describiendo el objetivo del software y el procedimiento para agregar funciones en la interfaz de MATLAB (archivos `.fig` y `callbacks`) [1].

### 3.4.1. Reproducibilidad y auditabilidad

El diseño prioriza la auditabilidad: el usuario observa (a) la ROI (Fig. 3.2), (b) manchas seleccionadas, (c) círculos y centros (Fig. 3.4), y (d) estadísticas agregadas, con lo cual puede detectar fallos por iluminación, ruido o parámetros mal ajustados [2].

## 3.5. Validación y resultados (estructura de reporte)

Los insumos declaran que el algoritmo fue probado junto a un experto, quien determinó su utilidad [4]. Para formalizar la evaluación, se propone comparar el conteo manual  $N_m$  y el conteo automático  $\hat{N}$  por pozo ( $p = 1, \dots, P$ ), reportando:

$$\text{MAE} = \frac{1}{P} \sum_{p=1}^P \left| N_m^{(p)} - \hat{N}^{(p)} \right|, \quad \text{Bias} = \frac{1}{P} \sum_{p=1}^P \left( \hat{N}^{(p)} - N_m^{(p)} \right),$$

y el error relativo medio:

$$\text{MRE} = \frac{1}{P} \sum_{p=1}^P \frac{|N_m^{(p)} - \hat{N}^{(p)}|}{\text{máx}(1, N_m^{(p)})}.$$

El cuadro 3.1 presenta un ejemplo experimental de comparación con error relativo dentro de  $\pm 5\%$  por muestra.

Tabla 3.1: Comparación entre conteo manual y conteo automático con APVirus, donde se muestra que se logró un error relativo dentro de  $\pm 5\%$  por muestra.

Muestra	Conteo manual	Conteo automático	Error abs. (A–M)	Error rel.
1	120	115	-5	-4.17 %
2	85	88	+3	+3.53 %
3	200	197	-3	-1.50 %
4	150	157	+7	+4.67 %
5	60	58	-2	-3.33 %
6	95	99	+4	+4.21 %
7	175	168	-7	-4.00 %
8	130	132	+2	+1.54 %
9	220	209	-11	-5.00 %
10	105	110	+5	+4.76 %
11	140	137	-3	-2.14 %
12	75	77	+2	+2.67 %

## 3.6. Discusión

El método se alinea con el criterio humano de crecimiento aproximadamente circular y resuelve conglomerados mediante extracción iterativa de círculos máximos. Su desempeño depende de: (i) estabilidad de iluminación y contraste, (ii) correcta selección de ROI (sin tocar borde), (iii) filtrado de manchas no relevantes, y (iv) el hiperparámetro  $r_{\min}$  [2], [9]. La binarización global por Otsu es eficiente, pero sensible a gradientes de iluminación; en escenarios de no uniformidad podría evaluarse corrección de fondo o umbralado adaptativo [8].

El criterio de porcentaje de área (75 % por defecto) aporta una capa de interpretación: permite centrarse en las colonias que explican la mayor porción del área total de manchas, sin perder información de los centros detectados [2]. Este mecanismo, además de informativo, reduce ambigüedad en reportes.

## 3.7. Conclusiones

APVirus formaliza y automatiza el conteo de colonias virales en pozos de agar a partir de imágenes, utilizando un pipeline reproducible: ROI, filtrado, binarización por Otsu y extracción iterativa de círculos máximos inscritos apoyada en transformada de distancia/watershed. El software entrega conteo, métricas geométricas y una interfaz auditable, con exportación para análisis posterior [1], [2]. Se deja explícito su registro como software ante la DNDA y la titularidad institucional, lo que apoya trazabilidad y transferencia [3].

## Agradecimientos

A la investigadora del grupo PECET y al equipo GEPAR involucrado en el desarrollo del software; así como al seminario Geolimna–GEPAR por la articulación del problema experimental con una solución técnica reproducible [1], [2].

## Referencias

- [1] *APVirus: Manual Técnico*, Insumo: Manual técnico APVirus.docx, Universidad de Antioquia, Grupos PECET y GEPAR, 2019.
- [2] *APVirus: Manual de Usuario*, Insumo: Manual de usuario APVirus.docx, Universidad de Antioquia, Grupos PECET y GEPAR, 2019.
- [3] D. Fernández Mc Cann, A. I. Trujillo Correa, H. A. Morales Hoyos y M. Á. Monsalve Londoño, *APVIRUS: Certificado de Registro de Soporte Lógico (Software), Libro-Tomo-Partida 13-75-451*, Dirección Nacional de Derecho de Autor (DNDA), Ministerio del Interior, Colombia, Fecha de registro: 20-sep-2019. Titular del derecho patrimonial: Universidad de Antioquia, sep. de 2019.
- [4] *Materiales y métodos: método de conteo de colonias de virus basado en técnicas de procesamiento digital de imágenes*, Documento interno del seminario GeoLimna-GEPAR, Insumo: materiales y metodos apvirus.docx.
- [5] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, n.º 1, págs. 62-66, 1979.
- [6] N. Zhu, G. Wang, G. Yang y W. Dai, "A fast 2D Otsu thresholding algorithm based on improved histogram," en *Proceedings of the 2009 Chinese Conference on Pattern Recognition (CCPR) and the 1st CJK Joint Workshop on Pattern Recognition (CJKPR)*, 2009.
- [7] L. Vincent y P. Soille, "Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, n.º 6, págs. 583-598, 1991.
- [8] R. C. Gonzalez, R. E. Woods y B. R. Masters, "Digital Image Processing, Third Edition," *Journal of Biomedical Optics*, vol. 14, n.º 2, pág. 029 901, 2009.
- [9] *Método de conteo de colonias de virus basado en técnicas de procesamiento digital de imágenes (APVirus)*, Documento interno del seminario GeoLimna-GEPAR, Insumo: paper\_apvirus\_ver\_3\_ago.docx.

# Capítulo 4

## Inventario automático de áreas de líquenes cortícolas

### Autores

Santiago Ruiz González<sup>1</sup>; Maycol Esteban Zuluaga Montoya<sup>2</sup>; Mauricio Andrés Correa Ochoa<sup>3</sup>; David Fernández Mc Cann<sup>4</sup>.

Grupos GLIMA, Geolimna y Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

Los líquenes cortícolas han sido ampliamente empleados como bioindicadores para evaluar calidad del aire, gracias a su sensibilidad a contaminantes y a su capacidad para reflejar condiciones ambientales integradas en el tiempo. En metodologías de campo consolidadas, la cobertura líquénica se registra sobre el forófito mediante acetatos transparentes (películas PET) donde el investigador traza manualmente contornos de especies/morfotipos con marcadores de color; estos registros luego se cuantifican con herramientas manuales o software generalista, generando un flujo lento y propenso a doble trabajo [1], [2], [3].

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica, Universidad de Antioquia. santiago.ruiz@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica, Universidad de Antioquia. maycol.zuluaga@udea.edu.co

<sup>3</sup>Profesor asociado, Departamento de Ingeniería Ambiental y Sanitaria, Universidad de Antioquia. mauricio.correa@udea.edu.co

<sup>4</sup>Profesor titular, Departamento de Ingeniería Electrónica, Universidad de Antioquia. david.fernandez@udea.edu.co

Este capítulo presenta *ARLIQ*, un software registrado ante la DNDA que automatiza el conteo e inventario de áreas de líquenes a partir de imágenes digitales correspondientes a los trazos de contornos plasmados sobre acetatos [4]. El método implementa un pipeline: marcación del acetato, escaneo a color en formato sin compresión, eliminación de fondo, segmentación por color en un espacio cromático robusto (CIE Lab) con apoyo de  $k$ -means en zonas de incertidumbre, reparación topológica de contornos (cierre de discontinuidades e islas) mediante morfología matemática y esqueletización, y cálculo de área por relleno e integración de píxeles, con exportación de resultados para inventarios [1], [2], [5]. Se presentan además los principales modos de fallo (mezcla de tintas, intersecciones, compresión y trazos incompletos) y las estrategias de mitigación descritas en los documentos del proyecto.

**Palabras clave:** líquenes cortícolas; bioindicación; cobertura liquénica; segmentación por color; CIE Lab;  $k$ -means; morfología matemática; topología digital; inventario de áreas; ARLIQ.

### 4.1. Introducción

La gestión ambiental urbana requiere instrumentos que permitan medir cambios cuantitativos en indicadores esenciales de calidad ambiental. En el contexto de calidad del aire, las redes instrumentales tradicionales pueden ser costosas, demandar mantenimiento especializado y ofrecer mediciones puntuales, lo cual limita su despliegue extensivo [1]. La bioindicación con líquenes aporta una alternativa de bajo costo y con información espacial integrada, usando métricas como cobertura y diversidad para derivar índices ambientales [1], [3].

En campañas regulares de inventario, el registro manual de coberturas puede incluir del orden de decenas a cientos de acetatos, cuya cuantificación manual (delimitación + medición + tabulación) genera un cuello de botella [2], [6]. ARLIQ nace para automatizar esta etapa: convertir registros manuales protocolizados en insumos digitales cuantificables, reduciendo tiempo de operación y habilitando densificación del muestreo en espacio y tiempo [1].

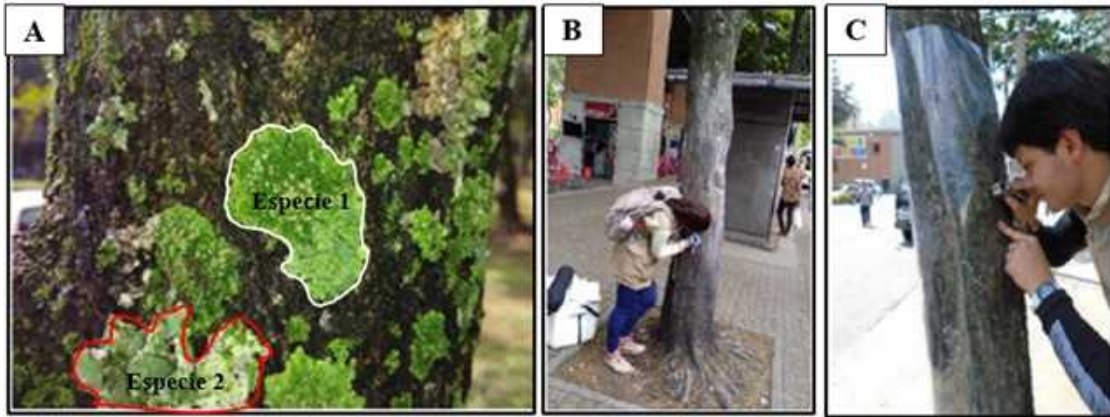


Figura 4.1: Contexto del biomonitoring con líquenes cortícolas y necesidad de cuantificar coberturas a partir de registros de campo. Metodología para el mapeo de líquenes cortícolas. A. Inspección visual de los forófitos para el reconocimiento inicial de las especies de líquenes. B. Mapeo de líquenes. C. Identificación de diferencias en las especies de líquenes.

## 4.2. Planteamiento del problema (formulación formal)

### 4.2.1. Cobertura como variable cuantitativa

Sea  $A_T$  el área total mapeada del forófito (por ejemplo,  $50 \times 100 \text{ cm}^2$  en acetato PET fijado a la corteza) [1]. Sea  $\mathcal{C} = \{1, \dots, K\}$  el conjunto de clases (morfotipos/especies) codificadas por convención de color (incluyendo combinaciones por punto interior cuando aplica) [2]. El objetivo del inventario es estimar el área cubierta por cada clase:

$$\hat{A}_k = \Phi_k(I; \theta), \quad k \in \mathcal{C},$$

donde  $I$  es la imagen digital del acetato escaneado y  $\theta$  agrupa parámetros de segmentación, reparación y factor de escala.

La cobertura relativa por clase se define como:

$$\hat{p}_k = \frac{\hat{A}_k}{A_T}, \quad \sum_{k=1}^K \hat{p}_k \leq 1,$$

y puede usarse para índices derivados (por ejemplo, índices de pureza atmosférica) [1].

### 4.2.2. Dificultades técnicas del dato: color, topología y trazos

En el acetato, los contornos son trazados con marcadores cuya tinta no es un color puntual sino una distribución en el espacio cromático; además, el escaneo puede introducir variaciones y artefactos (p. ej. compresión) que aumentan la incertidumbre de clasificación por color [2]. A ello se suma que muchos contornos no son cerrados, presentan discontinuidades o islas, e intersecciones entre trazos, impidiendo un relleno directo sin correcciones [2].

Así, el problema computacional se formula como: (i) segmentar con robustez las trayectorias por clase, (ii) reparar la topología de curvas para garantizar cerramiento, (iii) separar intersecciones y resolver ambigüedades, y (iv) estimar áreas internas con trazabilidad.

## 4.3. Materiales y métodos

### 4.3.1. Visión general del pipeline

El flujo metodológico adoptado se resume como: marcación del acetato, escaneo a color, eliminación de fondo, reparación de contornos, extracción de áreas por líquen y almacenamiento/exportación [5]. Este flujo se integra en ARLIQ, software registrado ante DNDA [4].

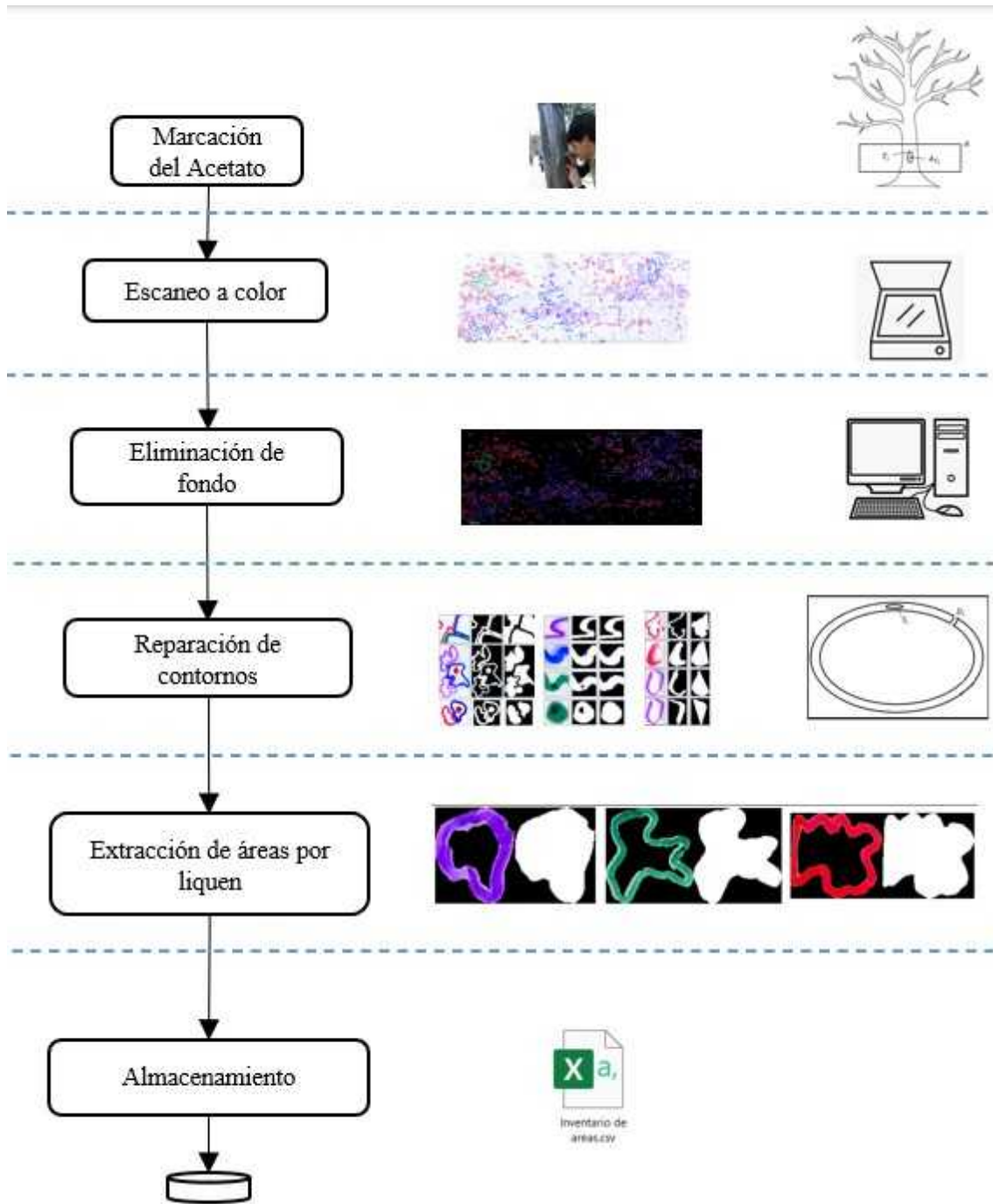


Figura 4.2: Flujo general de ARLIQ: Marcación → Escaneo → Eliminación de fondo → Reparación de contornos → Extracción de áreas → Almacenamiento/exportación.

### 4.3.2. Adquisición: mapeo en campo y digitalización

El registro en campo consiste en fijar una película PET (acetato) sobre la corteza y trazar manualmente los contornos de líquenes asignando un color por morfotipo; ante dudas taxonómicas se realiza verificación con lupa [1]. Los acetatos se digitalizan con escáner de gran formato en escala 1:1, incluyendo escala métrica, y se guardan sin compresión en formato .TIF para preservar información cromática [1].

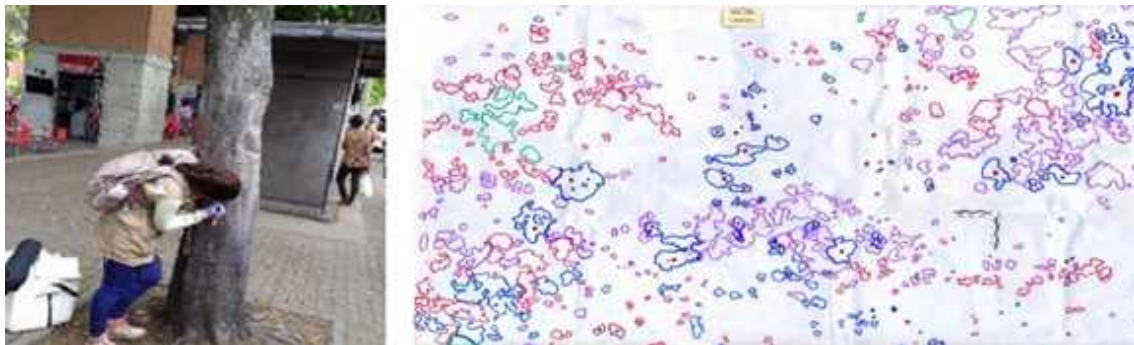


Figura 4.3: A la izquierda Acetato PET fijado sobre la corteza para mapeo de contornos. A la derecha Acetato digitalizado con escala métrica y contornos multicolor.

### 4.3.3. Modelo de segmentación por color y manejo de incertidumbre

Sea  $I : \Omega \rightarrow \mathbb{R}^3$  la imagen RGB. Se transforma a un espacio cromático perceptualmente más estable (CIE  $L^*a^*b^*$ ) para separar clases por componentes cromáticas [2], [7]. Denotamos  $J = \Psi(I)$  la imagen en CIE Lab.

Para cada clase  $k$ , se define un conjunto de prototipos de color (o rangos) que describen la tinta del marcador:

$$\mathcal{R}_k \subset \mathbb{R}^3, \quad k \in \mathcal{C}.$$

El primer paso produce una máscara preliminar:

$$M_k(x) = \mathbf{1}\{J(x) \in \mathcal{R}_k\}.$$

Debido a superposición/intersección de regiones cromáticas entre tintas (mezclas no puras), se define una zona de incertidumbre:

$$U = \{x : \exists k \neq \ell \text{ tal que } J(x) \in \mathcal{R}_k \cap \mathcal{R}_\ell\} \cup \{x : J(x) \notin \cup_k \mathcal{R}_k\}.$$

Sobre  $U$ , se aplica un esquema de agrupamiento ( $k$ -means o mezclas gaussianas) para

asignar etiquetas minimizando la ambigüedad [2], [8].

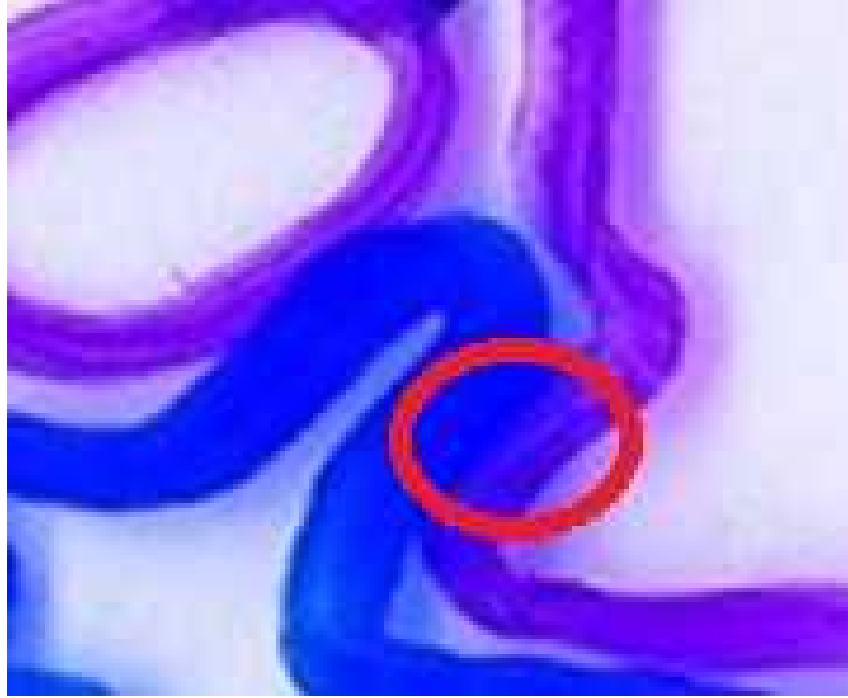


Figura 4.4: Distribución de colores de marcadores en CIE Lab y zonas de incertidumbre por intersección entre clases.

#### 4.3.4. Eliminación de fondo y consolidación de trazos

Se elimina el fondo del acetato para aislar únicamente trazos y artefactos, y se depura ruido por filtrado morfológico y por área mínima. La salida es una máscara por clase con trazos dominantes y reducción de islas no deseadas [2].

#### 4.3.5. Reparación topológica: cierre de contornos y corrección de discontinuidades

Para estimar áreas internas se requiere que las curvas que delimitan líquenes sean cerradas. En presencia de discontinuidades e islas, se asume que el contorno observado corresponde a una curva cerrada ideal y se realiza un cierre mediante operaciones morfológicas y/o esqueletización:

$$M'_k = \mathcal{T}(M_k),$$

donde  $\mathcal{T}$  busca garantizar conectividad y cerramiento. Conceptualmente, se utiliza la idea de que una curva cerrada separa interior y exterior (análoga al teorema de Jordan en topología digital) [9]. Operativamente, se localizan extremos y puntos de ramificación del esqueleto para unir extremos mediante segmentos mínimos y cerrar el ciclo [10].

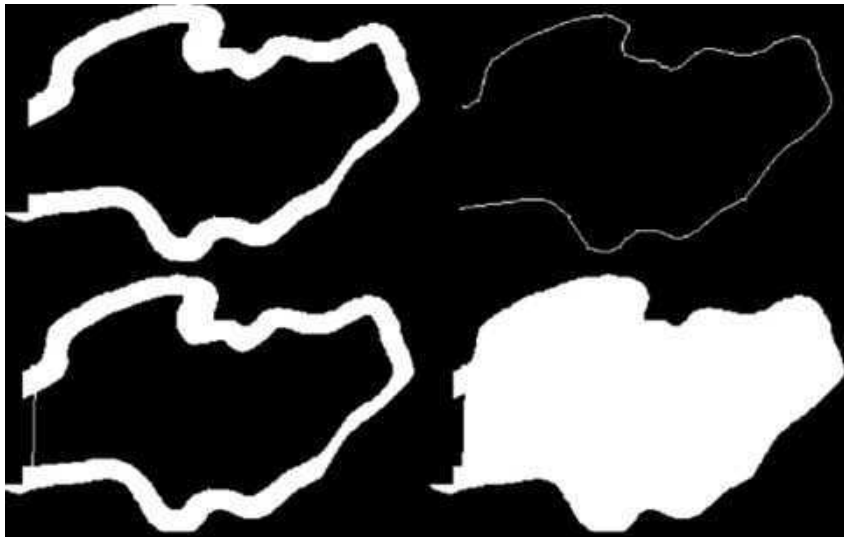


Figura 4.5: Ejemplos de (i) contorno incompleto arriba a la izquierda, (ii) esqueletización y detección de extremos arriba derecha, (iii) cierre del contorno abajo izquierda, (iv) contorno apto para relleno abajo derecha.

#### 4.3.6. Separación de intersecciones y conflictos entre trazos

Cuando dos trayectorias se intersectan, se decide la asignación por predominio cromático local y/o por consistencia geométrica del trazo delimitante, calculando el área por dos criterios (área “dentro” y área “con trazo”) y seleccionando la opción coherente con la clase [2]. Esta estrategia permite mantener inventarios aun con trazos humanos imperfectos.

#### 4.3.7. Cálculo de área por relleno e integración discreta

Una vez obtenida una máscara cerrada de la región interior  $R_k \subset \Omega$ , el área en píxeles se estima como:

$$\widehat{A}_k^{\text{px}} = \sum_{x \in \Omega} \mathbf{1}\{x \in R_k\}.$$

Con un factor de escala  $s$  ( $\text{cm}^2/\text{px}$ ) derivado de la escala métrica en el escaneo:

$$\widehat{A}_k = s \widehat{A}_k^{\text{px}}.$$

El software almacena resultados por clase y exporta tablas para consolidación del inventario [1], [2].

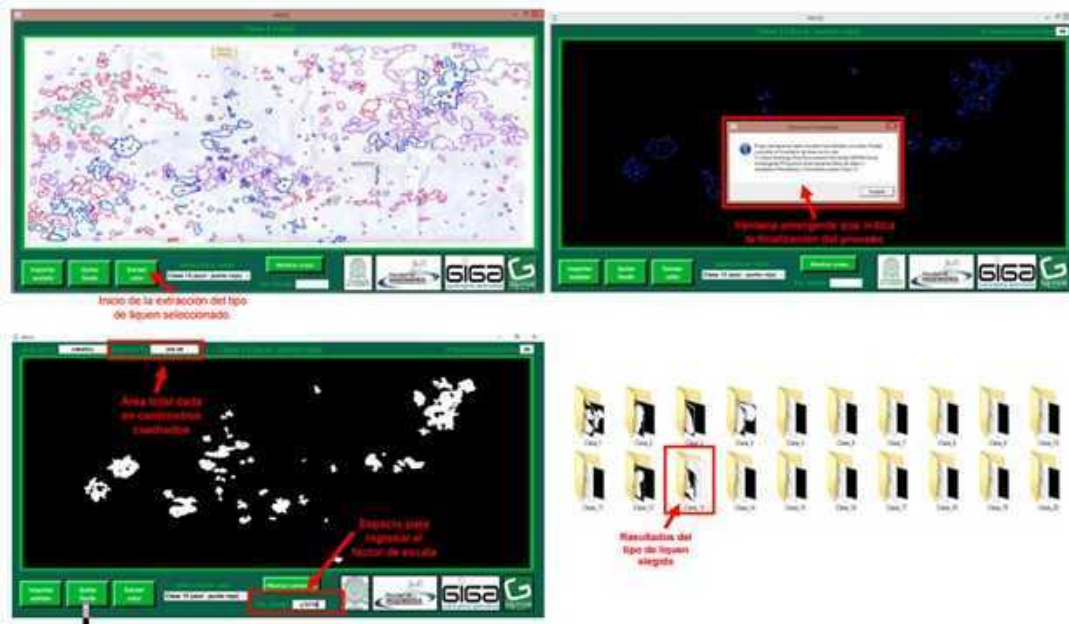


Figura 4.6: Relleno de contornos por clase y medición de área: izquierda arriba trazos, derecha arriba contornos cerrados, izquierda abajo regiones rellenas, derecha abajo reporte de áreas por objeto/clase.

### 4.4. Resultados (estructura de reporte)

Los documentos del proyecto reportan inventarios por clase y distinguen contornos cerrados vs. abiertos, así como área acumulada asociada a cada caso [2]. En un ejemplo de inventario, se reportan conteos de contornos por clase y áreas en píxeles cuadrados para contornos cerrados y abiertos, evidenciando la relevancia de la reparación topológica [2].

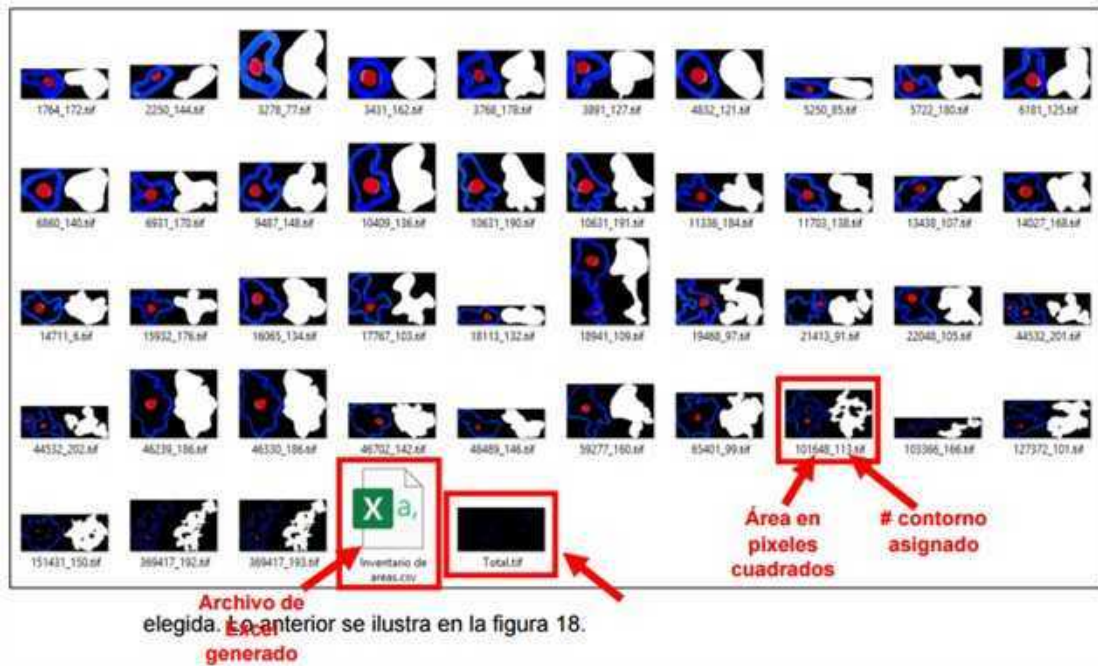


Figura 4.7: Tabla de resultados por clase: número de contornos abiertos/cerrados y áreas totales ( $\text{px}^2$  y/o  $\text{cm}^2$ ), mostrando el efecto de la reparación de contornos. Se especifica tanto por imagen como en un archivo tipo Excel para su fácil trazabilidad.

A nivel institucional, ARLIQ se consolidó como producto de colaboración GEPAR–GIGA para agilizar el levantamiento y la cuantificación de coberturas líquénicas y apoyar evaluación ambiental urbana [6]. Además, el software cuenta con registro DNDA (30-ago-2019) y titularidad patrimonial de la Universidad de Antioquia [4].

## 4.5. Discusión

El aporte técnico de ARLIQ es transformar un registro manual protocolizado (acetatos trazados en campo) en un inventario cuantitativo reproducible, evitando el “doble esfuerzo” de delineación manual en software generalista y tabulación posterior [1], [6]. Desde el punto de vista algorítmico, el principal reto no es el cálculo de área en sí, sino la robustez frente a: (i) variabilidad cromática de tintas (no puntualidad en CIE Lab), (ii) intersecciones y mezclas, (iii) artefactos del escaneo y (iv) contornos no cerrados con islas y discontinuidades [2].

El uso de imágenes sin compresión (TIF) mitiga creación de colores espurios inducidos por pérdidas típicas de compresión [1], [11]. La reparación morfológica y el cierre topológico constituyen el puente entre un trazo humano imperfecto y una región computable por

relleno, habilitando inventarios confiables sin exigir perfección de campo.

Como línea futura natural (planteada en el proyecto), se propone migrar de acetatos a registro fotográfico directo del tallo con construcción de mosaicos (*stitching*) y segmentación supervisada, para automatizar también la fase de levantamiento [2], [12].



Figura 4.8: Línea futura: mosaico panorámico del tallo mediante *stitching* y posterior segmentación supervisada para automatizar el levantamiento sin acetatos.

## 4.6. Conclusiones

Se presentó ARLIQ como desarrollo software para conteo e inventario de áreas de líquenes cortícolas, registrado ante DNDA, y orientado a convertir registros manuales en acetatos digitalizados en inventarios cuantitativos reproducibles [1], [4]. El método integra segmentación por color en CIE Lab con resolución de incertidumbre por agrupamiento, reparación topológica mediante morfología y esqueletización, y cálculo de áreas por relleno e integración discreta [2], [9], [10]. El resultado operacional reduce tiempos de procesamiento y habilita densificación del muestreo para biomonitoreo de calidad del aire [6].

## Agradecimientos

A los grupos GEPAR y GIGA por la articulación entre el problema ambiental (bioindicación con líquenes) y la solución técnica (visión artificial y automatización del inventario). Asimismo, al equipo de campo por el levantamiento en acetatos y a la Universidad de Antioquia por el soporte institucional.

## Referencias

- [1] *Metodología para la obtención de inventarios de coberturas de líquenes cortícolas mediante técnicas de procesamiento de imágenes*, Documento interno del proyecto (manuscrito), Insumo: Manuscript.docx.
- [2] *Inventario de áreas de líquenes sobre árboles y arbustos mediante técnicas de procesamiento de imágenes*, Documento interno del proyecto (borrador de artículo), Insumo: 0\_Paper\_inventario\_liquenes\_utilizando\_PDI\_.docx / 1\_Paper\_inventario\_liquenes\_utilizando\_PDI\_04\_jun\_2019.docx.
- [3] J. Monge-Nájera, M. I. González, M. Rivas Rossi y V. H. Méndez, “A new method to assess air pollution using lichens as bioindicators,” *Revista de Biología Tropical*, vol. 50, n.º 1, págs. 321-325, 2002.
- [4] D. Fernández Mc Cann, S. Ruiz González, M. E. Zuluaga Montoya y M. A. Correa Ochoa, *ARLIQ: Certificado de Registro de Soporte Lógico (Software)*, Libro-Tomo-Partida 13-75-192, Dirección Nacional de Derecho de Autor (DNDA), Ministerio del Interior, Colombia, Fecha de registro: 30-ago-2019. Titular del derecho patrimonial: Universidad de Antioquia, ago. de 2019.
- [5] *Flujo de procesamiento ARLIQ (marcación, escaneo, fondo, reparación, extracción, almacenamiento)*, Diapositiva interna del proyecto, Insumo: graficos\_arliq.pptx.
- [6] *ARLIQ, el software para inventariar los líquenes de la UdeA y la ciudad*, Portal institucional Universidad de Antioquia (nota divulgativa), Insumo: Arliq\_merge.pdf (captura del portal UdeA), 2020.
- [7] Hunter Associates Laboratory, *CIE L\* a\* b\* Color Scale*, Technical note, 2008.
- [8] M. Ivanovici, N. Richard y D. Paulus, “Color Image Segmentation,” en *Advanced Color Image Processing and Analysis*, 2013.
- [9] U. Eckhardt y L. J. Latecki, *Digital Topology*, Preprint, Hamburg. Beiträge zur Angewandten Mathematik, 1994.

- [10] J. Goutsias, L. Vincent y D. S. Bloomberg, “Mathematical Morphology and Its Applications to Image and Signal Processing,” en *Proceedings / collection on mathematical morphology*, 2000.
- [11] G. K. Wallace, “The JPEG Still Picture Compression Standard,” *IEEE Transactions on Consumer Electronics*, 1992.
- [12] *SISUAL / Stitching para mosaico del tallo: línea futura del inventario de líquenes*, Presentación interna del seminario GeoLimna–GEPAR, Insumo: presentación\_proyecto\_stitching.pptx.

# Capítulo 5

## Extracción automática de características geométricas de microperforaciones en placas GEM

### Autores

Tatiana Escobar Alzate<sup>1</sup>; David Fernández Mc Cann<sup>2</sup>; Nelson Vanegas Arbeláez<sup>3</sup>.  
Grupos GFIF y Gepar, Instituto de Física y Facultad de Ingeniería, Universidad de Antioquia

### Resumen

Las placas GEM (*Gas Electron Multiplier*) son componentes críticos en detectores gaseosos para física de altas energías, y su desempeño depende de la geometría de sus microperforaciones (del orden de decenas de micrómetros) y de defectos inducidos por fabricación o exposición. Este capítulo presenta *ExtraGEM*, una aplicación de escritorio desarrollada en MATLAB para extraer automáticamente características geométricas y estadísticas de perforaciones a partir de micrografías, con el fin de apoyar análisis de calidad y comparación entre proveedores en instrumentación vinculada a experimentos del CERN [1], [2], [3], [4], [5]. El método integra un pipeline reproducible: carga de imagen, preprocesamiento,

---

<sup>1</sup>Estudiante de maestría, Ingeniería de Sistemas, Universidad de Antioquia. tatiana.escobar@udea.edu.co

<sup>2</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones, Universidad de Antioquia. david.fernandez@udea.edu.co

<sup>3</sup>Profesor asociado, Instituto de Física, Universidad de Antioquia. nelson.vanegas@udea.edu.co

binarización tipo Otsu, depuración por borde y por área mínima, etiquetado y extracción de propiedades mediante `regionprops`, con exportación a CSV [2], [3]. El software cuenta con registro ante la DNDA y licencia de uso institucional para GFIF y GEPAR [6], [7].

**Palabras clave:** GEM; microperforaciones; inspección por visión; Otsu; `regionprops`; excentricidad; estadística de defectos; MATLAB; ExtraGEM.

## 5.1. Introducción

En detectores gaseosos modernos, la estabilidad y ganancia dependen de estructuras microfabricadas cuyas dimensiones deben permanecer dentro de tolerancias. En placas GEM, millones de microperforaciones distribuidas sobre áreas extensas hacen impracticable una inspección manual completa; incluso muestreos manuales limitados pueden introducir sesgos y baja potencia estadística [1], [4]. Se requieren por tanto herramientas de análisis automatizado que permitan medir grandes volúmenes de perforaciones con trazabilidad y control de calidad.

ExtraGEM integra, en una sola aplicación, las etapas necesarias para: (i) transformar micrografías en representaciones binarias de perforaciones, (ii) depurar ruido y artefactos, (iii) medir propiedades geométricas por objeto y (iv) exportar resultados para análisis estadístico y comparativo [1], [2]. El desarrollo fortalece capacidades de servicio interno: GFIF obtiene una herramienta objetiva de extracción de características y GEPAR consolida capacidad de construir aplicaciones de visión artificial transferibles [1].

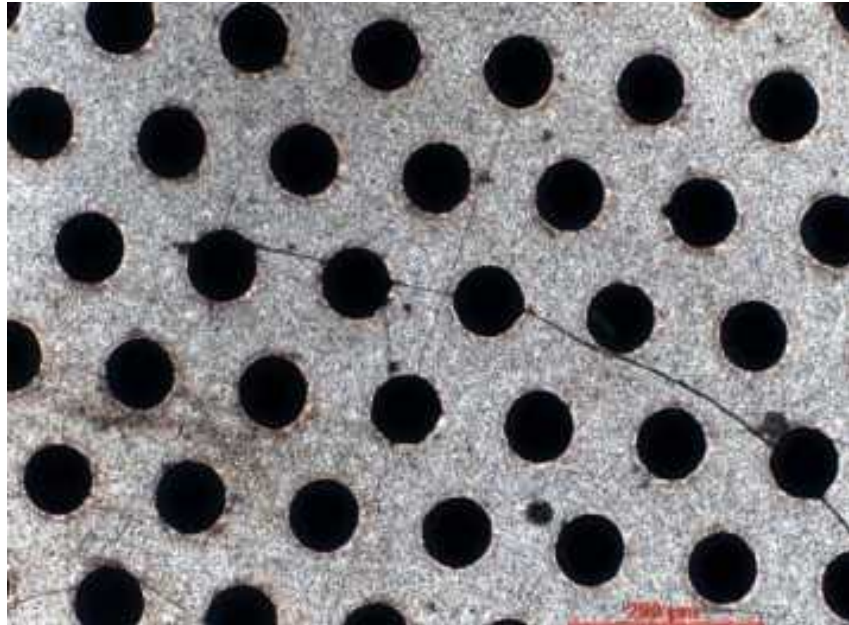


Figura 5.1: Micrografía típica de una placa GEM. Las perforaciones aparecen como regiones repetitivas con contraste respecto al sustrato.

## 5.2. Planteamiento del problema (formulación formal)

### 5.2.1. Objetivo de extracción de propiedades

**Definición del dato.** Sea  $I$  una micrografía digital. Denotamos por  $\Omega \subset \mathbb{Z}^2$  el conjunto de coordenadas de píxel de la imagen (dominio discreto), y por  $I : \Omega \rightarrow \mathbb{R}^3$  una imagen RGB que asigna a cada píxel un vector de intensidades.

**Objetos de interés.** Una vez procesada la imagen, se espera obtener un conjunto de  $N$  objetos conectados  $\{O_i\}_{i=1}^N$ , donde cada  $O_i \subset \Omega$  corresponde a una perforación.

**Características por objeto.** Para cada perforación  $O_i$ , se define un vector de características geométricas

$$\mathbf{f}(O_i) = [A_i, d_i, e_i, \theta_i, P_i, a_i, b_i]^\top,$$

donde:  $A_i$  es el área (en píxeles),  $d_i$  es el diámetro equivalente,  $e_i$  es la excentricidad,  $\theta_i$  es la orientación,  $P_i$  es el perímetro,  $a_i$  es el eje mayor,  $b_i$  es el eje menor. Estas magnitudes son las reportadas por el software [2].

**Objetivo global.** El sistema debe producir: (i) los valores  $\mathbf{f}(O_i)$  para  $i = 1, \dots, N$  y (ii) estadísticas agregadas para cada característica (mínimo, máximo, media, varianza), útiles

para comparar placas (radiadas vs. nuevas) o proveedores [1].

### 5.2.2. Estadísticos agregados

**Conjunto de muestras.** Sea  $f_k(O_i)$  la componente  $k$ -ésima de  $\mathbf{f}(O_i)$ , con  $k \in \{1, \dots, 7\}$ . Para cada  $k$  se define el conjunto de muestras  $\{f_k(O_i)\}_{i=1}^N$ .

**Estadísticos.** Los estadísticos agregados se definen como:

$$f_{k,\text{mín}} = \min_{1 \leq i \leq N} f_k(O_i), \quad f_{k,\text{máx}} = \max_{1 \leq i \leq N} f_k(O_i),$$
$$\mu_k = \frac{1}{N} \sum_{i=1}^N f_k(O_i), \quad \sigma_k^2 = \frac{1}{N-1} \sum_{i=1}^N (f_k(O_i) - \mu_k)^2.$$

ExtraGEM reporta precisamente este tipo de resumen (min, max, media, varianza) por característica [2].

### 5.2.3. Desafíos del dato

**Variabilidad e interferencias.** En micrografías reales se presentan variaciones de iluminación, textura del sustrato, partículas, y artefactos en bordes; por ello el pipeline debe incluir depuración por borde (para eliminar objetos conectados al marco) y un filtro por área mínima para descartar ruido de pequeña escala [2], [3].

## 5.3. Metodología (formulación formal y pipeline)

### 5.3.1. Visión general del flujo

**Etapas del pipeline.** ExtraGEM sigue un flujo reproducible: (1) lectura de imagen, (2) filtrado, (3) conversión a grises, (4) binarización tipo Otsu, (5) depuración por borde y área mínima, (6) etiquetado de objetos, (7) extracción de propiedades y (8) exportación [2], [3].

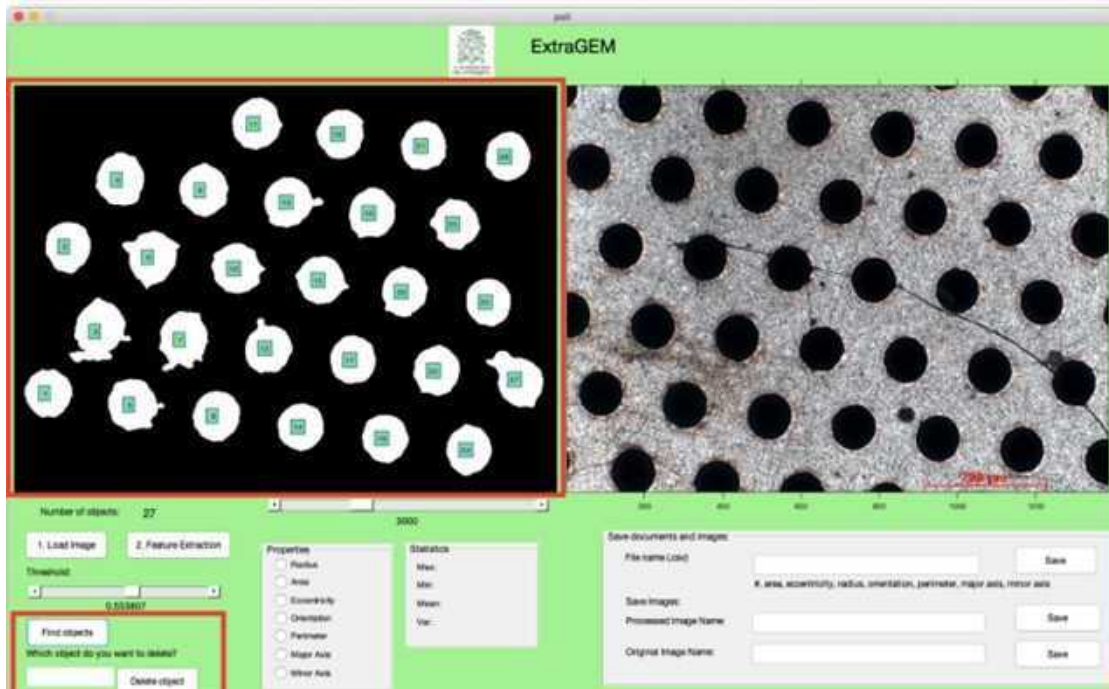


Figura 5.2: Pipeline en interfaz gráfica de ExtraGEM desde micrografía hasta exportación de características y estadísticos.

### 5.3.2. Preprocesamiento

**Imagen suavizada.** Sea  $h$  un kernel de filtrado espacial (por ejemplo, promedio o gaussiano) y  $*$  el operador de convolución discreta. Se define la imagen suavizada:

$$I_s = I * h.$$

Este paso reduce ruido de alta frecuencia, estabiliza el contraste y facilita la segmentación [3].

### 5.3.3. Conversión a grises

**Canal de luminancia.** Sea  $G : \Omega \rightarrow \mathbb{R}$  la imagen en escala de grises obtenida por una transformación estándar (por ejemplo, combinación ponderada de canales RGB):

$$G = \text{rgb2gray}(I_s).$$

En la implementación, esta operación se realiza con funciones MATLAB estándar [3].

### 5.3.4. Binarización tipo Otsu

**Umbral óptimo.** Sea  $t$  un umbral candidato. El criterio de Otsu selecciona  $t^*$  que maximiza la varianza entre clases  $\sigma_B^2(t)$  [8]:

$$t^* = \arg \max_t \sigma_B^2(t).$$

**Máscara binaria.** Con  $t^*$  se define una máscara binaria  $B : \Omega \rightarrow \{0, 1\}$ :

$$B(x) = \mathbf{1}\{G(x) \geq t^*\}.$$

En MATLAB esto corresponde a `graythresh` y binarización (p. ej. `imbinarize`) [3], [9].

### 5.3.5. Depuración por borde

**Objetos conectados al borde.** Sea  $\partial\Omega$  el conjunto de píxeles del borde de la imagen. Se elimina todo objeto conectado (en conectividad 4 u 8) a  $\partial\Omega$ , produciendo  $B_1$ :

$$B_1 = \text{clearborder}(B),$$

operación implementada por `imclearborder` [3].

### 5.3.6. Filtrado por área mínima

**Área mínima.** Sea  $A_{\text{mín}}$  un umbral de área (en píxeles) controlado por el usuario mediante interfaz [2]. Sea  $\mathcal{O}(B_1)$  el conjunto de componentes conectados en  $B_1$ . Se retienen solo objetos con área suficiente:

$$B_2 = \bigcup_{O \in \mathcal{O}(B_1): |O| \geq A_{\text{mín}}} O,$$

donde  $|O|$  denota el número de píxeles del objeto (área discreta).

### 5.3.7. Etiquetado de objetos

**Etiquetas.** Sea  $L : \Omega \rightarrow \{0, 1, \dots, N\}$  la imagen etiquetada obtenida de  $B_2$ , donde  $L(x) = i$  si  $x \in O_i$  y  $L(x) = 0$  si es fondo:

$$L = \text{bwlabel}(B_2).$$

Esto permite asociar cada píxel a un objeto y preparar la extracción de propiedades [3].

### 5.3.8. Extracción de características (regionprops)

**Propiedades por objeto.** Para cada objeto  $O_i$ , se calculan propiedades geométricas con `regionprops`, que retorna área, perímetro, ejes principales, orientación, excentricidad y diámetro equivalente [3]. Formalmente, ExtraGEM produce  $\mathbf{f}(O_i)$  y los estadísticos agregados definidos en la sección anterior.

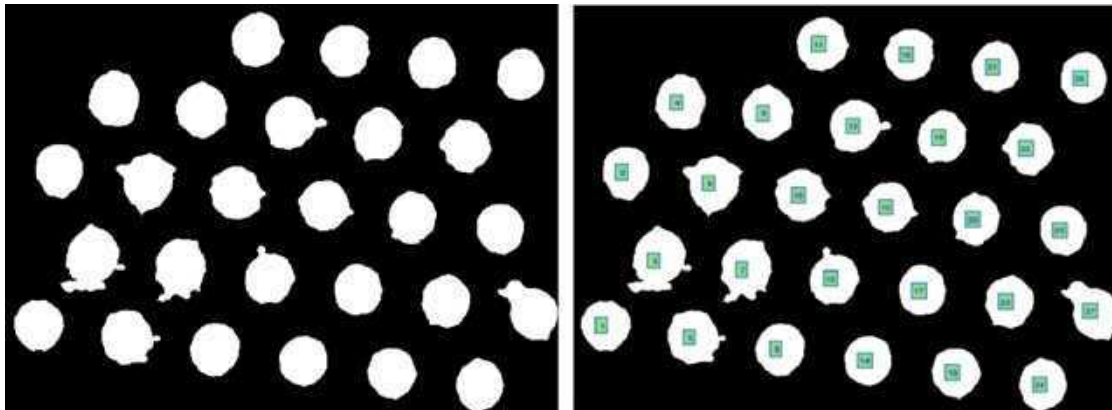
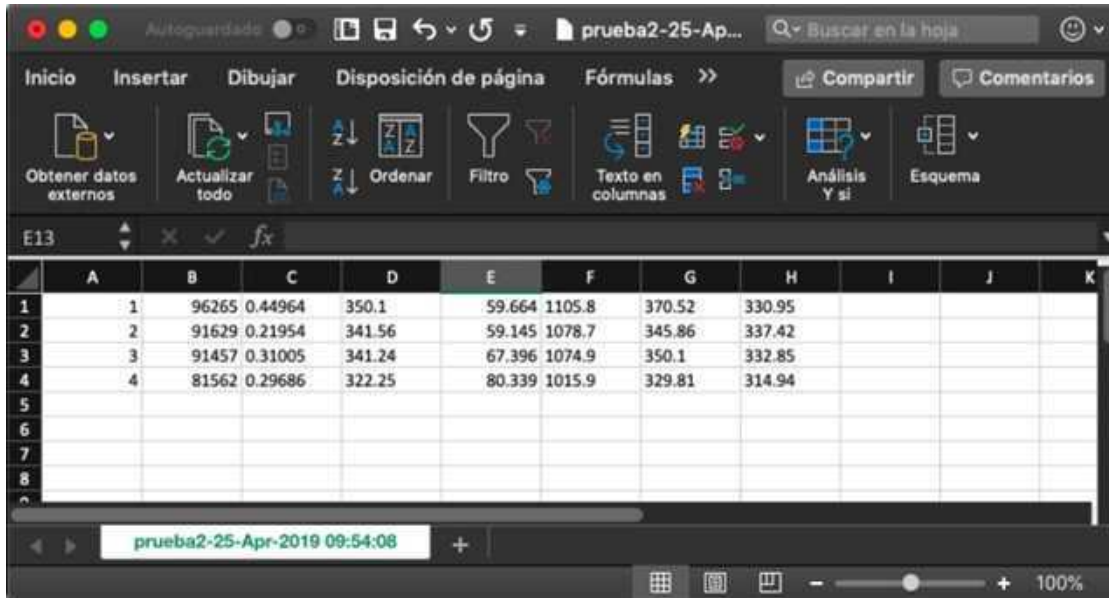


Figura 5.3: (A) Imagen binaria depurada; (B) objetos etiquetados para auditoría visual del conteo.

### 5.3.9. Exportación y trazabilidad

**Archivo de salida.** Sea  $\mathcal{T}$  una tabla con filas indexadas por objetos  $i = 1, \dots, N$  y columnas con características. ExtraGEM exporta  $\mathcal{T}$  a formato CSV para análisis en Excel u otros entornos [2]. Además, permite almacenar una copia de la imagen original y/o procesada con nombre/fecha para trazabilidad [2].

## 5 Extracción automática de características geométricas de microperforaciones en placas GEM



	A	B	C	D	E	F	G	H	I	J	K
1	1	96265	0.44964	350.1	59.664	1105.8	370.52	330.95			
2	2	91629	0.21954	341.56	59.145	1078.7	345.86	337.42			
3	3	91457	0.31005	341.24	67.396	1074.9	350.1	332.85			
4	4	81562	0.29686	322.25	80.339	1015.9	329.81	314.94			
5											
6											
7											
8											

Figura 5.4: Salida: panel de estadísticos (min, max, media, varianza) y ejemplo de archivo CSV.

### 5.3.10. Implementación y extensibilidad

**Arquitectura.** El manual técnico describe la organización del software sobre MATLAB GUIDE (interfaz `.fig` y callbacks), incluyendo funciones para carga, extracción, exportación, eliminación de objetos y ajuste de umbral [3]. Esta estructura facilita la incorporación de nuevas métricas (p. ej. circularidad  $4\pi A/P^2$ , rugosidad de borde, distribución espacial) sin rediseñar la aplicación.

## 5.4. Resultados y validación (estructura formal)

### 5.4.1. Comparación de mediciones (manual vs automática)

**Mediciones de referencia.** Sea  $d_i^{(m)}$  una medición manual asistida del diámetro equivalente del objeto  $i$  (referencia), y  $d_i^{(a)}$  la medición automática entregada por ExtraGEM. Se definen métricas de diferencia:

**Error absoluto medio (MAE).**

$$\text{MAE}(d) = \frac{1}{N} \sum_{i=1}^N |d_i^{(a)} - d_i^{(m)}|.$$

**Sesgo (Bias).**

$$\text{Bias}(d) = \frac{1}{N} \sum_{i=1}^N (d_i^{(a)} - d_i^{(m)}).$$

Se pueden definir métricas análogas para  $A, e, P, a, b$ .

### 5.4.2. Comparación entre placas o proveedores

**Distribuciones comparativas.** Dado un conjunto de placas  $p \in \{1, \dots, P\}$  y una característica  $f_k$ , se evalúan distribuciones  $p(f_k)$  mediante histogramas, cuantiles y pruebas de diferencia entre grupos (p. ej. placas radiadas vs. no radiadas) [1].

### 5.4.3. Respaldo institucional

La certificación institucional establece que GFIF y GEPAR cuentan con licencia de uso del software Extra\_GEM otorgada por la Universidad de Antioquia, lo cual respalda su despliegue como herramienta de laboratorio [7]. El software está protegido mediante registro ante la DNDA (Libro-Tomo-Partida 13-76-12, 27-sep-2019) con titularidad patrimonial de la Universidad de Antioquia [6].

## 5.5. Discusión

El aporte de ExtraGEM es integrar en una sola aplicación un flujo de inspección geométrica que, de otro modo, requeriría herramientas dispersas y operaciones manuales propensas a error, especialmente cuando se busca estadística sobre grandes poblaciones de perforaciones [1]. La combinación de umbralado tipo Otsu con depuración por borde y área mínima refleja un compromiso entre simplicidad y control operacional: el usuario puede ajustar umbral y eliminar objetos para corregir casos límite (sombras, partículas, artefactos) preservando trazabilidad [2].

Como limitación, el desempeño depende de la calidad de imagen (iluminación, enfoque, resolución). Una línea futura natural es incorporar corrección de iluminación no uniforme, segmentación adaptativa o aprendizaje supervisado para escenarios de baja separabilidad, manteniendo interpretabilidad y reporte estadístico [9].

## 5.6. Conclusiones

Se presentó ExtraGEM como herramienta de visión artificial para extraer automáticamente características geométricas de microperforaciones en placas GEM. El método define explícitamente la imagen de entrada, los objetos (perforaciones), las características por objeto y los estadísticos agregados; luego implementa un pipeline reproducible de filtrado, binarización tipo Otsu, depuración por borde y área mínima, etiquetado y extracción de propiedades con `regionprops`, complementado por exportación a CSV [2], [3], [8]. El software cuenta con registro DNDA y soporte institucional de uso en GFIF y GEPAR [6], [7]. En conjunto, el desarrollo habilita medición objetiva y escalable para análisis comparativos de calidad de placas asociadas a instrumentación para física de partículas [1], [4].

## Agradecimientos

A los grupos GFIF y GEPAR por la articulación del problema experimental con una solución computacional reproducible; y a la Universidad de Antioquia por el soporte institucional y el proceso de protección/registro del software [1], [6].

## Referencias

- [1] N. Vanegas Arbeláez y D. S. Fernández Mc Cann, *Solicitud para asignación de puntos por desarrollo de software: Extra\_GEM*, Formato institucional, Universidad de Antioquia, Fecha: 29-oct-2019, oct. de 2019.
- [2] *ExtraGEM: Extracción de características de placas GEM (Gas Electron Multiplier), Manual de Usuario*, Desarrollado en MATLAB R2018a, Universidad de Antioquia (GFIF/GEPAR), 2019.
- [3] *ExtraGEM: Manual Técnico*, Documentación de código y extensibilidad en MATLAB GUIDE, Universidad de Antioquia (GFIF/GEPAR), 2019.
- [4] F. Sauli, “The Gas Electron Multiplier (GEM): Operating Principles and Applications,” *Nuclear Instruments and Methods in Physics Research Section A*, 2016, Citado en los manuales del proyecto; fecha divulgada 2015.
- [5] B. A. Rodríguez R. y J. E. Mahecha G., *¿UdeA en el epicentro de la física mundial? Nota sobre colaboración UdeA–CERN*, Documento interno/divulgativo, Universidad de Antioquia, feb. de 2017.

- [6] D. Fernández Mc Cann, N. Vanegas Arbeláez y T. Escobar Alzate, *EXTRAGEM: Certificado de Registro de Soporte Lógico (Software)*, Libro–Tomo–Partida 13-76-12, Dirección Nacional de Derecho de Autor (DNDA), Ministerio del Interior, Colombia, Fecha de registro: 27-sep-2019. Titular del derecho patrimonial: Universidad de Antioquia, sep. de 2019.
- [7] N. Vanegas Arbeláez y D. S. Fernández Mc Cann, *Certificación de uso/licencia del software Extra\_GEM por los grupos GFIF y GEPAR*, Anexo institucional, Universidad de Antioquia, Fecha: 29-oct-2019, oct. de 2019.
- [8] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, n.º 1, págs. 62-66, 1979.
- [9] R. C. Gonzalez, R. E. Woods y B. R. Masters, “Digital Image Processing, Third Edition,” *Journal of Biomedical Optics*, vol. 14, n.º 2, pág. 029 901, 2009.

# Capítulo 6

## Estabilización digital de videos de *Daphnia pulex* para análisis fisiológico *in vivo*

### Autores

Santiago Ruiz González<sup>1</sup>; Maycol Esteban Zuluaga Montoya<sup>2</sup>; Yarin Tatiana Puerta Quintana<sup>3</sup>; Fabio Vélez Macías<sup>4</sup>; Néstor Jaime Aguirre Ramírez<sup>5</sup>; David Fernández McCann<sup>6</sup>.

Grupos Geolimna y Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

*Daphnia pulex* es un microcrustáceo ampliamente utilizado como bioindicador en estudios de calidad de agua y biotoxicidad, donde respuestas fisiológicas como la frecuencia cardiaca se cuantifican a partir de observación bajo microscopio y grabación de video [1], [2], [3],

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. santiago.ruiz@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. maycol.zuluaga@udea.edu.co

<sup>3</sup>Estudiante de maestría, Ingeniería Ambiental. tatiana.puerta@udea.edu.co

<sup>4</sup>Profesor, Departamento de Ingeniería Ambiental y Sanitaria. fabio.velez@udea.edu.co

<sup>5</sup>Profesor, Departamento de Ingeniería Ambiental y Sanitaria. nestor.aguirre@udea.edu.co

<sup>6</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

[4]. Sin embargo, la alta motilidad del organismo introduce movimientos de traslación y rotación que dificultan el seguimiento de regiones internas (p.ej. el corazón) y obligan frecuentemente a técnicas de inmovilización física o química que pueden inducir estrés adicional y sesgar la respuesta medida [1], [4]. En este capítulo se presenta *ESVIDA*, un software registrado ante la DNDA que implementa un método de estabilización digital de video, compensando traslación y rotación del organismo fotograma a fotograma para generar un video estabilizado en el que *D. pulex* se percibe en posición fija [2], [5]. El método combina (i) segmentación de la región de interés mediante contraste en el canal *L* del espacio CIELab, (ii) estimación de centroide y orientación a partir de la elipse de mínima área que encierra el contorno, (iii) validación del movimiento mediante flujo óptico y (iv) aplicación de una transformación afín para alinear cada fotograma a una pose de referencia [2], [4]. Se reporta reducción significativa del movimiento observado y diferencias estadísticamente significativas entre coordenadas de una región de observación antes y después de la estabilización [4].

**Palabras clave:** bioindicador; biotoxicidad; estabilización de video; compensación de movimiento; CIELab; flujo óptico; elipse equivalente; transformación afín; *ESVIDA*; *Daphnia pulex*.

## 6.1. Introducción

En ecología acuática y ciencias ambientales, la visión por computador se ha consolidado como herramienta para cuantificar dinámicas biológicas a partir de imágenes y videos, especialmente cuando se requieren mediciones repetibles, escalables y menos dependientes del observador [4], [6]. En el caso de *Daphnia pulex*, su sensibilidad a cambios ambientales la convierte en modelo de referencia para evaluar contaminantes, toxicidad y efectos de sustancias; la frecuencia cardiaca bajo microscopio constituye un indicador fisiológico de uso frecuente [1], [2].

El obstáculo técnico principal es que el organismo vivo no permanece estático durante la grabación: exhibe movimientos globales de traslación y rotación, además de movimientos locales de apéndices y órganos. Para análisis automático de regiones internas (p.ej. conteo de latidos), resulta crítico estabilizar el movimiento global para permitir seguimiento consistente de un ROI sin recurrir a inmovilización *in vivo* que puede agregar estrés y sesgo fisiológico [1], [4].



Figura 6.1: Anatomía de *Daphnia pulex* y ejemplos de regiones internas de interés fisiológico (p. ej. corazón).

## 6.2. Planteamiento del problema (formulación formal)

### 6.2.1. Video, fotogramas y objetivo de estabilización

**Definición del dato.** Sea  $v$  un video compuesto por  $n$  fotogramas. Denotamos por  $t_i$  el tiempo (o índice) del fotograma  $i$ , con  $i \in \{1, \dots, n\}$ . Cada fotograma es una imagen discreta

$$f_i : \Omega \rightarrow \mathbb{R}^3,$$

donde  $\Omega \subset \mathbb{Z}^2$  es el dominio de píxeles y  $f_i(x, y)$  es el vector RGB en la coordenada  $(x, y)$  [4].

**Objetivo.** El objetivo de estabilización es construir un nuevo video  $v'$  con fotogramas  $\{g_i\}_{i=1}^n$  tales que el organismo quede alineado a una pose de referencia (por ejemplo la del primer fotograma). En términos de parámetros de pose, se busca que cada fotograma  $i$  cumpla:

$$(x_{c,i}, y_{c,i}, \alpha_i) \rightarrow (x_{c,1}, y_{c,1}, \alpha_1),$$

donde  $(x_{c,i}, y_{c,i})$  es el centroide estimado del organismo en el fotograma  $i$  y  $\alpha_i$  su orientación global [4].

### 6.2.2. Restricción: distinguir movimiento global vs movimiento interno

**Motivación.** El organismo tiene movimiento interno (apéndices/órganos) que no se desea “cancelar” porque puede ser la señal de interés. Se requiere, por tanto, compensar principalmente el movimiento global (traslación/rotación) y no el movimiento local. Para ello se incorpora una medida de significancia del desplazamiento usando flujo óptico entre fotogramas consecutivos [2], [4].

## 6.3. Metodología (formulación formal y pipeline)

### 6.3.1. Visión general del pipeline

El método implementado en ESVIDA sigue la secuencia: (i) segmentación de ROI del organismo, (ii) estimación de parámetros geométricos (centroide y orientación), (iii) estimación de desplazamiento por puntos rastreados (flujo óptico), (iv) decisión de compensación si el movimiento es significativo, y (v) transformación afín (rotación + traslación) para producir el fotograma estabilizado  $g_i$  [2], [4].

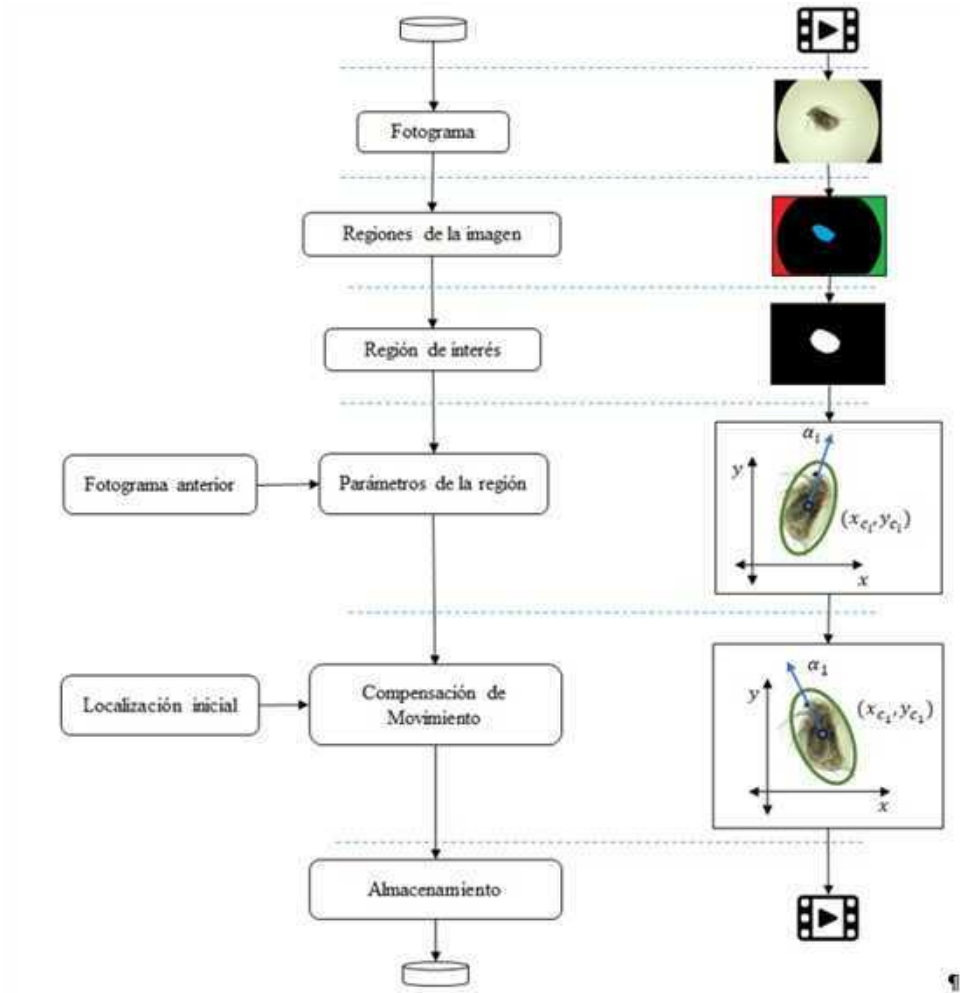


Figura 6.2: Pipeline ESVIDA: video  $\rightarrow$  ROI  $\rightarrow$  parámetros  $(x_{c,i}, y_{c,i}, \alpha_i)$  y desplazamientos  $\{d_k\}$   $\rightarrow$  transformación afín  $\rightarrow$  video estabilizado.

### 6.3.2. Del fotograma a la región de interés

**Conversión a CIELab y selección del canal  $L$ .** Sea  $f_i(x, y)$  el fotograma RGB. Se transforma a CIELab y se selecciona la componente  $L$  por ofrecer mayor contraste entre organismo y fondo. Denotamos por  $L_i : \Omega \rightarrow [0, 255]$  dicha componente [4].

**Ecualización de histograma.** Sea  $L_i$  la imagen de luminancia del fotograma. Se obtiene una versión ecualizada  $L'_i$  aplicando una transformación monótonica  $T(\cdot)$  construida a partir de la distribución acumulada del histograma, para mejorar contraste [4].

**Filtrado espacial.** Sea  $L'_i$  la luminancia ecualizada. Se define una imagen suavizada  $L''_i$  aplicando un filtro promedio  $3 \times 3$ . Si  $(x, y)$  es un píxel y  $(q, w) \in \{-1, 0, 1\}^2$  recorre la

vecindad, entonces el valor filtrado se calcula como:

$$L_i''(x, y) = \frac{1}{9} \sum_{q=-1}^1 \sum_{w=-1}^1 L_i'(x + q, y + w),$$

donde la suma promedia la vecindad local [4].

**Binarización por umbral adaptativo.** Sea  $m(x, y)$  el promedio local de intensidades de  $L_i''$  en una ventana cuadrada de tamaño  $n$  centrada en  $(x, y)$ . Se define un rango de aceptación  $[\mu_m, \mu_M]$  alrededor de  $m(x, y)$  (por ejemplo, un margen porcentual). La imagen binaria  $B_i$  se define como:

$$B_i(x, y) = \begin{cases} 1, & \mu_m(x, y) < L_i''(x, y) < \mu_M(x, y), \\ 0, & \text{en otro caso,} \end{cases}$$

con  $n$  determinado experimentalmente [4].

**Selección de la ROI.** Sea  $\{r_{i,j}\}_{j=1}^J$  el conjunto de regiones conectadas extraídas de  $B_i$ . Se selecciona  $r_{i,1}$  como la región asociada al organismo (por tamaño/criterios de región), y se toma como ROI del fotograma  $i$  [4].

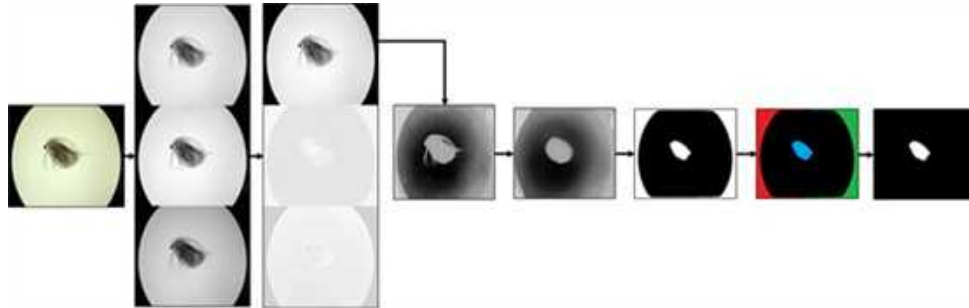


Figura 6.3: Secuencia de obtención de ROI: fotograma RGB  $\rightarrow$  canal  $L$  (CIELab)  $\rightarrow$  ecualización  $\rightarrow$  filtrado  $\rightarrow$  binarización adaptativa  $\rightarrow$  región de interés  $r_{i,1}$ .

### 6.3.3. Parámetros geométricos: centroide, orientación y elipse equivalente

**Conjunto de píxeles de la región.** Sea  $r_{i,1}$  la región conectada que representa al organismo en el fotograma  $i$ . Denotamos por  $\{(x_j, y_j)\}_{j=1}^{N_i}$  el conjunto de coordenadas de sus  $N_i$  píxeles.

**Centroide.** El centroide  $(x_{c,i}, y_{c,i})$  se define como el promedio de coordenadas:

$$x_{c,i} = \frac{1}{N_i} \sum_{j=1}^{N_i} x_j, \quad y_{c,i} = \frac{1}{N_i} \sum_{j=1}^{N_i} y_j,$$

lo cual aproxima el “centro de masa” discreto de la región [4].

**Momentos de inercia y orientación.** Sea  $(x_{c,i}, y_{c,i})$  el centroide. Se definen los momentos discretos:

$$I_x = \sum_{j=1}^{N_i} (y_j - y_{c,i})^2, \quad I_y = \sum_{j=1}^{N_i} (x_j - x_{c,i})^2, \quad I_{xy} = \sum_{j=1}^{N_i} (x_j - x_{c,i})(y_j - y_{c,i}),$$

y la orientación global  $\alpha_i$  se calcula como:

$$\alpha_i = \frac{1}{2} \arctan\left(\frac{-2I_{xy}}{I_x - I_y}\right),$$

que corresponde a la orientación del eje principal de la región [4].

**Ejes de la elipse equivalente.** Se define la matriz de inercia

$$\mathbf{I} = \begin{bmatrix} I_x & I_{xy} \\ I_{xy} & I_y \end{bmatrix},$$

y sus valores propios  $\lambda_1 \geq \lambda_2$ . Sea  $A_i$  el área de la región (número de píxeles),  $A_i = N_i$ . Los ejes mayor y menor equivalentes se derivan (según el modelo del manuscrito) como:

$$r_a = 2\sqrt{\frac{\lambda_1}{A_i}}, \quad r_b = 2\sqrt{\frac{\lambda_2}{A_i}},$$

lo que permite representar una elipse de orientación  $\alpha_i$  y centro  $(x_{c,i}, y_{c,i})$  para auditoría visual [2], [4].

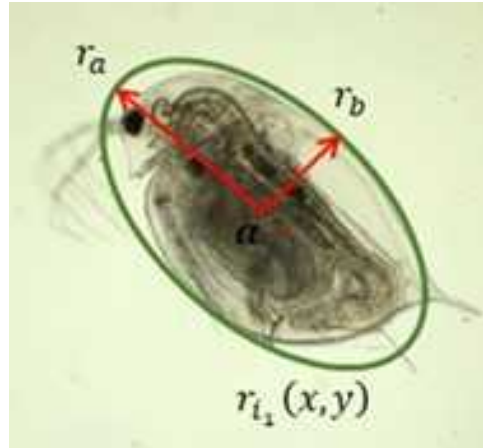


Figura 6.4: Elipse equivalente: centroide  $(x_{c,i}, y_{c,i})$ , orientación  $\alpha_i$  y ejes  $(r_a, r_b)$  sobre la ROI del organismo.

### 6.3.4. Estimación del movimiento mediante puntos rastreados

**Puntos rastreables.** Sea  $f_{i-1}$  el fotograma anterior. Se selecciona un conjunto de  $k$  puntos de interés

$$\mathcal{P}_{i-1} = \{(x_1, y_1), \dots, (x_k, y_k)\},$$

usando el criterio “good features to track” [4].

**Correspondencias por Lucas–Kanade.** Sea  $f_i$  el fotograma actual. Se estiman los puntos correspondientes

$$\mathcal{Q}_i = \{(u_1, v_1), \dots, (u_k, v_k)\},$$

mediante el método de Lucas–Kanade, interpretado como flujo óptico local [2], [4].

**Desplazamientos.** Para cada punto  $m \in \{1, \dots, k\}$ , el desplazamiento euclídeo se define como:

$$d_m = \sqrt{(u_m - x_m)^2 + (v_m - y_m)^2}.$$

Estos valores resumen magnitud de movimiento entre fotogramas [4].

**Decisión de movimiento significativo.** Sea  $F_{\text{máx}}$  la moda (o valor de mayor frecuencia) del histograma de  $\{d_m\}_{m=1}^k$  y sea  $\beta$  un umbral experimental. Se decide aplicar compensación si:

$$F_{\text{máx}} > \beta,$$

evitando correcciones cuando el desplazamiento no es representativo [4].



Figura 6.5: Puntos rastreados y vectores de desplazamiento  $\{d_m\}$  superpuestos sobre el organismo para estimar movimiento global.

### 6.3.5. Compensación: transformación afín fotograma a fotograma

**Pose objetivo.** Sea la pose de referencia la del primer fotograma:  $(x_{c,1}, y_{c,1}, \alpha_1)$ . Para el fotograma  $i$ , se estima la pose  $(x_{c,i}, y_{c,i}, \alpha_i)$ .

**Transformación.** Se aplica una transformación afín  $T_i$  compuesta por una rotación de ángulo  $(\alpha_1 - \alpha_i)$  y una traslación que alinea centroides. El fotograma estabilizado se define como:

$$g_i = T_i(f_i),$$

de forma que el organismo se percibe alineado a la referencia [4].

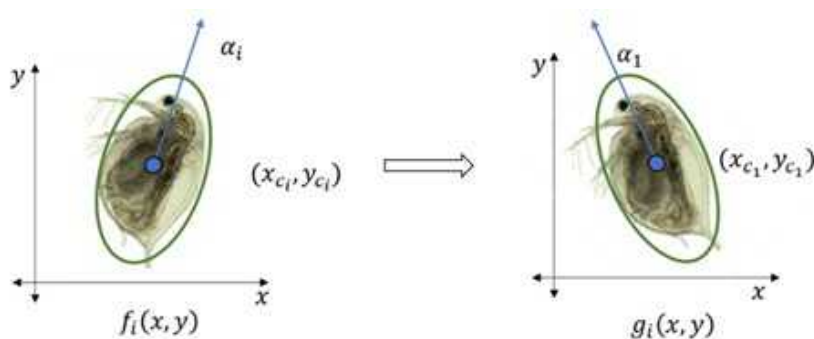


Figura 6.6: Compensación: el fotograma  $f_i$  se rota y traslada para alinear  $(x_{c,i}, y_{c,i}, \alpha_i)$  con  $(x_{c,1}, y_{c,1}, \alpha_1)$ , produciendo  $g_i$ .

## 6.4. Implementación software (ESVIDA)

ESVIDA fue implementado en Python 3.7 con OpenCV 4.1.2 y una interfaz gráfica desarrollada con Qt Designer y PyQt5 [2]. El manual técnico describe el núcleo funcional como el método `Estab_Daphnia()`, encargado de (i) separar el organismo del fondo, (ii) estimar la elipse mínima y (iii) cuantificar rotación/traslación con respecto a una posición inicial, además de validar el movimiento con flujo óptico [2]. El manual de usuario describe el flujo operativo: importar video, iniciar ejecución, detener, y almacenamiento automático del video estabilizado y un video de comparación [3].



Figura 6.7: Interfaz del software ESVIDA: (izquierda) fotograma original con elipse y puntos de movimiento; (derecha) fotograma estabilizado.

## 6.5. Resultados (evidencias y evaluación)

### 6.5.1. Análisis secuencial fotograma a fotograma

El manuscrito reporta la comparación visual de secuencias consecutivas: en el video original un punto de referencia no permanece asociado a la misma región del organismo; tras la estabilización, el punto se mantiene consistente, evidenciando un efecto de “inmovilización digital” [4].

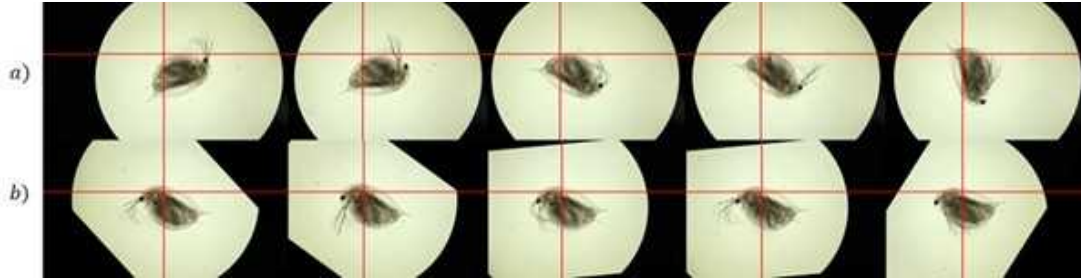


Figura 6.8: Secuencia de fotogramas: (A) original con traslación/rotación; (B) estabilizado con consistencia espacial del punto de referencia.

### 6.5.2. Análisis espacial y contraste estadístico

**Región de observación.** Sea una región fija en el plano imagen (plantilla) seleccionada para seguimiento. Denotamos por  $(c_x(i), c_y(i))$  la posición estimada en el fotograma  $i$  en el video original, y por  $(c'_x(i), c'_y(i))$  la posición en el video estabilizado, estimadas mediante *template matching* [4].

**Comparación.** El manuscrito reporta diferencias significativas (T-test,  $p < 0,05$ ) entre coordenadas antes y después de estabilizar, así como reducción marcada de dispersión de las posiciones en el video estabilizado [4].

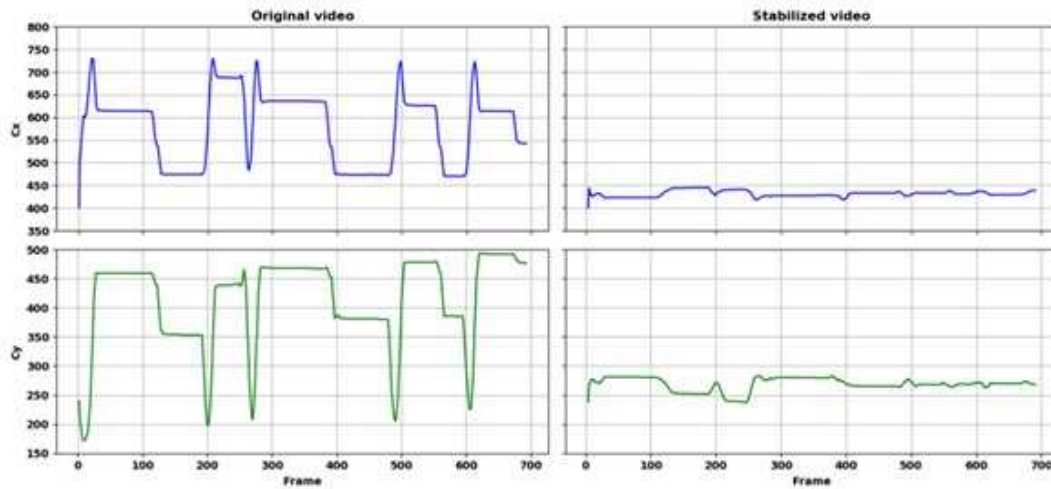


Figura 6.9: Evolución de  $c_x(i), c_y(i)$  vs.  $c'_x(i), c'_y(i)$  en el tiempo (fotogramas), mostrando estabilidad en el video procesado.

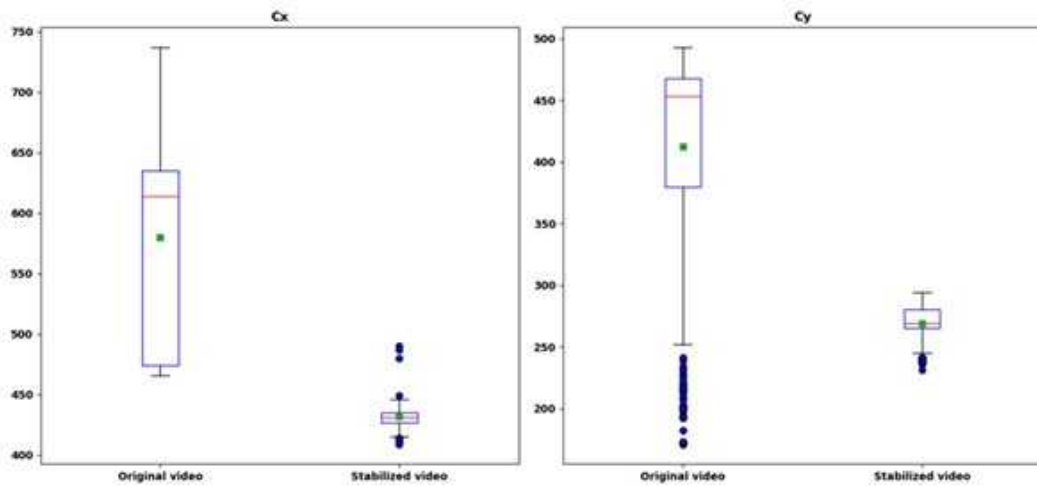


Figura 6.10: Diagramas de cajas para coordenadas en el video original y estabilizado, mostrando reducción de dispersión tras la compensación.

## 6.6. Discusión

El aporte metodológico de ESVIDA es trasladar la “fijación” desde mecanismos físicos/químicos hacia una fijación digital del movimiento global, lo que reduce interferencias experimentales asociadas a estrés por manipulación y permite análisis más limpio de variables fisiológicas bajo microscopio [1], [4]. La combinación ROI–elipse–flujo óptico aporta interpretabilidad: (i) la ROI delimita el organismo, (ii) la elipse resume pose global, y (iii) el flujo óptico valida si procede compensar. Esto diferencia el problema de estabilización aquí abordado (objeto vivo en movimiento) de la estabilización clásica orientada a vibración de cámara [4].

Como limitaciones, el método depende del contraste en el canal  $L$  y de la calidad del video (iluminación, enfoque, saturación). En condiciones de bajo contraste o alto ruido, pueden requerirse mejoras como corrección de iluminación, segmentación adaptativa más robusta o modelos supervisados para ROI, manteniendo la trazabilidad del pipeline [2].

## 6.7. Conclusiones

Se presentó ESVIDA como un desarrollo de software registrado ante la DNDA que implementa un método de estabilización de videos de *Daphnia pulex* mediante compensación de traslación y rotación por fotograma. El método define y estima explícitamente la ROI, el centroide y la orientación del organismo a partir de una elipse equivalente, valida la

magnitud de movimiento mediante puntos rastreados (flujo óptico) y aplica una transformación afín para alinear cada fotograma a una pose de referencia, produciendo un video estabilizado [2], [4]. Este enfoque habilita análisis fisiológico más preciso (p.ej. conteo cardiaco) sin recurrir a inmovilización *in vivo*, reduciendo potenciales sesgos por estrés experimental [1]. El software cuenta con certificación de uso en los grupos Geolimna y GEPAR [7] y registro de propiedad intelectual con titularidad patrimonial institucional [5].

## Agradecimientos

A los grupos Geolimna y GEPAR por la articulación del problema ambiental con el desarrollo de una solución reproducible de visión artificial; y al soporte institucional de la Universidad de Antioquia para la protección y uso del software [1], [7].

## Referencias

- [1] D. S. Fernández Mc Cann, F. d. J. Vélez Macías y N. J. Aguirre Ramírez, *Solicitud para asignación de puntos por desarrollo de software: ESVIDA (Software para la estabilización de videos de Daphnia pulex)*, Formato institucional, Universidad de Antioquia, Fecha: 24-sep-2020. Incluye motivación en biotoxicidad y registro DNDA 13-81-196, sep. de 2020.
- [2] *ESVIDA: Manual Técnico. Software para la estabilización de videos de la Daphnia pulex*, Grupos Gepar–GeoLimna, Universidad de Antioquia, 2020.
- [3] *ESVIDA: Manual de Usuario. Software para la estabilización de videos de la Daphnia pulex*, Grupos Gepar–GeoLimna, Universidad de Antioquia, 2020.
- [4] S. Ruiz, M. Zuluaga, Y. T. Puerta, D. Fernández, F. Vélez y N. Aguirre, *Método de compensación del movimiento aplicado a videos de Daphnia pulex observada al microscopio*, Manuscrito / paper interno (PDF), Insumo: 1\_Manuscrito\_ESVIDA.pdf, 2020.
- [5] D. Fernández Mc Cann, N. J. Aguirre Ramírez, F. Vélez Macías, S. Ruiz González, M. E. Zuluaga Montoya e Y. T. Puerta Quintana, *ESVIDA: Software para la estabilización de videos de la Daphnia pulex, Certificado de Registro de Soporte Lógico (Software), Libro–Tomo–Partida 13-81-196*, Dirección Nacional de Derecho de Autor (DNDA), Ministerio del Interior, Colombia, Fecha de registro: 11-sep-2020. Titular del derecho patrimonial: Universidad de Antioquia, sep. de 2020.

- [6] S. Ruiz, M. Zuluaga, Y. T. Puerta, D. Fernández, F. Vélez y N. Aguirre, *Método de compensación del movimiento para videos de Daphnia pulex observadas al microscopio*, Documento interno (DOCX), Insumo: 0\_Paper\_Metodo estabilizacion daphnia.docx, 2020.
- [7] F. d. J. Vélez Macías y D. S. Fernández Mc Cann, *Certificación de la Unidad Académica sobre el uso del software ESVIDA (licencia de uso para GEOLIMNA y GEPAR)*, Anexo institucional, Universidad de Antioquia, Medellín, 24-sep-2020, sep. de 2020.

# Capítulo 7

## Estimación de la frecuencia cardiaca de *Daphnia pulex* mediante fotocardiograma

### Autores

Yarin Tatiana Puerta Quintana<sup>1</sup>; Fabio de Jesús Vélez Macías<sup>2</sup>; Néstor Jaime Aguirre Ramírez<sup>3</sup>; David Fernández Mc Cann<sup>4</sup>.

Grupos Geolimna y Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

*Daphnia pulex* es un microcrustáceo usado como bioindicador en análisis de calidad de agua y biotoxicidad, donde la frecuencia cardiaca constituye un biomarcador fisiológico sensible a estresores ambientales [1], [2]. El conteo visual de latidos, además de demandar alta atención, es vulnerable a error humano cuando la tasa de eventos supera varios latidos por segundo; adicionalmente, el movimiento traslacional y rotacional del individuo durante la observación introduce variaciones espurias de iluminación y de encuadre que degradan la confiabilidad de mediciones manuales [1], [3]. En este capítulo se presenta *FECAD*, un software que estima automáticamente la frecuencia cardiaca a partir de un video corto

---

<sup>1</sup>Estudiante de maestría, Ingeniería Ambiental. [tatiana.puerta@udea.edu.co](mailto:tatiana.puerta@udea.edu.co)

<sup>2</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. [fabio.velez@udea.edu.co](mailto:fabio.velez@udea.edu.co)

<sup>3</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. [nestor.aguirre@udea.edu.co](mailto:nestor.aguirre@udea.edu.co)

<sup>4</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. [david.fernandez@udea.edu.co](mailto:david.fernandez@udea.edu.co)

capturado con microscopio invertido, mediante un enfoque denominado *fotocardiograma digital*: (i) compensación de movimiento con descriptores SURF, (ii) preprocesamiento para invariancia parcial a iluminación (filtro gaussiano, operaciones morfológicas y umbral adaptativo), (iii) extracción de una señal escalar de intensidad en una sección fija que intersecta la región cardiaca, y (iv) detección de cambios de pendiente para contar pulsaciones en una ventana temporal y convertirlas a latidos por minuto (bpm) [1], [4]. En condiciones de laboratorio se reporta una medición promedio de 480 bpm y un error menor al 1 % respecto a conteo experto en el escenario evaluado [1]. El desarrollo se acompaña de manual técnico, manual de usuario, certificación de uso institucional y registro de propiedad intelectual reportado en la solicitud [3], [4], [5], [6].

**Palabras clave:** biomarcador; ecotoxicología; *Daphnia pulex*; visión artificial; fotocardiograma; SURF; umbral adaptativo; morfología matemática; FECAD.

## 7.1. Introducción

La evaluación de efectos subletales mediante biomarcadores se ha consolidado como complemento a *endpoints* tradicionales (mortalidad/movilidad) en ensayos de toxicidad, al permitir inferir estrés fisiológico sin requerir necesariamente muerte del organismo [1]. En organismos traslúcidos como *D. pulex*, el corazón es observable al microscopio invertido y su dinámica (sístole/diástole) puede cuantificarse para estimar respuesta a fármacos y contaminantes [1].

El reto operativo surge por dos fuentes de error: (i) la alta tasa de eventos por unidad de tiempo, que puede exigir conteos del orden de 10 eventos/s, y (ii) la motilidad del individuo, que introduce desplazamientos entre fotogramas y variaciones de iluminación debidas a cambios geométricos entre la fuente y el lente [1], [2]. FECAD aborda estos retos mediante un flujo automático y trazable de procesamiento de video y extracción de señal.



Figura 7.1: Fenotipo de *Daphnia pulex* observado al microscopio; se resalta la zona donde se ubica el corazón para el análisis de frecuencia cardíaca.

## 7.2. Planteamiento del problema (formulación formal)

### 7.2.1. Datos de entrada: video y fotogramas

**Definición del video.** Sea  $V$  un video digital a color. Denotamos por  $(x, y)$  la coordenada discreta de píxel, con  $(x, y) \in \Omega \subset \mathbb{Z}^2$ , y por  $t$  el índice temporal (fotograma). El video se representa como una función

$$V : \Omega \times \{1, \dots, n\} \rightarrow [0, 255]^3,$$

donde  $n$  es el número total de fotogramas y  $V(x, y, t) = (r, g, b)$  corresponde a los valores RGB en el fotograma  $t$  [1].

**Fotograma.** Para cada tiempo discreto  $t_i$  (fotograma  $i$ ), definimos la imagen  $f_i(x, y) = V(x, y, t_i)$ .

### 7.2.2. Objetivo: pulsaciones por unidad de tiempo

**Observación.** Sea  $t_O$  el tiempo total de observación (duración efectiva del segmento de video procesado). Sea  $n_b$  el número de pulsaciones detectadas en el intervalo  $t_O$ .

**Frecuencia cardiaca.** Definimos la frecuencia cardiaca como

$$P_s(t_O) = \frac{n_b}{t_O},$$

donde  $P_s(t_O)$  se interpreta como pulsaciones por segundo (o por unidad de tiempo según unidades de  $t_O$ ). Para expresar en latidos por minuto (bpm), se usa  $\text{bpm} = 60 P_s(t_O)$  si  $t_O$  está en segundos [1].

### 7.2.3. Restricción: movimiento del individuo

**Necesidad de estabilización.** El individuo se desplaza durante la observación. Denotamos por  $\Delta f_i$  la diferencia entre fotogramas consecutivos  $f_{i+1} - f_i$ . Para medir latidos con estabilidad espacial se requiere minimizar, mediante un operador de compensación, la variabilidad de fondo y la traslación global del organismo, preservando principalmente los cambios locales asociados al ciclo cardiaco [1].

## 7.3. Metodología (formulación formal y pipeline)

### 7.3.1. Visión general

El pipeline de FECAD se estructura como: (i) selección de región/sección de análisis que atraviesa el corazón, (ii) compensación de movimiento por correspondencias de puntos SURF, (iii) preprocesamiento para homogeneidad de iluminación, (iv) extracción de una señal escalar de intensidad  $I(t)$  sobre una sección fija, y (v) detección de eventos (pulsaciones) a partir de cambios de pendiente de  $I(t)$  [1], [3], [4].

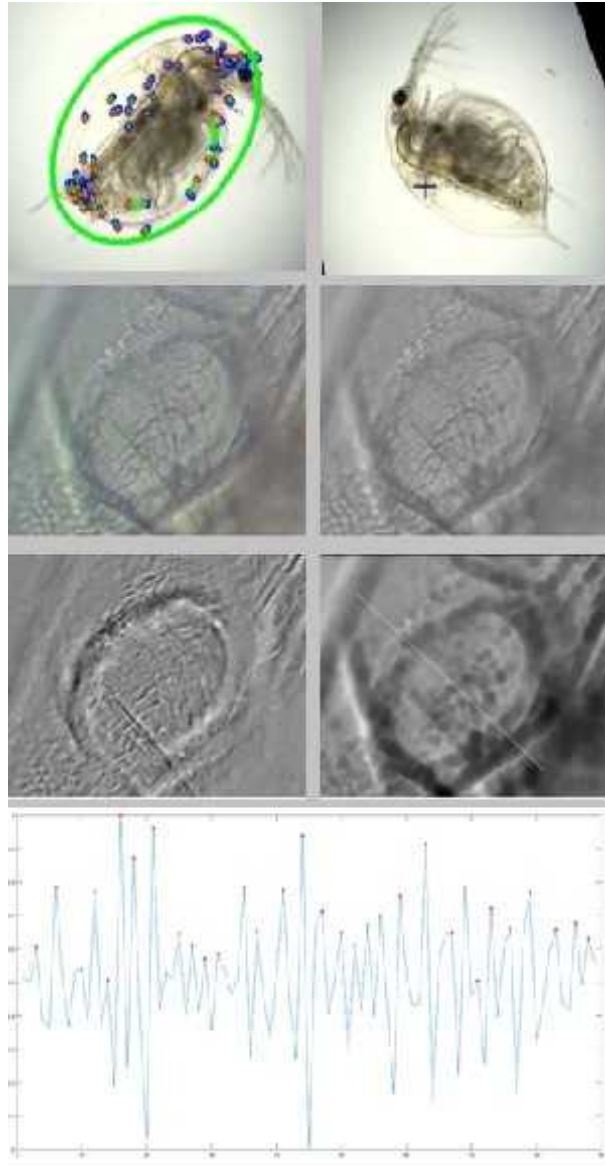


Figura 7.2: Pipeline de FECAD: video  $\rightarrow$  estabilización SURF  $\rightarrow$  preprocesamiento (gausiano+morfología+umbral adaptativo)  $\rightarrow$  señal  $I(t)$   $\rightarrow$  conteo de pulsaciones  $\rightarrow$  bpm.

### 7.3.2. Definición de la sección de análisis

**Sección del plano imagen.** Sea  $S$  una sección (máscara) del plano imagen que interseca la zona cardíaca. En la práctica,  $S$  se define a partir de dos puntos seleccionados por el usuario sobre el fotograma (una línea que pasa por el corazón), y se trabaja con su vecindad (ancho de línea) para robustez [3], [4].

**Máscara binaria.** Definimos  $M_S : \Omega \rightarrow \{0, 1\}$  como la máscara binaria de la sección,

donde  $M_S(x, y) = 1$  si  $(x, y)$  pertenece a la sección y 0 en caso contrario.

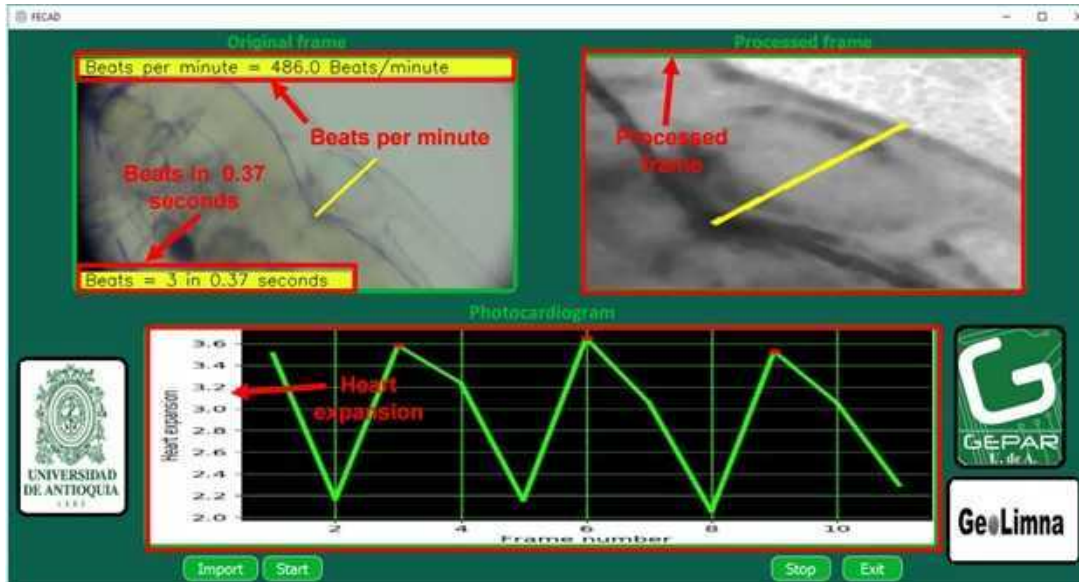


Figura 7.3: Definición de la sección  $S$ : el usuario marca dos puntos y se traza una línea que atraviesa el corazón; esta línea define la máscara  $M_S$  para extraer la señal de expansión/intensidad.

### 7.3.3. Compensación de movimiento mediante SURF

**Puntos de interés.** Sea  $\mathcal{P}_i = \{p_{i,1}, \dots, p_{i,k}\}$  el conjunto de  $k$  puntos de interés detectados por SURF en el fotograma  $f_i$ , donde cada punto  $p_{i,j} \in \mathbb{R}^2$  representa una coordenada de imagen [1].

**Correspondencias.** Sea  $\pi$  una función de emparejamiento que asocia puntos entre dos fotogramas consecutivos:  $\pi(p_{i,j}) = p_{i+1,j'}$ . Denotamos por  $\Delta p_{i,j} = \pi(p_{i,j}) - p_{i,j}$  el vector de desplazamiento del punto emparejado.

**Desplazamiento global.** Para compensar traslación dominante, se define un vector de desplazamiento global como el promedio robusto (por ejemplo, mediana componente a componente) de los  $\Delta p_{i,j}$ :

$$\Delta \bar{p}_i = \text{median}\{\Delta p_{i,j}\}_{j=1}^k.$$

Luego se genera el fotograma estabilizado  $\tilde{f}_{i+1}$  desplazando  $f_{i+1}$  por  $-\Delta \bar{p}_i$  [1].



Figura 7.4: Estabilización SURF: puntos detectados y emparejados entre  $f_i$  y  $f_{i+1}$ ; el vector global  $\Delta\bar{p}_i$  se usa para compensar el movimiento del organismo. (Ver presentación de Estabilización de Imagen Pulga de Agua -ESVIDA-)

### 7.3.4. Preprocesamiento para homogeneidad de iluminación

**Conversión a escala de grises.** Sea  $\tilde{f}_i$  el fotograma estabilizado. Definimos una imagen en grises  $G_i : \Omega \rightarrow \mathbb{R}$  como una combinación estándar de canales RGB. Sobre  $G_i$  se aplica el flujo de filtrado [1].

**Filtro gaussiano.** Sea  $\sigma$  el parámetro de dispersión del kernel. Definimos el kernel gaussiano

$$\mathcal{G}_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right).$$

La imagen suavizada se define como la convolución:

$$G_i^{(g)} = G_i * \mathcal{G}_\sigma,$$

lo cual reduce ruido de alta frecuencia y estabiliza intensidades locales [1].

**Erosión en escala de grises.** Sea  $B$  un elemento estructurante (por ejemplo, una matriz  $3 \times 3$  de unos). La erosión morfológica en grises se define como:

$$G_i^{(m)} = G_i^{(g)} \ominus B,$$

con el objetivo de aumentar contraste relativo entre regiones oscuras y claras en la zona

de interés [1].

**Umbral adaptativo.** Para compensar variaciones temporales de iluminación, se define una ventana local de tamaño  $n \times n$ . Sea  $\bar{m}(x, y)$  el promedio local de  $G_i^{(g)}$  en la ventana centrada en  $(x, y)$ . Se define un margen relativo  $\eta$  (por ejemplo 20%), y los umbrales:

$$u_M(x, y) = \bar{m}(x, y)(1 + \eta), \quad u_m(x, y) = \bar{m}(x, y)(1 - \eta).$$

Luego se genera una imagen binaria  $B_i$ :

$$B_i(x, y) = \begin{cases} 1, & u_m(x, y) < G_i^{(g)}(x, y) < u_M(x, y), \\ 0, & \text{en otro caso.} \end{cases}$$

Este esquema busca normalizar decisiones de segmentación a cambios globales de iluminación [1].

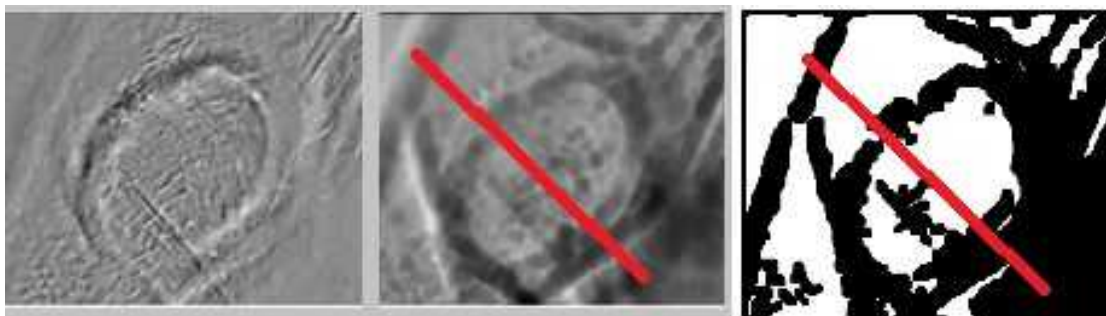


Figura 7.5: Preprocesamiento: (a) sección original, (b) grises, (c) gaussiano, (d) morfología y binarización adaptativa.

### 7.3.5. Extracción de la señal de fotocardiograma

**Señal escalar.** Sea  $I(t_i)$  la señal escalar asociada al fotograma  $i$ . Para definirla, primero restringimos el fotograma preprocesado a la sección  $S$  mediante la máscara  $M_S$ . Denotamos por  $H_i(x, y)$  una representación en grises (o intensidad) ya preprocesada. La señal se define como la suma de intensidades dentro de la sección:

$$I(t_i) = \sum_{(x,y) \in \Omega} H_i(x, y) M_S(x, y).$$

Esta suma actúa como “integral discreta” de la sección y varía con la expansión/contracción cardíaca [1].

### 7.3.6. Detección de pulsaciones

**Pendiente discreta.** Para dos tiempos consecutivos  $t_{i-1}$  y  $t_i$ , definimos la diferencia:

$$\Delta I_i = I(t_i) - I(t_{i-1}).$$

La señal de signo de pendiente se define como  $s_i = \text{sign}(\Delta I_i)$ .

**Evento de pulsación.** Se cuenta una pulsación cuando ocurre un cambio de signo de pendiente que corresponde al patrón de expansión y contracción (por ejemplo, de positiva a negativa en máximos locales). Denotamos por  $n_b$  el número total de eventos detectados en  $t_O$  [1].

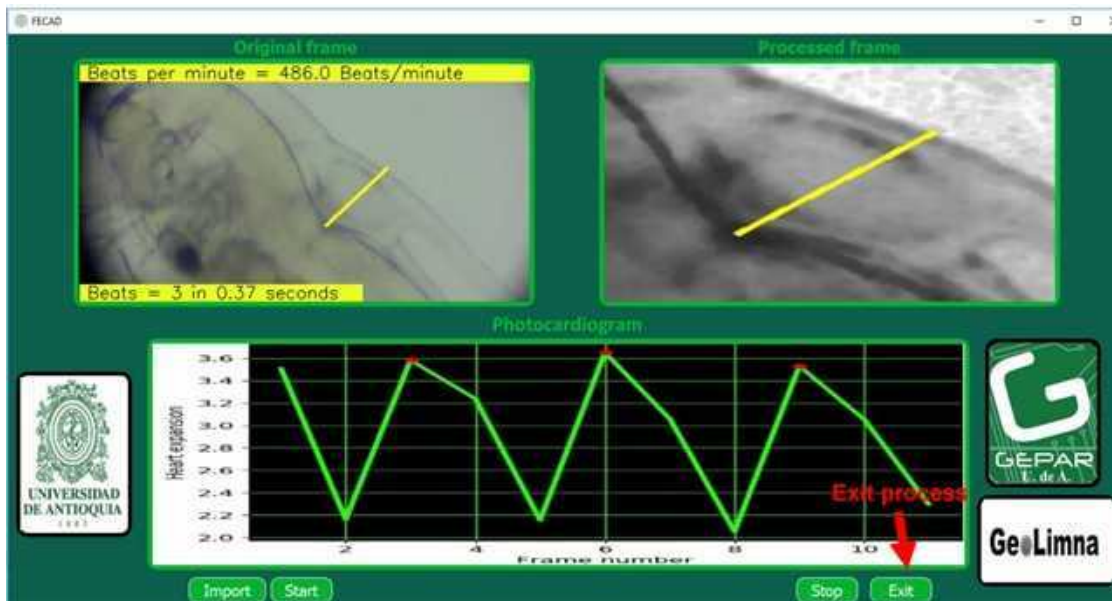


Figura 7.6: Señal  $I(t)$  (fotocardiograma digital): máximos locales asociados a latidos; los cambios de pendiente soportan el conteo automático de pulsaciones.

## 7.4. Implementación software (FECAD)

FECAD se implementa como una aplicación con interfaz gráfica que permite: importar video, trazar la sección  $S$  sobre el corazón (dos clics), ejecutar el procesamiento y visualizar simultáneamente (i) fotograma original, (ii) fotograma procesado y (iii) el fotocardiograma  $I(t)$ , además de reportar bpm y estadísticas (máximo, mínimo, media, varianza, desviación estándar de la expansión) [4]. La documentación técnica describe una arquitectura modular desarrollada en Python 3.7 con PyQt5 y OpenCV, donde el método

`heart_rate_calculation()` constituye el núcleo: preprocesa el fotograma, aplica una operación AND con la máscara de la línea de análisis y estima la expansión cardiaca por una medida geométrica asociada a dicha intersección, almacenando máximos para derivar frecuencia [3].

## 7.5. Resultados

En el estudio reportado se presenta correspondencia entre la estimación automática y el conteo por observación directa (experto), con un error menor al 1% en el escenario evaluado [1]. Bajo condiciones de laboratorio, el método reporta una frecuencia promedio del orden de 480 bpm, dentro del rango discutido frente a literatura, con énfasis en la necesidad de automatizar debido a la alta tasa de eventos [1]. Adicionalmente, la documentación de usuario describe que el software almacena resultados en archivos `.xlsx` al detener o finalizar la ejecución, preservando trazabilidad por video y carpeta de salida [4].

## 7.6. Discusión

El enfoque de fotocardiograma digital reemplaza el conteo visual por una señal cuantitativa derivada de intensidades en una sección fija que atraviesa la región cardiaca. Su robustez depende de dos mecanismos complementarios: (i) compensación del movimiento global del individuo (SURF) y (ii) preprocesamiento que estabiliza condiciones de iluminación (gaussiano, morfología y umbral adaptativo) [1]. La motivación experimental es evitar procedimientos de fijación mecánica/química que pueden inducir taquicardia y contaminar el biomarcador bajo estudio, privilegiando mediciones “más puras” del tensor toxicológico [2].

Como limitación práctica, el rendimiento depende de la calidad de captura: enfoque, estabilidad del microscopio, iluminación de fondo y selección correcta de la sección  $S$ . Por ello, el diseño del software incorpora un paso guiado de selección de línea y visualización simultánea de fotograma procesado y señal  $I(t)$  para auditoría del usuario [4].

## 7.7. Conclusiones

Se presentó FECAD como un desarrollo aplicado de visión artificial que estima automáticamente la frecuencia cardiaca de *Daphnia pulex* a partir de videos de microscopio inver-

tido, mediante un fotocardiograma digital basado en suma de intensidades en una sección fija, estabilización por SURF y preprocesamiento para invariancia parcial a iluminación. La formulación definió explícitamente el video  $V(x, y, t)$ , la señal  $I(t)$  y la conversión a frecuencia  $P_s(t_O) = n_b/t_O$ , con reporte en bpm. El método reporta correspondencia con conteo experto y error menor al 1% en el escenario evaluado, habilitando medición escalable y reproducible en estudios de biotoxicidad [1]. El uso institucional del software está certificado para GEOLIMNA y GEPAR [5] y la solicitud de puntaje reporta protección de propiedad intelectual vía registro DNDA (Libro–Tomo–Partida 13-81-196) [6].

## Agradecimientos

Se agradece al soporte institucional y a los grupos de investigación GEOLIMNA y GEPAR por el entorno experimental y la articulación del problema ambiental con la solución computacional, así como a los apoyos descritos en la documentación del proyecto y en el artículo asociado [1], [2].

## Referencias

- [1] D. Fernández, Y. T. Puerta, F. Velez y N. Aguirre, *Method for estimating the heart rate of D. pulex*, Manuscrito / PDF interno (adjunto), Insumo: Method for estimating the heart rate of D. pulex.pdf, 2020.
- [2] D. S. Fernández Mc Cann, F. d. J. Vélez Macías y N. J. Aguirre Ramírez, *Solicitud para asignación de puntos por desarrollo de software: FECAD*, Formato institucional (PDF), Universidad de Antioquia, Insumo: 1\_Formato\_solicitud\_ESVIDA.pdf (allí se reporta el software FECAD), 2020.
- [3] *FECAD: Software to estimate the heart rate of the Daphnia pulex. Service Manual*, Insumo: Service Manual - FECAD.pdf, Gepar y GeoLimna groups, Universidad de Antioquia, 2020.
- [4] *FECAD: Software to estimate the heart rate of the Daphnia pulex. User Guide*, Insumo: User Guide - FECAD.pdf, Gepar y GeoLimna groups, Universidad de Antioquia, 2020.
- [5] F. d. J. Vélez Macías y D. S. Fernández Mc Cann, *Certificación de uso del software FECAD por los grupos GEOLIMNA y GEPAR*, Documento interno (DOCX), Universidad de Antioquia, Insumo: 2\_Certificación\_uso\_FECAD.docx, 2020.

- [6] Dirección Nacional de Derecho de Autor (DNDA), *Registro de software reportado para FECAD (Libro-Tomo-Partida 13-81-196; 11-sept.-2020)*, Citado en el formato de solicitud institucional, El número de registro se consigna en la solicitud institucional (anexo)., 2020.

# Capítulo 8

## Sistema inalámbrico para la estimación de la velocidad de flujo de agua

### Autores

John Fernando Ramírez Gómez<sup>1</sup>; Mateo Arroyave Cañas<sup>2</sup>; David Fernández Mc Cann<sup>3</sup>; Néstor Jaime Aguirre Ramírez<sup>4</sup>; Fabio de Jesús Vélez Macías<sup>5</sup>.

Grupos Geolimna y Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

La medición de velocidad de flujo en corrientes naturales constituye una tarea central en estudios hidrológicos, limnológicos y de monitoreo ambiental, donde la calidad de los datos depende tanto del principio físico de medición como de la robustez operativa del instrumento usado en campo [1], [2]. En este capítulo se presenta *CorrentómetroBT*, un sistema desarrollado para estimar la velocidad del agua mediante un sensor de pulsos, un microcontrolador Arduino Nano, un enlace Bluetooth y una aplicación móvil Android para visualización y almacenamiento de datos [1], [3]. La propuesta reorganiza la lógica convencional de medición al desacoplar físicamente el punto de sensado del punto de lectura, permitiendo que el sensor opere cerca del flujo mientras la información es recibida en

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. john.ramirez@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. mateo.arroyave@udea.edu.co

<sup>3</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

<sup>4</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. nestor.aguirre@udea.edu.co

<sup>5</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. fabio.velez@udea.edu.co

un teléfono inteligente, reduciendo exposición de la electrónica y mejorando la ergonomía en campo [2], [4]. Se describe la arquitectura general del sistema, la formulación matemática que relaciona pulsos, frecuencia, velocidad y caudal, así como la implementación electrónica y software del dispositivo. El desarrollo muestra cómo una necesidad concreta de Geolimna fue traducida por GEPAR en una herramienta instrumental replicable, documentada y con valor académico y práctico [1], [3].

**Palabras clave:** correntómetro; Bluetooth; Arduino Nano; velocidad de flujo; caudal; instrumentación ambiental; Android; Geolimna; GEPAR.

## 8.1. Introducción

Medir el movimiento del agua es, en el fondo, traducir un fenómeno continuo del mundo físico a una magnitud registrable y comparable. En estudios ambientales, esta traducción adquiere un papel decisivo porque variables como la velocidad de flujo y el caudal permiten interpretar dinámicas hidráulicas, caracterizar corrientes superficiales y sustentar análisis ecológicos e hidrológicos [1]. Sin embargo, en campañas de campo, la necesidad de medir no siempre coincide con la disponibilidad de equipos cómodos, robustos o económicamente accesibles.

En ese contexto surge *CorrentómetroBT*, un desarrollo que articula sensado, procesamiento embebido, comunicación inalámbrica y una interfaz móvil de usuario. Más que reproducir un correntómetro convencional, la propuesta reordena el proceso de medición: el sensado permanece cerca del agua, mientras que la lectura y gestión de datos se desplazan al teléfono del operador. Esta decisión técnica resulta especialmente relevante en ambientes húmedos o inestables, donde conviene minimizar la exposición directa del bloque electrónico [2], [4].

La Figura 8.1 resume conceptualmente esta arquitectura de trabajo, mostrando la articulación entre sensor, unidad de procesamiento, transmisión Bluetooth y aplicación móvil. Esa relación entre bloques define la esencia del desarrollo y evidencia su carácter de sistema integrado más que de dispositivo aislado.

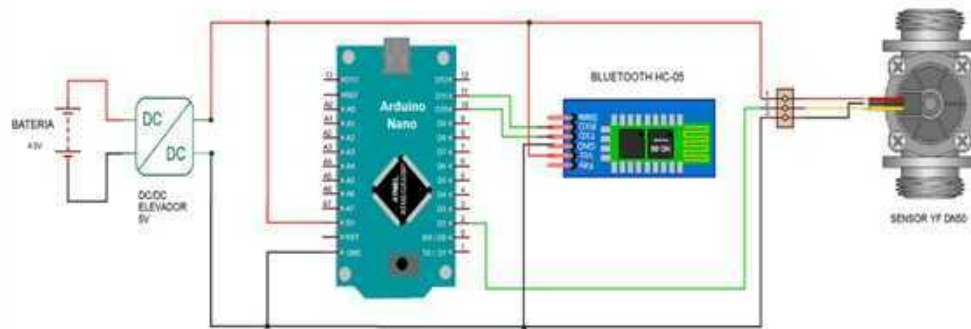


Figura 8.1: Esquema general del sistema *CorrentómetroBT*. Se ilustra la relación funcional entre el sensor de flujo, la etapa de adquisición basada en Arduino Nano, el módulo Bluetooth y la aplicación móvil encargada de recibir, visualizar y almacenar la información.

## 8.2. Planteamiento del problema

La medición de velocidad en corrientes naturales suele realizarse con instrumentos especializados que, aunque funcionales, presentan limitaciones relevantes en contextos de trabajo universitario y campañas de campo. Estas limitaciones incluyen el costo de adquisición, la dificultad de adaptación a necesidades particulares y, sobre todo, la incomodidad operativa derivada de leer localmente el instrumento mientras se maniobra el sensor en condiciones ambientales complejas [1], [2].

El problema no era solamente obtener una estimación de velocidad, sino hacerlo de una manera compatible con la práctica real de medición. Desde la perspectiva de Geolimna, se requería un instrumento útil para el trabajo de campo. Desde la perspectiva de GEPAR, el desafío consistía en materializar una solución electrónica portátil, inalámbrica y suficientemente robusta para ambientes naturales. Por ello, el proyecto se formuló como el diseño de un sistema capaz de recibir pulsos de un sensor de flujo, convertirlos en velocidad promedio, transmitirlos a distancia y desplegarlos en una interfaz móvil legible y almacenable.

La importancia de esta reformulación se hace más evidente cuando se observa el tipo de escenario donde el equipo debía operar. La Figura 8.2 sugiere el contexto de uso esperado: zonas de medición donde la seguridad, la movilidad y la reducción de riesgo para la electrónica son condiciones tan importantes como la lectura misma.



Figura 8.2: Contexto general de uso del *CorrentómetroBT* en labores de medición de flujo en campo. La figura busca resaltar el carácter aplicado del desarrollo y la necesidad de desacoplar el punto de sensado del punto de lectura para mejorar ergonomía y seguridad operativa.

### 8.3. Objetivo general y objetivos específicos

La documentación del desarrollo plantea como propósito central la construcción de un correntómetro que, utilizando el sensor y la infraestructura suministrada por Geolimna, pudiera operar a distancia [1]. Esta formulación no solo enuncia una meta funcional, sino que revela el principio de diseño que orienta todo el sistema: medir sin obligar al operador a permanecer físicamente unido al punto de lectura.

#### Objetivo general

Realizar un correntómetro con el sensor e infraestructura suministrada por el grupo de investigación Geolimna que funcione a distancia.

A partir de este propósito, el trabajo se desplegó en una serie de objetivos específicos que abarcan el diseño de circuito, la caracterización del sensor, la programación del microcontrolador y la construcción de la aplicación móvil. En conjunto, estos objetivos muestran que el proyecto no se limitó a ensamblar componentes, sino que construyó una arquitectura

completa de instrumentación.

## Objetivos específicos

1. Diseñar un circuito que cumpla con necesidades de portabilidad y funcionalidad para trabajar en ambientes naturales.
2. Hallar la relación pulsos-velocidad del sensor YF-DN50 suministrado para el desarrollo.
3. Programar un microcontrolador Arduino Nano v3.0 para recibir los pulsos del sensor y calcular la velocidad promedio equivalente.
4. Crear una aplicación Android para la recepción, visualización y almacenamiento de la información transmitida por Bluetooth.

## 8.4. Arquitectura general del sistema

### 8.4.1. Visión general

La arquitectura del *CorrentómetroBT* puede entenderse como una cadena de transformación: el agua en movimiento acciona el sensor, el sensor produce pulsos, el microcontrolador interpreta esos pulsos, el módulo Bluetooth transmite la información y la aplicación móvil la convierte en una experiencia operativa de lectura y almacenamiento [1], [4]. De esta forma, el sistema no solo mide, sino que organiza el tránsito desde el fenómeno físico hasta el dato utilizable.

Los bloques principales del sistema son:

1. **Bloque de sensado:** sensor de flujo YF-DN50.
2. **Bloque de procesamiento embebido:** Arduino Nano v3.0.
3. **Bloque de comunicación inalámbrica:** módulo Bluetooth HC-05.
4. **Bloque de visualización y almacenamiento:** aplicación Android.

Esta estructura modular puede apreciarse con mayor claridad en la Figura donde se sintetiza el diseño electrónico general. La figura permite al lector reconstruir visualmente la arquitectura física del sistema y comprender cómo se integran alimentación, adquisición y transmisión en un mismo desarrollo.

### 8.4.2. Componentes principales

La selección de componentes del sistema responde a una lógica de accesibilidad, portabilidad y compatibilidad entre módulos. La documentación reporta el uso de:

- Microcontrolador Arduino Nano v3.0.
- Módulo Bluetooth HC-05.
- Regulador elevador DC–DC de 5 V.
- Portapilas con tres baterías AAA de 1.2 V.
- Conector impermeable de 3 pines.
- Sensor de flujo YF-DN50.
- Teléfono inteligente con sistema operativo Android.

Más allá del inventario técnico, esta selección deja ver una decisión de diseño importante: construir una solución instrumental replicable en contexto universitario, basada en componentes abiertos y de amplia disponibilidad.

## 8.5. Formulación matemática del principio de medición

Toda instrumentación útil requiere una relación explícita entre la señal adquirida y la variable física que se desea estimar. En este caso, la señal primaria está formada por pulsos eléctricos generados por el sensor, y la variable buscada es la velocidad del agua. El valor del sistema está en convertir esa señal discreta en una magnitud con sentido hidráulico [1].

### 8.5.1. Conteo de pulsos y frecuencia

Sea  $N_p(T)$  el número de pulsos detectados durante una ventana temporal de duración  $T$ . La frecuencia de pulsos se define como

$$f_p = \frac{N_p(T)}{T},$$

donde  $f_p$  se expresa en pulsos por segundo.

### 8.5.2. Conversión de frecuencia a velocidad

Sea  $k_f$  el factor de conversión entre frecuencia y velocidad, y sea  $v$  la velocidad del agua. Entonces,

$$v = k_f f_p.$$

La documentación del desarrollo reporta el uso del parámetro `FacConv = 0.2` para esta conversión en el software embebido [1]. Como el flujo puede presentar variaciones temporales, se considera además un promedio sobre  $M$  observaciones sucesivas:

$$\bar{v} = \frac{1}{M} \sum_{i=1}^M v_i.$$

### 8.5.3. Estimación de caudal

Si  $A$  representa el área transversal efectiva de la sección de medición, el caudal  $Q$  puede estimarse mediante

$$Q = A \bar{v}.$$

Esta cadena de relaciones convierte al dispositivo en algo más que un contador de pulsos: lo transforma en una herramienta capaz de entregar información físicamente interpretable para estudios ambientales.

## 8.6. Implementación electrónica

### 8.6.1. Diseño del circuito y alimentación

El circuito integra el sensor, el microcontrolador, el módulo Bluetooth y una alimentación portátil basada en baterías AAA y un elevador DC-DC de 5 V [4]. Esta arquitectura responde a la necesidad de operar en campo sin depender de infraestructura fija y manteniendo compatibilidad eléctrica entre módulos.

La Figura 8.3 muestra el montaje físico del sistema o su disposición general de ensamblaje. Su inclusión es importante porque permite pasar del plano lógico al objeto material,

haciendo visible cómo el desarrollo sale del esquema y toma forma como instrumento real.



Figura 8.3: Montaje físico o disposición general del *CorrentómetroBT*. La figura permite apreciar la materialización del sistema y su orientación hacia un uso portátil en campañas de campo.

### 8.6.2. Sellado y protección física

Uno de los aportes prácticos más importantes del desarrollo es la decisión de proteger el bloque transmisor en un encapsulado estanco. Esta característica no solo protege la electrónica frente a salpicaduras y manipulación en terreno, sino que expresa una comprensión madura del contexto de uso: un equipo de campo no solo debe funcionar, sino resistir las contingencias reales de operación [2].

## 8.7. Implementación de software embebido

### 8.7.1. Lógica general en Arduino

El software embebido constituye el primer nivel de interpretación del sistema. Su tarea consiste en contar pulsos, convertirlos en velocidad promedio y transmitir los resultados mediante Bluetooth. La documentación del desarrollo reporta el uso de interrupciones sobre el pin de entrada para capturar la señal del sensor, una estrategia adecuada cuando se desea evitar pérdida de eventos y mantener consistencia temporal [1], [5].

Si  $C(t)$  representa el contador acumulado hasta el instante  $t$ , cada pulso válido actualiza el estado según

$$C(t_k^+) = C(t_k^-) + 1,$$

donde  $t_k$  representa el instante del pulso  $k$ .

La Figura del flujo embebido del correntometro se resume esta lógica de operación en forma de flujo. La visualización del algoritmo ayuda a entender que el sistema no depende de una sola lectura instantánea, sino de una secuencia organizada de adquisición, cálculo, promediado y transmisión.

## 8.8. Aplicación móvil y recepción de datos

### 8.8.1. Interfaz y funciones principales

La aplicación Android es una parte constitutiva del instrumento. No se limita a recibir datos, sino que organiza la experiencia de uso del operador en campo. Según la documentación, la interfaz permite visualizar la lectura actual, conservar un historial y ejecutar acciones como pausar, reiniciar, guardar, cargar y desconectar [2].

Las funciones principales incluyen:

- visualización del dato actual;
- historial de mediciones;
- indicador de estado de conexión;
- Pausar;
- Reiniciar;
- Guardar datos;
- Cargar datos;
- Desconectar.

La Figura 8.4 muestra esta interfaz principal. Su valor dentro del capítulo es importante porque evidencia cómo el sistema traslada la lectura y la gestión de datos a un entorno más seguro, cómodo y auditables para el usuario.



Figura 8.4: Interfaz principal de la aplicación móvil de *CorrentómetroBT*. Se observan el valor actual, el historial de mediciones, el estado de conexión y los controles operativos de la sesión.

### 8.8.2. Alcance del enlace inalámbrico

La documentación señala que el sistema puede alcanzar del orden de 10 m de operación promedio, con menciones de hasta 20 m en espacio abierto [1], [2]. Esta diferencia es razonable si se interpreta como distinción entre alcance máximo teórico y distancia conservadora de uso. En cualquier caso, el logro técnico relevante es el mismo: liberar al operador de la lectura local junto al punto de sensado.

## 8.9. Operación en campo y manejo básico de errores

Un instrumento madura realmente cuando no solo funciona, sino cuando su uso y sus posibles fallas pueden ser comprendidos y diagnosticados por terceros. La documentación del *CorrentómetroBT* reporta problemas típicos asociados a alimentación, conexión Bluetooth y recepción de datos, lo cual constituye una señal de madurez documental [2].

Estos escenarios pueden agruparse así:

1. **Problemas de alimentación:** el equipo no enciende o presenta fallas por baterías o polaridad.
2. **Problemas de comunicación:** el teléfono no detecta el módulo o pierde conexión.

### 3. Problemas de adquisición: existe conexión, pero no llegan datos desde el sensor.

La Figura 8.5 ilustra la secuencia básica de operación del sistema desde el encendido hasta la lectura móvil. Esta referencia visual ayuda a que el lector entienda el dispositivo no solo como arquitectura técnica, sino como procedimiento de uso.



Figura 8.5: Secuencia general de uso del *CorrentómetroBT*: encendido del sistema, emparejamiento Bluetooth, recepción de datos y gestión de la medición en la aplicación móvil.

## 8.10. Aportes técnicos y alcance del desarrollo

El principal aporte del *CorrentómetroBT* es la construcción de una solución instrumental completa para una necesidad concreta de medición ambiental. El sistema integra sensado, procesamiento, comunicación y visualización en una arquitectura coherente y replicable [1], [3].

Además, el desarrollo articula varias capas de formación en un mismo proyecto: electrónica, programación embebida, transmisión inalámbrica, desarrollo móvil e interpretación hidráulica. Esa integración le otorga valor pedagógico, además de su utilidad práctica como herramienta de trabajo de campo.

## 8.11. Discusión

El *CorrentómetroBT* muestra con claridad cómo una necesidad ambiental real puede transformarse en un desarrollo ingenieril con sentido práctico. Su mérito no reside únicamente en usar Bluetooth o en desplegar información en Android, sino en haber reorganizado el acto de medir para hacerlo más compatible con las condiciones reales de campo [1], [2].

Las figuras incluidas a lo largo del capítulo permiten seguir esa transformación de manera progresiva: desde la arquitectura general del sistema y el plano electrónico hasta el montaje físico, la lógica embebida y la interfaz móvil. En conjunto, estas evidencias visuales no son ornamentales: cumplen la función de mostrar cómo el desarrollo pasa del problema al artefacto y del artefacto al uso.

Como limitaciones razonables, el sistema depende de la calibración apropiada del factor pulsos-velocidad, de la estabilidad del enlace inalámbrico y de la compatibilidad con ciertas versiones del ecosistema Android. No obstante, estas restricciones no disminuyen el valor del trabajo; más bien delimitan con claridad el espacio de mejora futura.

## 8.12. Conclusiones

Se presentó *CorrentómetroBT* como un sistema inalámbrico para la estimación de la velocidad de flujo de agua, construido mediante la integración de sensor de pulsos, Arduino Nano, Bluetooth y aplicación móvil Android. El capítulo mostró que su valor no radica solamente en la fabricación de un dispositivo, sino en la construcción de una solución instrumental completa, trazable y orientada al uso real en campo [1], [3].

La formulación matemática hace explícita la lógica del desarrollo: a partir del conteo de pulsos se obtiene una frecuencia; a partir de la frecuencia se estima una velocidad; y, conociendo el área de la sección, puede estimarse el caudal. Esa cadena convierte la señal electrónica en una variable hidráulica utilizable.

Desde una perspectiva práctica, el sistema mejora la ergonomía del operador y reduce la exposición del bloque electrónico al trasladar la lectura y gestión de datos hacia el teléfono móvil. Desde una perspectiva académica e institucional, constituye una muestra clara del tipo de convergencia entre Geolimna y GEPAR: una necesidad ambiental concreta resuelta mediante una traducción ingenieril verificable y documentada.

## Agradecimientos

Se agradece a los grupos de investigación GEOLIMNA y GEPAR por el entorno interdisciplinario que dio origen a este desarrollo, así como a las personas que participaron en su concepción, implementación, documentación y uso en actividades de investigación y formación [1], [2].

## Referencias

- [1] Grupo GEPAR y Grupo GeoLimna, *CorrentómetroBT: Software para transmisión y recepción de datos de velocidad medida por sensor. Manual técnico*, Universidad de Antioquia, Medellín, Colombia, 2020.
- [2] Grupo GEPAR y Grupo GeoLimna, *CorrentómetroBT: Correntómetro con conexión Bluetooth a aplicación móvil. Manual de usuario*, Universidad de Antioquia, Medellín, Colombia, 2020.
- [3] D. F. M. Cann, N. J. A. Ramírez, F. V. Macías, J. F. R. Gómez y M. A. Cañas, *CorrentómetroBT: Sistema para la estimación de la velocidad de flujo de agua*, Registro de soporte lógico–software, Dirección Nacional de Derecho de Autor, Libro–Tomo–Partida 13-78-260, fecha de registro: 05-mar-2020, 2020.
- [4] Grupo GEPAR y Grupo GeoLimna, *CorrentómetroBT: Hardware receptor y emisor de datos para correntómetro con conexión Bluetooth. Planos de diseño electrónico*, Universidad de Antioquia, Medellín, Colombia, 2020.
- [5] Arduino, *attachInterrupt() Reference*, Documentación oficial de interrupciones externas, Arduino, 2020.

# Capítulo 9

## SICOP Panorámica: construcción de una imagen panorámica de camas de plántulas

### Autores

Luis Javier Lora Larios<sup>1</sup>; Juan Sebastián Lobo Ramírez<sup>2</sup>; David Fernández Mc Cann<sup>3</sup>.  
Grupos Gepar y Geolimna, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

En la floricultura intensiva, conocer con precisión cuántas plántulas han sido sembradas en una cama y disponer de una representación visual completa del cultivo constituye un problema práctico con consecuencias directas sobre pronóstico de cosecha, control de calidad, productividad y uso de insumos. Sin embargo, una cama de siembra puede alcanzar dimensiones del orden de  $40 \times 1,4$  m y contener cerca de 3000 plántulas, lo que hace inviable su inspección detallada mediante observación directa o registro fotográfico convencional [1], [2], [3]. En este capítulo se presenta *SICOP Panorámica*, módulo de software del Sistema de Conteo de Plántulas en Cama (SICOP), cuyo objetivo es construir una imagen panorámica segmentada de alta resolución de toda la cama a partir de un video capturado mediante un protocolo estandarizado. El desarrollo integra un método de

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. luis.lora@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería de Sistemas. juan.lobo@udea.edu.co

<sup>3</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

captura con estabilización desde la fuente, selección de fotogramas, corrección de rotación, corrección de escala, centrado, recorte y un proceso de *stitching* basado en correlación para producir una sola imagen de la cama completa [2], [3]. El resultado de esta etapa es el insumo estructural para análisis posteriores; sin embargo, la individualización y conteo de plántulas pertenece al módulo *SICOP Analítica*, que se tratará en otro capítulo. Aquí se privilegia la lógica instrumental y computacional de la panorámica como solución autónoma a un problema de visión artificial en condiciones de alta repetitividad visual.

**Palabras clave:** panorámica; *stitching*; visión artificial; floricultura; plántulas; correlación; homografía; OpenCV; SICOP.

## 9.1. Introducción

La construcción de panorámicas suele asociarse, en primera instancia, a escenas urbanas o paisajes donde existen suficientes rasgos distintivos para alinear imágenes consecutivas. El caso de una cama sembrada de plántulas es distinto y mucho más exigente: la escena es altamente repetitiva, los elementos vegetales presentan gran similitud entre sí, la perspectiva cambia a medida que el operario avanza y la captura se realiza en condiciones reales de campo, donde pequeñas variaciones de inclinación, altura o velocidad afectan el resultado final. En otras palabras, no se trata simplemente de unir fotos, sino de imponer orden geométrico sobre una secuencia de imágenes muy parecidas entre sí.

SICOP surge precisamente en ese contexto. Fue desarrollado como una solución a la necesidad de registrar una cama completa de plántulas en una sola imagen de alta resolución que pudiera servir de base para inspección posterior. La etapa panorámica no pretende resolver por sí misma el conteo o la estadística morfológica final; su tarea es anterior y estructural: producir una representación visual continua, suficientemente estable y geoméricamente consistente, que haga posible el análisis posterior [1], [2].

Desde la perspectiva del seminario Geolimna–GEPAR, este desarrollo es particularmente representativo porque convierte una necesidad productiva concreta del sector floricultor en un problema formal de visión artificial. El reto no consistía en reconocer una plántula individual desde el comienzo, sino en garantizar que una cama extensa pudiera observarse como una sola unidad visual, minimizando las distorsiones introducidas por rotación, escala y desplazamiento. La lógica del capítulo se construye, por tanto, sobre esa pregunta central: ¿cómo obtener una panorámica útil cuando los algoritmos convencionales fallan por la regularidad extrema de la escena? [1].

La visión general de SICOP, incluyendo las etapas de captura, panorámica y analítica, se resume en la Figura 9.1. En este capítulo, no obstante, el foco se restringe a la fase

panorámica; la etapa analítica solo se menciona como continuidad del sistema y será desarrollada aparte.

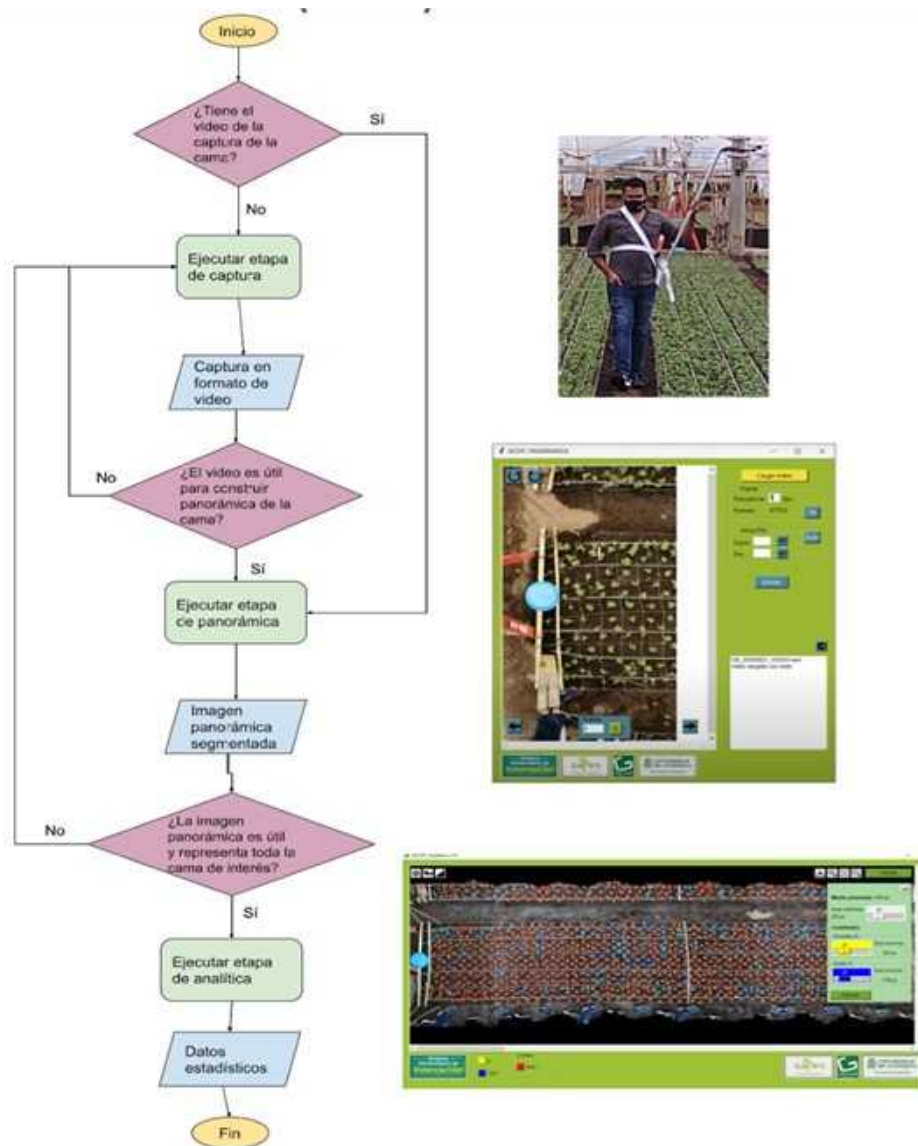


Figura 9.1: Diagrama general del sistema SICOP. El flujo completo contempla captura, panorámica y analítica. En este capítulo se estudia exclusivamente la etapa panorámica, encargada de transformar un video de la cama en una imagen panorámica segmentada de alta resolución que servirá como insumo para etapas posteriores.

## 9.2. Planteamiento del problema

En una floricultora, una cama sembrada puede contener miles de plántulas distribuidas manualmente con irregularidad espacial. Aunque desde producción sería deseable conocer con precisión cuántas plantas hay sembradas, evaluar la uniformidad del establecimiento y revisar visualmente la calidad de la cama, este procedimiento resulta impracticable cuando se intenta realizar plántula por plántula en campo. El problema no es solamente de tamaño físico, sino de tiempo, repetitividad visual y fatiga humana [1].

Formalmente, la dificultad puede expresarse así: dado un video  $V$  capturado durante un recorrido longitudinal sobre una cama sembrada, se requiere construir una imagen panorámica  $P$  de alta resolución que represente de forma continua la región completa de plántulas, corrigiendo errores de rotación, escala y traslación entre fotogramas, y preservando suficiente coherencia geométrica para que la imagen final pueda ser usada en inspección o análisis posterior [2]. Esta formulación desplaza el problema desde la observación humana hacia la representación computacional del cultivo.

La naturaleza de la escena impone además restricciones particulares. Según la documentación técnica, se hace necesario: i) extraer del video los fotogramas adecuados para conservar continuidad entre imágenes consecutivas; ii) corregir la rotación para que la cama quede verticalmente estable; iii) corregir variaciones de escala para que el ancho aparente de la cama sea uniforme; y iv) realizar la panorámica mediante técnicas de correlación sobre imágenes previamente corregidas [2]. Dicho de otro modo, la panorámica no es una sola operación, sino una secuencia de normalizaciones geométricas antes del cosido final.

La fase de captura estandarizada es parte esencial del problema. El desarrollo establece el uso de un arnés lateral para carga de barra liviana, un pivote estabilizador de video para celular (*gimbal*), un teléfono con capacidad de grabación a 1080p FHD y 60 fps, y testigos circulares azules para indicar el inicio y fin de la cama [1], [3]. La Figura 9.2 ilustra este principio de adquisición en campo, evidenciando que la calidad del algoritmo depende también de la calidad y regularidad del video de entrada.



Figura 9.2: Método de captura estandarizado para SICOP. El operador recorre longitudinalmente la cama mientras registra video con un sistema estabilizado. Esta etapa define el insumo principal del módulo panorámico y reduce variaciones de perspectiva desde la fuente.

### 9.3. Objetivo general y objetivos específicos

La documentación técnica del desarrollo define con claridad la intención del módulo panorámico: construir, a partir de un video, una sola imagen continua de la cama que pueda ser observada y usada como base para análisis posterior. Esta formulación es importante porque evita reducir el problema a una simple unión de cuadros; el propósito real es producir una representación útil del cultivo completo [2].

#### Objetivo general

Crear una panorámica de una cama de flores a partir de un video por medio de técnicas de visión artificial, con el fin de observar una sola imagen de la cama completa para su posterior análisis.

#### Objetivos específicos

1. Realizar una imagen panorámica de la cama por medio de técnicas de correlación.
2. Corregir los errores de escalamiento, rotación y traslación presentes en cada fotograma por medio de la matriz homografía.
3. Disminuir el tiempo de trabajo a la hora de hacer el conteo e inventario de cada una de las plántulas.
4. Desarrollar una interfaz gráfica que sea fácil de usar para el usuario por medio de Python.

Estos objetivos revelan dos niveles complementarios del proyecto. El primero es geométrico y algorítmico: estabilizar, alinear y coser imágenes. El segundo es operativo: hacer que el sistema pueda ser utilizado por una persona no especializada en programación, a través de una interfaz concreta y una secuencia guiada de pasos.

## 9.4. Modelo del problema y formulación básica

Sea  $V = \{f_1, f_2, \dots, f_N\}$  la secuencia de fotogramas extraídos de un video de captura, donde cada fotograma  $f_i$  es una imagen digital definida sobre un dominio discreto  $\Omega \subset \mathbb{Z}^2$ . El objetivo del módulo panorámico es construir una imagen compuesta  $P$  que represente la cama completa a partir de un subconjunto ordenado de fotogramas de  $V$ .

La primera decisión del sistema consiste en seleccionar un subconjunto

$$V_s = \{f_{i_1}, f_{i_2}, \dots, f_{i_M}\}, \quad 1 \leq i_1 < i_2 < \dots < i_M \leq N,$$

donde la separación entre índices está controlada por una frecuencia de muestreo elegida por el usuario. Si  $r$  representa la frecuencia de fotogramas considerados para el *stitching*, entonces de manera simplificada puede escribirse

$$i_{k+1} = i_k + r.$$

En la guía de uso se recomienda un valor de frecuencia igual a 3 para videos capturados con el método estándar de SICOP [3].

Cada fotograma seleccionado debe ser transformado geoméricamente antes del cosido final. Si  $T_i$  representa la transformación compuesta aplicada al fotograma  $f_i$ , entonces la imagen corregida correspondiente puede expresarse como

$$\tilde{f}_i = T_i(f_i),$$

donde  $T_i$  incorpora correcciones de rotación, escala, centrado y recorte. La documentación técnica asocia estas correcciones a rutinas específicas y señala el uso de homografía para compensar variaciones geométricas entre cuadros [2].

Una vez obtenida la secuencia corregida  $\{\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_M\}$ , la panorámica final puede presentarse como la composición incremental

$$P = \mathcal{S}(\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_M),$$

donde  $\mathcal{S}$  denota el operador de *stitching*. En este desarrollo, dicho operador se apoya en técnicas de correlación para localizar zonas de coincidencia entre imágenes consecutivas y coserlas sobre un mismo lienzo [2].

La importancia de esta formulación es conceptual: SICOP Panorámica no toma una sola fotografía de gran campo, sino que reconstruye computacionalmente una imagen extensa a partir de una secuencia de observaciones parciales. El problema es, en esencia, un problema de consistencia geométrica entre vistas locales del cultivo.

## 9.5. Metodología y pipeline del módulo panorámico

La metodología de SICOP Panorámica puede entenderse como una cadena de siete etapas, precedida por decisiones operativas del usuario en la interfaz. El usuario no interviene directamente en la matemática interna de cada rutina, pero sí define condiciones críticas del proceso: carga del video, orientación general, frecuencia de cuadros, y selección del frame inicial y final que delimitan la cama de interés [3]. Esa combinación de intervención humana guiada y procesamiento automático es una de las fortalezas del sistema.

En términos funcionales, el módulo permite:

1. cargar el video;
2. girar el video si es necesario;
3. visualizar fotogramas hacia adelante y hacia atrás;
4. ajustar la frecuencia de fotogramas para el *stitching*;
5. marcar el inicio y fin de la cama;
6. ejecutar la corrección geométrica de cuadros;
7. realizar el *stitching* y exportar la panorámica en formato `.tif`.

La Figura 9.3 resume la interfaz principal de operación. Allí se visualiza el cuadro actual, el valor de frecuencia, los campos de inicio y fin, el botón de extracción y la zona de mensajes de progreso. Esta interfaz evidencia que el software fue diseñado no solo como demostración algorítmica, sino como herramienta operativa para personal técnico o de campo.



Figura 9.3: Interfaz principal de SICOP Panorámica. El usuario puede cargar el video, revisar la orientación, definir la frecuencia de muestreo, establecer los frames de inicio y fin y ejecutar la extracción que conduce a la panorámica final.

Una vez definidas estas entradas, el sistema ejecuta una secuencia de procesamiento en siete etapas:

1. extracción de fotogramas;
2. rotación mediante método de Hough;
3. corrección de la rotación aplicada;
4. escalamiento;
5. corrección de escala;
6. centrado de imagen;
7. recorte de imágenes.

Después de estas etapas se genera la panorámica final y se almacena en la carpeta raíz del ejecutable como `stitching.TIF` [3]. La Figura ?? sintetiza esta lógica de transformación desde el video hasta la panorámica de alta resolución.

## 9.6. Implementación computacional

El desarrollo se implementó en Python 3.7.4 usando Spyder 3.3.6 del paquete Anaconda y la librería OpenCV 4.1.0, sobre un entorno Windows 10 con un equipo que en la documentación técnica se describe con procesador Intel Core i5-8250U, 8 GB de RAM, almacenamiento SSD de 256 GB y tarjeta gráfica Nvidia GeForce MX150 de 4 GB [2]. Estos datos no son accesorios; muestran el nivel computacional real sobre el cual el sistema fue desarrollado y probado.

El manual técnico reporta además una arquitectura modular organizada en rutinas como `rutinaRotacion`, `rutinaCorreccionRotacion`, `rutinaEscala`, `rutinaCorreccionEscala`, `rutinaCentrar`, `rutinaRecortar`, además de módulos vinculados al *stitching* y la correlación [2]. Esta organización modular es importante por dos razones. Primero, porque separa conceptualmente las tareas geométricas del cosido final. Segundo, porque hace posible extender o modificar el sistema sin reescribir toda la aplicación.

Desde la lógica de ingeniería del software, el módulo panorámico no fue concebido como un script monolítico, sino como una secuencia de subproblemas controlados. Cada rutina responde a una desviación concreta introducida por la captura real: inclinación, variación de tamaño aparente, desalineación de contenido útil o exceso de fondo negro. El *stitching* ocurre al final, cuando la imagen ya ha sido suficientemente normalizada como para hacer viable la correlación entre cuadros.

El sistema también contempla una estructura de carpetas y almacenamiento intermedio para los fotogramas corregidos en distintas etapas, lo que facilita la inspección técnica y la depuración. Esta trazabilidad es valiosa porque permite verificar visualmente el efecto de cada transformación antes de la panorámica final.

## 9.7. Resultados del módulo panorámico

El resultado esperado y reportado por la documentación es una imagen panorámica segmentada de alta resolución que representa la cama completa y que queda disponible como archivo `stitching.TIF` [3], [4]. Más que un archivo de salida, este resultado constituye el insumo estructural para cualquier examen posterior del cultivo: observación global, inspección visual y preparación para la etapa de análisis cuantitativo.

La Figura 9.4 muestra un ejemplo del producto generado por la etapa panorámica. Su importancia no radica solo en el tamaño de la imagen, sino en que sintetiza en una sola vista una región extensa del cultivo que, de otro modo, solo podría ser revisada frag-

mentariamente. Ese cambio de representación altera profundamente la forma de observar la cama: lo que antes exigía desplazamiento físico y memoria visual del operario, ahora queda condensado en una imagen continua.



Figura 9.4: Ejemplo de imagen panorámica resultante generada por SICOP Panorámica. La panorámica integra en una sola vista continua la cama completa de plántulas y constituye el insumo principal para las etapas posteriores de inspección y analítica.

La documentación institucional enfatiza que el desarrollo resuelve una tarea que los algoritmos convencionales no lograron abordar satisfactoriamente debido al alto parecido entre imagen e imagen a lo largo del recorrido de la cama [1]. Ese punto merece destacarse: el valor del módulo no es simplemente “hacer una panorámica”, sino hacerlo en un escenario visual donde la repetición geométrica y cromática reduce la discriminabilidad entre cuadros consecutivos.

En términos de aplicación, la panorámica habilita beneficios productivos importantes: mejor pronóstico de cosecha, soporte para control visual de la cama y eventual reducción de sobreproducción e insumos al mejorar la objetividad del inventario. No obstante, el conteo individual de plántulas y la estadística morfológica no se discuten aquí, pues corresponden al módulo SICOP Analítica, que será presentado en otro capítulo [1], [3].

## 9.8. Discusión

SICOP Panorámica muestra que, en ciertos problemas de visión artificial aplicada, el verdadero reto no está en clasificar o segmentar objetos desde el principio, sino en construir primero una representación adecuada del escenario. En este caso, antes de hablar de conteo, morfología o calidad de plántulas, era indispensable resolver la continuidad visual de la cama. Esa decisión metodológica es profunda: desplaza el énfasis desde el objeto individual hacia la estructura global del cultivo.

El desarrollo también evidencia una idea importante para la ingeniería aplicada: cuando la escena es muy regular, los métodos genéricos suelen ser insuficientes. El cultivo presenta alta repetición de patrones, cambios graduales de perspectiva y restricciones de captura impuestas por el entorno. Por ello, SICOP no depende solamente de una librería de *stitching* estándar, sino de una estrategia previa de estandarización geométrica apoyada en

corrección de rotación, escala, centrado y recorte. Esa secuencia convierte un problema inicialmente ambiguo en uno tratable.

Otra virtud del sistema es la articulación entre hardware de captura y software de procesamiento. El arnés, el *gimbal*, el teléfono a 1080p/60fps y los testigos azules no son complementos secundarios; forman parte de la solución completa. El algoritmo necesita una cierta disciplina de adquisición para funcionar con robustez. Esta relación entre método de captura y método computacional es un rasgo de madurez tecnológica del proyecto.

Como limitaciones razonables, debe señalarse que la calidad final depende del cumplimiento del protocolo de captura, de la correcta selección del tramo útil del video y de la estabilidad del recorrido sobre la cama. Además, al tratarse de una escena extensa, las pequeñas inconsistencias geométricas pueden acumularse si el video de entrada no respeta suficientemente la perpendicularidad y continuidad esperadas. No obstante, precisamente por eso el sistema tiene valor: porque formaliza un procedimiento reproducible para controlar esas variaciones y producir una salida útil.

### 9.9. Conclusiones

Se presentó SICOP Panorámica como un módulo de visión artificial orientado a construir una imagen panorámica de alta resolución de una cama completa de plántulas a partir de un video capturado en campo. El desarrollo formaliza un problema real del sector floricultor y lo resuelve mediante una secuencia ordenada de selección de fotogramas, corrección geométrica y *stitching* por correlación [1], [2].

La principal contribución del módulo es producir una representación continua del cultivo donde antes solo existían vistas parciales, fragmentarias y difíciles de comparar. Esa panorámica no es todavía el conteo, pero sí su condición de posibilidad. En ese sentido, la etapa panorámica tiene autonomía técnica: resuelve el problema de construir el insumo visual sobre el cual podrá operar la etapa analítica.

Desde el punto de vista computacional, el desarrollo muestra una arquitectura modular implementada en Python y OpenCV, con interfaz de usuario, trazabilidad entre etapas y almacenamiento del resultado final en formato `.tif`. Desde el punto de vista metodológico, confirma que en problemas de alta repetitividad visual la corrección geométrica previa es decisiva para hacer viable la panorámica. Y desde el punto de vista institucional, deja un software registrado, documentado y aplicado a una necesidad concreta del sector productivo.

Como proyección futura, este módulo puede servir de base para mejoras en automatización

de la captura, estimación adaptativa de la frecuencia de muestreo, validación cuantitativa del error geométrico acumulado y extensiones hacia otros cultivos o escenas agrícolas lineales de gran longitud.

## Agradecimientos

Se agradece a la Universidad de Antioquia, al grupo de investigación GEPAR, a la Unidad de Innovación Tecnológica y a la empresa Flores El Capiro S.A. por el contexto técnico y productivo que dio origen al desarrollo SICOP. Asimismo, se reconoce el trabajo de los estudiantes y autores que participaron en la construcción del software, su documentación y su protección intelectual [1], [4].

## Referencias

- [1] D. S. F. McCann, *SICOP - Panorámica: solicitud para asignación de puntos por desarrollo de software*, Documento institucional, Grupo GEPAR, Universidad de Antioquia, 13 de diciembre de 2021, 2021.
- [2] Grupo GEPAR, *SICOP Panorámica: Sistema de creación de imagen panorámica. Manual técnico*, Universidad de Antioquia, Medellín, Colombia, 2020.
- [3] Grupo GEPAR, *SICOP - Panorámica: Manual de usuario*, Universidad de Antioquia, Medellín, Colombia, 2020.
- [4] D. F. M. Cann, J. S. L. Ramirez y L. J. L. Larios, *SICOP - PANORAMICA*, Registro de soporte lógico–software, Dirección Nacional de Derecho de Autor, Libro–Tomo–Partida 13-85-301, fecha de registro: 16-jul-2021, 2021.

# Capítulo 10

## SICOP Analítica: Conteo y caracterización morfométrica de plántulas sembradas

### Autores

Juan Sebastián Lobo Ramírez<sup>2</sup>; Luis Javier Lora Larios<sup>3</sup>; David Fernández Mc Cann<sup>1</sup>;  
Grupo Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

En sistemas de producción florícola, disponer de una imagen completa de la cama sembrada es solo una parte del problema. La pregunta decisiva sigue siendo cuántas plántulas están realmente presentes, cómo se distribuyen morfológicamente y qué tan útil puede ser esa información para apoyar control de calidad, pronóstico de cosecha y seguimiento del cultivo. En este capítulo se presenta *SICOP Analítica*, módulo final del sistema SICOP, orientado a individualizar plántulas, generar conteo y producir estadísticas morfométricas a partir de una imagen panorámica segmentada generada por el módulo SICOP Panorámica [1], [2]. El software fue desarrollado por el grupo GEPAR en el marco del proyecto FIT-19-02-01 y se implementó en el contexto productivo de Flores El Capiro S.A. como

---

<sup>1</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

<sup>2</sup>Estudiante, Ingeniería de Sistemas. juan.lopez@udea.edu.co

<sup>3</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. luis.lora@udea.edu.co

una solución basada en visión artificial para resolver una tarea que, realizada manualmente, resulta impracticable por extensión del cultivo, irregularidad de la siembra y alta exigencia de atención sostenida [2], [3]. El capítulo describe el problema formal, la lógica funcional del software, la segmentación e individualización de plántulas, la exploración paramétrica de áreas y el reporte estadístico que produce el sistema. La documentación disponible sugiere además que el desarrollo se apoya en estrategias de reconocimiento de patrones y que fue concebido como un software robusto frente a ruido, oclusiones y presencia de elementos ajenos al cultivo, dejando como resultado una herramienta registrada, documentada y transferible [2].

**Palabras clave:** plántulas; conteo automático; morfometría; floricultura; visión artificial; segmentación; panorámica; SICOP; caracterización morfológica.

## 10.1. Introducción

En agricultura intensiva, la observación visual del cultivo ha sido históricamente una práctica indispensable, pero no siempre suficiente. Ver una cama sembrada permite intuir uniformidad, cobertura y vigor general; sin embargo, cuando la escala crece y la variabilidad de la siembra aumenta, la inspección humana deja de ser una herramienta cuantitativa confiable. El caso de las camas de plántulas en floricultura es un ejemplo claro: la disposición espacial es irregular, la cantidad de individuos puede superar varios miles por cama y la necesidad de contar con rapidez y precisión se vuelve incompatible con la revisión manual exhaustiva [2].

SICOP fue concebido precisamente para responder a esa dificultad mediante una arquitectura en tres etapas: captura, panorámica y analítica. Si la primera genera el video y la segunda construye una imagen continua y de alta resolución de la cama, la tercera —objeto de este capítulo— transforma dicha panorámica en datos objetivos sobre número de plántulas y rasgos morfométricos del cultivo [1]. En otras palabras, SICOP Analítica no se ocupa de reconstruir la escena, sino de interpretarla cuantitativamente.

Este cambio de enfoque es central. Una panorámica útil todavía no equivale a un conteo útil. Entre una imagen extensa del cultivo y una estadística confiable media un conjunto de operaciones que deben separar individuos, descartar ruido, agrupar regiones pertinentes y convertir el resultado en medidas comprensibles para el usuario. Ahí es donde la visión artificial deja de ser solamente visualización y pasa a ser medición.

La relación entre las tres etapas de SICOP puede apreciarse en la Figura que muestra la analítica general. En este capítulo, el interés se concentra en el módulo analítico, es decir, en la forma en que una panorámica ya disponible es procesada para individualizar plán-

tulas y generar estadísticas. La construcción de la panorámica, por su parte, corresponde al capítulo anterior dedicado a SICOP Panorámica [1].

## 10.2. Planteamiento del problema

En una cama sembrada de crisantemos, el problema de conteo no se reduce a identificar una rejilla ideal ni a multiplicar filas por columnas. La siembra manual introduce irregularidad, los individuos pueden aparecer parcialmente ocluidos, existen elementos ajenos como travesaños, mallas o residuos de campo, y la propia textura del suelo puede inducir falsas respuestas visuales. Por ello, contar plántulas sobre una panorámica no es un problema geométrico elemental, sino un problema de segmentación, individualización y clasificación de regiones en una escena compleja [2].

La documentación del software formula esta etapa como la fase final de SICOP, cuyo objetivo es individualizar y generar estadística de plántulas a partir de la imagen panorámica de la cama de interés. El resultado esperado no es solo el número total de plántulas, sino también información complementaria sobre la morfología de la cama analizada [1]. Esta doble salida —conteo y caracterización— es importante, porque el valor del sistema no radica únicamente en decir “cuántas hay”, sino también en describir cómo son las regiones detectadas en términos de tamaño y distribución.

Formalmente, el problema puede expresarse así: dada una imagen panorámica segmentada  $P$  que contiene una cama de plántulas, se requiere identificar un conjunto de regiones  $\mathcal{R} = \{R_1, R_2, \dots, R_n\}$  asociadas a individuos o agrupaciones relevantes, filtrar aquellas que correspondan a ruido o regiones no válidas y producir un conteo y un resumen estadístico de áreas que represente de manera útil la condición del cultivo [1], [2]. En esta formulación, la dificultad principal no es la existencia de imagen, sino la decisión de qué regiones deben considerarse plántulas válidas dentro de la escena.

La Figura 10.1 muestra el tipo de insumo requerido por el sistema: una panorámica previamente generada por SICOP Panorámica. Esta dependencia debe quedar explícita, porque el módulo analítico no opera sobre video ni sobre imágenes crudas de captura, sino sobre una representación ya estabilizada y consolidada del cultivo [1].



Figura 10.1: Imagen panorámica segmentada de entrada utilizada por SICOP Analítica. Esta imagen es el producto del módulo SICOP Panorámica y constituye el insumo sobre el cual se realiza la individualización, el conteo y la generación de estadísticas.

### 10.3. Objetivo general y objetivos analíticos

La documentación disponible permite reconstruir con claridad el propósito central del módulo analítico: transformar una panorámica del cultivo en datos objetivos sobre cantidad de plántulas y morfología general de la cama [1], [2].

#### Objetivo general

Realizar el análisis de una imagen panorámica de una cama de plántulas para individualizar, contar y caracterizar morfométricamente las plántulas sembradas, con tiempos de respuesta compatibles con el uso productivo.

A partir de este objetivo general, pueden reconocerse los siguientes objetivos analíticos de trabajo, coherentes con la lógica funcional del software documentado:

#### Objetivos analíticos

1. Cargar y visualizar una imagen panorámica segmentada proveniente del módulo SICOP Panorámica.
2. Individualizar regiones de interés asociadas a plántulas a partir de la imagen panorámica.
3. Generar estadísticas de conteo basadas en área promedio, área mínima y área máxima.
4. Permitir la exploración de parámetros para discriminar ruido o regiones no válidas.
5. Exportar un reporte estadístico utilizable para trazabilidad y análisis posterior.

Estos objetivos muestran que el sistema no se limita a una decisión binaria de presencia o ausencia, sino que incorpora una capa de control paramétrico que le permite al usuario ajustar el criterio de análisis según las condiciones de la imagen y del cultivo.

## 10.4. Modelo del problema y formulación básica

Sea  $P : \Omega \rightarrow \mathbb{R}^c$  una imagen panorámica, donde  $\Omega \subset \mathbb{Z}^2$  representa el dominio discreto de píxeles y  $c$  el número de canales de color. El objetivo de SICOP Analítica es producir una representación segmentada de los individuos de interés y un conjunto de descriptores estadísticos derivados de sus áreas [1].

Denotemos por  $B : \Omega \rightarrow \{0, 1\}$  una máscara binaria asociada al proceso de segmentación, donde

$$B(x, y) = \begin{cases} 1, & \text{si el píxel } (x, y) \text{ pertenece a una región considerada plántula,} \\ 0, & \text{en otro caso.} \end{cases}$$

A partir de esta máscara se obtiene un conjunto de componentes conexas

$$\mathcal{R} = \{R_1, R_2, \dots, R_n\},$$

donde cada  $R_i \subset \Omega$  corresponde a una región candidata a ser contabilizada.

El área en píxeles de cada región  $R_i$  puede definirse como

$$A_i = |R_i|,$$

siendo  $|R_i|$  el número de píxeles que conforman la componente conexa. Sobre este conjunto de áreas, el sistema calcula una media

$$\bar{A} = \frac{1}{n} \sum_{i=1}^n A_i,$$

que sirve como referencia para clasificar las plántulas según su tamaño relativo [1].

La lógica de exploración paramétrica descrita en el manual sugiere tres clases principales: regiones por debajo de un umbral mínimo relativo, regiones alrededor del área promedio y regiones que superan un umbral máximo relativo. Si  $\alpha_{\min}$  y  $\alpha_{\max}$  representan porcentajes relativos definidos por el usuario, entonces pueden escribirse condiciones de aceptación como:

$$A_i \geq \alpha_{\min} \bar{A}, \quad A_i \leq \alpha_{\max} \bar{A}.$$

De manera complementaria, el usuario define un área de eliminación  $A_{\text{ruido}}$  para descartar regiones pequeñas interpretadas como basura o ruido. Así, una condición básica de filtrado puede expresarse como

$$A_i > A_{\text{ruido}}.$$

La importancia de esta formulación no reside en una sofisticación matemática extrema, sino en hacer explícita la lógica del software: SICOP Analítica segmenta, identifica regiones, mide áreas y clasifica dichas regiones respecto a un tamaño medio de referencia. El conteo final emerge entonces como el número de regiones que satisfacen simultáneamente los criterios de validez y tamaño.

## 10.5. Metodología funcional del módulo analítico

La operación de SICOP Analítica puede comprenderse como una secuencia guiada de seis pasos: ejecución del software, carga de imagen panorámica, segmentación, generación de estadísticas, exploración de parámetros y descarga de reporte [1]. Esta organización es importante porque revela una filosofía de diseño muy clara: el sistema fue pensado para ser usado por personal operativo, no solo por desarrolladores o investigadores.

En términos de interacción, el usuario inicia ejecutando el archivo `SICOP_ANALITICA.exe`, accede a una interfaz gráfica principal y posteriormente carga la imagen panorámica de interés. Una vez cargada, el sistema permite segmentar las plántulas, pasar al módulo estadístico, ajustar parámetros de área y finalmente generar un reporte de salida. El hecho de que la exploración de parámetros ocurra después de una primera segmentación refleja una lógica iterativa de análisis: el sistema da una propuesta inicial y luego permite refinarla [1].

La interfaz principal del software se resume en la Figura 10.2. Allí se aprecia la organización operativa del módulo, con opciones para cargar la imagen, segmentar, acceder a estadísticas y descargar el reporte. Esta figura es importante porque muestra que el software no fue concebido como una simple rutina automática de consola, sino como una herramienta de trabajo con una interfaz explícita para el usuario final.

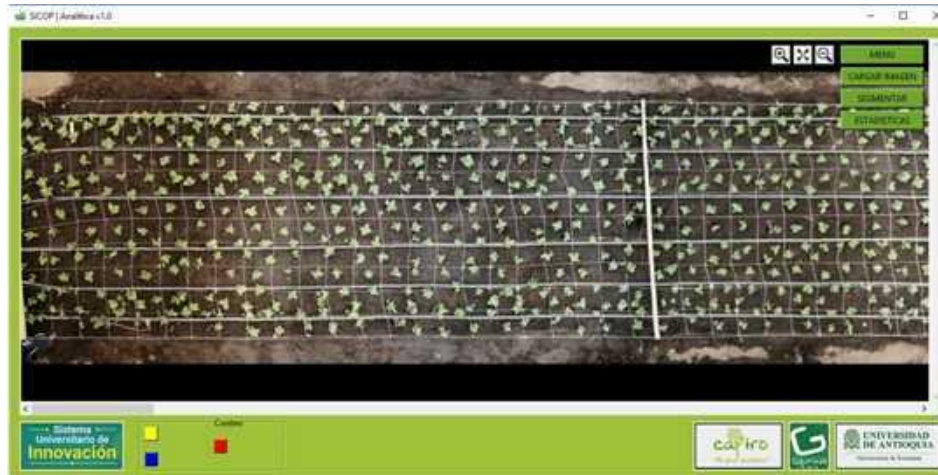


Figura 10.2: Interfaz principal de SICOP Analítica. Desde esta ventana el usuario carga la panorámica, ejecuta la segmentación, accede al módulo de estadísticas y exporta el reporte analítico del cultivo.

## 10.6. Segmentación e individualización de plántulas

Una vez la panorámica ha sido cargada en la interfaz, el usuario puede ejecutar la opción de segmentación. El manual indica que, tras esta acción, el sistema despliega una barra de progreso y posteriormente presenta la imagen panorámica segmentada [1]. Esto sugiere que el proceso puede implicar un tiempo computacional no despreciable, coherente con la resolución de la imagen y con la necesidad de procesar una escena extensa con numerosos individuos.

Desde la perspectiva analítica, la segmentación es el paso que convierte la panorámica en una escena interpretable por regiones. Allí dejan de importar tanto las texturas generales del suelo y pasan a ser relevantes las agrupaciones de píxeles que el sistema logra individualizar como plántulas o candidatos a plántula. La solicitud de puntos por software señala además que el sistema fue concebido para operar con imágenes algo sucias, con individuos sobrepuestos, oclusiones y presencia de otros elementos diferentes de las plántulas, como travesaños, mallas y basuras de campo [2]. Este punto es valioso porque describe la robustez esperada del módulo en condiciones no ideales.

La misma solicitud menciona que el carácter innovador del sistema radica en el uso de herramientas tecnológicas como *R-MASK* y *CNN* aplicadas al reconocimiento de patrones para la identificación de individuos de plántulas [2]. Dado que el manual de usuario no desarrolla el detalle algorítmico interno, en este capítulo conviene registrar esa información con prudencia: la documentación institucional atribuye al sistema una base en técnicas

inteligentes de reconocimiento de patrones, mientras que la interfaz de usuario expone su efecto funcional como segmentación e individualización.

La Figura 10.3 muestra el resultado visual de esta etapa, donde la imagen original es transformada en una escena segmentada con regiones candidatas a plántulas claramente diferenciadas del fondo.

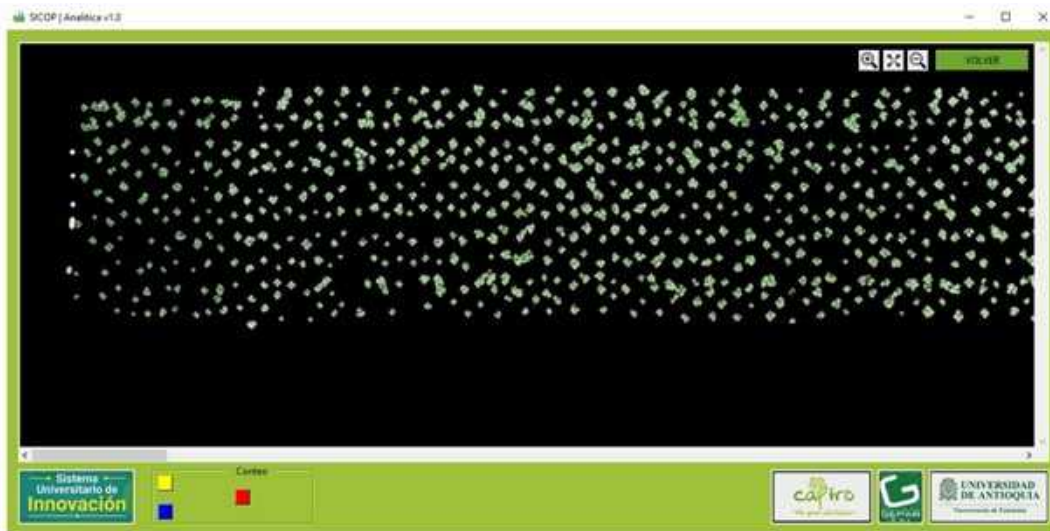


Figura 10.3: Resultado de la segmentación en SICOP Analítica. La escena resultante resalta las regiones asociadas a plántulas y permite al sistema pasar del dominio visual continuo de la panorámica al dominio discreto de regiones medibles.

### 10.7. Módulo estadístico y exploración paramétrica

Después de segmentar, el software permite acceder a un módulo de estadísticas donde se visualizan los resultados del conteo junto con parámetros ajustables. El manual describe explícitamente tres referencias cromáticas para las clases de tamaño: regiones por debajo del criterio mínimo, regiones cercanas al promedio y regiones por encima del criterio máximo, además de un parámetro de área a eliminar usado para descartar ruido o basura [1]. Este diseño es interesante porque convierte el análisis estadístico en una experiencia visual inmediata.

En términos conceptuales, el sistema no solo cuenta regiones, sino que organiza el conteo alrededor de una noción de tamaño medio. Esta decisión es importante porque evita tratar todas las regiones detectadas como equivalentes. En presencia de ruido, oclusión o fusión parcial de individuos, un conteo puramente bruto sería insuficiente; en cambio, un conteo mediado por criterios de área relativa permite interpretar mejor la estructura del cultivo.

El usuario puede modificar:

1. un valor de área a eliminar, medido en píxeles, para descartar ruido;
2. un porcentaje de área mínima respecto al área promedio;
3. un porcentaje de área máxima respecto al área promedio.

Una vez definidos estos parámetros, el sistema recalcula el resumen estadístico y actualiza las cantidades asociadas a cada clase. La documentación indica además la presencia visible de la media de plántulas, el área promedio y la desviación estándar [1]. Esta combinación de conteo y dispersión convierte al sistema en una herramienta morfométrica básica, no solo en un contador de objetos.

La Figura 10.4 ilustra este módulo. En ella se observa cómo la panorámica sirve de fondo para superponer regiones clasificadas y, a la vez, presentar un panel lateral con parámetros y métricas del cultivo.



Figura 10.4: Módulo de estadísticas de SICOP Analítica. La interfaz integra la visualización de la cama con parámetros de área mínima, área máxima, área a eliminar, conteo por clases, área promedio y otras medidas de resumen útiles para la caracterización del cultivo.

## 10.8. Reporte y trazabilidad del análisis

El último paso de la secuencia funcional consiste en la descarga de un reporte estadístico. El manual indica que, al hacer clic en el botón de reporte, el sistema genera un archivo en la carpeta raíz del ejecutable, el cual puede abrirse posteriormente como hoja de cálculo

[1]. Esta decisión de diseño es importante porque conecta el procesamiento visual con una salida documental trazable y reutilizable en procesos administrativos o productivos.

El reporte, según la evidencia mostrada en el manual, incluye campos como fecha, nombre de archivo, número de regiones en distintas clases cromáticas, área promedio y desviación estándar. Desde una perspectiva metodológica, esto significa que el software no se limita a mostrar resultados en pantalla: deja un registro persistente del análisis realizado sobre una imagen dada. En sistemas de producción, esta persistencia resulta tan importante como el algoritmo mismo, porque habilita comparación entre camaras, auditoría de resultados y acumulación histórica de información.

La Figura 10.5 resume esta salida final, mostrando la transición entre el análisis en pantalla y el archivo de reporte exportado. De esta forma, SICOP Analítica cierra el ciclo que inició con la panorámica: la imagen deja de ser solamente una observación del cultivo y se convierte en un documento cuantitativo.



	A	B	C	D	E	F	G	H
1		fecha	nombre archivo	# amarillas	# rojas	# azules	area promedio	desv. estandar
2	0	2021-02-14-17-56-36	2_CAMA_2_0919_380frames.TIF	72	3127	679	1123	425.1394335
3								
4								
5								

Figura 10.5: Generación y visualización del reporte estadístico de SICOP Analítica. El sistema exporta una hoja de cálculo con información del archivo procesado, conteo por clases, área promedio y medidas de dispersión, fortaleciendo la trazabilidad del análisis.

## 10.9. Implementación, robustez y alcance del desarrollo

La solicitud institucional describe a SICOP Analítica como un software nuevo, autónomo y complementario respecto a SICOP Panorámica, desarrollado en el marco del proyecto FIT-19-02-01 y pensado para operar sobre cualquier imagen que contenga plántulas sembradas [2]. Esta afirmación es importante porque sugiere que el valor del módulo trasciende la cama panorámica estricta y se extiende a otras imágenes de cultivos con plántulas, siempre que el tipo de escena sea compatible.

La misma documentación le atribuye varias propiedades relevantes: robustez frente a imágenes algo sucias, oclusiones y presencia de otros especímenes; extensibilidad del método; uso moderado de memoria y almacenamiento; y tiempos de operación cercanos a cinco minutos [2]. Aunque estos rasgos no se desarrollan algorítmicamente en el manual de usuario, sí ofrecen un marco técnico para comprender el alcance pretendido del sistema.

Desde el punto de vista institucional, el software cuenta con certificación de uso por parte del grupo GEPAR y registro ante la Dirección Nacional de Derecho de Autor con partida **\*\*13-85-302\*\***, lo que le da soporte como producto técnico documentado y protegido [3], [4]. En el contexto del libro, este hecho es especialmente valioso porque convierte al desarrollo en algo más que una demostración funcional: lo consolida como activo tecnológico con trazabilidad.

## 10.10. Discusión

SICOP Analítica representa la etapa donde la visión artificial se aproxima con mayor claridad al lenguaje de la producción. Mientras la panorámica resuelve el problema de construir una imagen completa y coherente de la cama, la analítica responde a la pregunta que realmente interesa al sistema productivo: cuántas plántulas hay y cómo se comportan morfológicamente dentro del cultivo.

Una de las fortalezas del desarrollo es haber entendido que el conteo útil no es simplemente una suma de regiones detectadas. El sistema incorpora área promedio, umbrales relativos y descarte de ruido, es decir, introduce criterios de interpretación sobre la segmentación. Eso lo hace más cercano a una herramienta de análisis que a un mero detector binario. También le da al usuario un papel importante: no como programador, sino como analista que puede explorar parámetros y revisar la sensibilidad del resultado.

Otro elemento destacable es la relación entre robustez y contexto. La documentación subraya que el software puede operar con imágenes imperfectas, con oclusiones y elementos ajenos al cultivo [2]. Esta afirmación es significativa porque responde al verdadero entorno de aplicación. En campo, la imagen rara vez será ideal; por eso, un sistema útil no es el que solo funciona en condiciones perfectas, sino el que conserva desempeño razonable en escenas contaminadas por la práctica real.

Como limitación, debe reconocerse que la documentación entregada privilegia la operación y el valor institucional del sistema más que la descripción exhaustiva de sus rutinas internas. Por ello, en este capítulo la formulación matemática se ha mantenido deliberadamente prudente: expresa con claridad la lógica de regiones, áreas y umbrales, pero evita atribuir detalles no completamente desarrollados en los manuales disponibles. Esa prudencia no debilita el capítulo; al contrario, preserva su fidelidad documental.

## 10.11. Conclusiones

Se presentó SICOP Analítica como el módulo final del sistema SICOP, encargado de individualizar, contar y caracterizar morfométricamente plántulas a partir de una imagen panorámica previamente construida por SICOP Panorámica [1]. El capítulo mostró que el valor del desarrollo no reside únicamente en la segmentación visual, sino en la transformación de una escena panorámica en datos objetivos sobre número de plántulas, áreas relativas y medidas de resumen del cultivo.

La formulación propuesta permitió expresar la lógica esencial del sistema: segmentar la imagen, obtener regiones, medir sus áreas, descartar ruido y clasificar regiones con respecto a un área promedio. Esta secuencia convierte el análisis de una cama sembrada en un procedimiento reproducible y ajustable, más compatible con las necesidades de producción que el conteo manual exhaustivo.

Desde el punto de vista aplicado, SICOP Analítica aporta una herramienta concreta para apoyar control de calidad, pronóstico de cosecha y reducción de incertidumbre en floricultura. Desde el punto de vista académico, integra visión artificial, análisis morfométrico y diseño de interfaz en un mismo sistema. Y desde el punto de vista institucional, deja un software documentado, certificado y registrado ante la DNDA, lo que confirma su madurez como desarrollo tecnológico.

Como proyección futura, sería deseable profundizar en la documentación formal del núcleo algorítmico, explorar validaciones cuantitativas frente a conteo experto y extender la lógica del sistema a otros cultivos o escenarios donde el problema de individualización y conteo sobre imágenes extensas siga siendo relevante.

## Agradecimientos

Se agradece a la Universidad de Antioquia, al grupo de investigación GEPAR, a la Unidad de Innovación Tecnológica y a la empresa Flores El Capiro S.A. por el contexto académico, tecnológico y productivo que dio origen al desarrollo SICOP. Asimismo, se reconoce la participación de los autores y colaboradores involucrados en la construcción, documentación, uso y protección intelectual del software [2], [3].

## Referencias

- [1] Grupo GEPAR, *SICOP - Analítica: Manual de usuario*, Universidad de Antioquia, Medellín, Colombia, 2020.
- [2] D. S. F. McCann, *SICOP - Analítica: solicitud para asignación de puntos por desarrollo de software*, Documento institucional, Grupo GEPAR, Universidad de Antioquia, 13 de diciembre de 2021, 2021.
- [3] D. S. F. M. Cann, *Certificación de uso del software SICOP Analítica*, Certificación institucional de uso otorgada al grupo GEPAR, Universidad de Antioquia, 16 de diciembre de 2021, 2021.
- [4] D. F. M. Cann, J. S. L. Ramirez y L. J. L. Larios, *SICOP - ANALITICA*, Registro de soporte lógico–software, Dirección Nacional de Derecho de Autor, Libro–Tomo–Partida 13-85-302, fecha de registro: 16-jul-2021, 2021.

# Capítulo 11

## Conteo automático y análisis morfológico de *Hyalella* mediante visión artificial

### Autores

Ludy Yanith Pineda Alarcón<sup>1</sup>; Santiago Ruiz González<sup>2</sup>; Maycol Esteban Zuluaga Montoya<sup>3</sup>; Yarin Tatiana Puerta Quintana<sup>4</sup>; Julio Eduardo Cañón Barriga<sup>5</sup>; David Fernández Mc Cann<sup>6</sup>; Néstor Jaime Aguirre Ramírez<sup>7</sup>; Fabio de Jesús Vélez Macías<sup>8</sup>.

Grupos Geolimna, Gepar y Gaia, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

El conteo y la caracterización morfológica de organismos del género *Hyalella* constituyen tareas relevantes en estudios ecológicos, limnológicos y ecotoxicológicos, ya que este

---

<sup>1</sup>Estudiante de maestría, Ingeniería Ambiental. ludy.pineda@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. santiago.ruiz@udea.edu.co

<sup>3</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. maycol.zuluaga@udea.edu.co

<sup>4</sup>Estudiante de maestría, Ingeniería Ambiental. tatiana.puerta@udea.edu.co

<sup>5</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. julio.canon@udea.edu.co

<sup>6</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

<sup>7</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. nestor.aguirre@udea.edu.co

<sup>8</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. fabio.velez@udea.edu.co

anfípodo es ampliamente utilizado como bioindicador de calidad ambiental en ecosistemas acuáticos. Sin embargo, el procedimiento tradicional, basado en observación manual bajo estereomicroscopio, demanda tiempos prolongados, alta concentración del analista y está expuesto a errores asociados a fatiga, repetición y subjetividad. En este capítulo se presenta *HyACS* (*Hyalella Automatic Counting System*), un software desarrollado para identificar, contar y extraer automáticamente características morfológicas de individuos de *Hyalella* contenidos en imágenes de cajas de Petri, mediante técnicas de visión artificial, procesamiento digital de imágenes y un modelo de detección basado en YOLO [1], [2], [3]. El sistema genera resultados individuales y promedios para variables como longitud, ancho, longitud de arco, perímetro, área y excentricidad, además de exportar la información a una hoja de cálculo y producir una imagen etiquetada con los organismos detectados [2]. Más allá de automatizar el conteo, el desarrollo introduce una forma más objetiva y detallada de describir la población, incorporando medidas que consideran la curvatura corporal del organismo y mejorando la relación entre tiempo invertido y cantidad de información obtenida. El capítulo sintetiza el problema, la lógica funcional del sistema, su formulación básica, su implementación y su relevancia como herramienta de apoyo a estudios de calidad del agua y dinámica poblacional.

**Palabras clave:** *Hyalella*; bioindicador; conteo automático; morfometría; YOLO; visión artificial; procesamiento digital de imágenes; ecotoxicología.

## 11.1. Introducción

En el estudio de ecosistemas acuáticos, ciertos organismos adquieren un valor que trasciende su tamaño y su aparente sencillez. Tal es el caso del género *Hyalella*, un anfípodo ampliamente distribuido en ambientes dulceacuícolas y utilizado como bioindicador en estudios ecológicos y ecotoxicológicos por su sensibilidad a cambios en la calidad del agua, a la presencia de metales pesados y a diversos tóxicos ambientales [2], [4]. Su abundancia, densidad y variabilidad morfológica pueden ayudar a inferir condiciones del ecosistema, fluctuaciones de nutrientes, alteraciones físico-químicas y dinámicas poblacionales en el tiempo y en el espacio.

Sin embargo, la utilidad científica de *Hyalella* depende en buena medida de la posibilidad de cuantificarla correctamente. Contar individuos y medir rasgos corporales sigue siendo, en muchos laboratorios, una tarea manual: se separan los organismos, se observan bajo estereomicroscopio y se registran uno a uno sus valores aproximados. Este procedimiento es costoso en tiempo, repetitivo y vulnerable al error humano. Cuando el número de individuos es alto, el problema se agrava, y la exactitud del resultado empieza a depender no solo del espécimen observado, sino del cansancio del observador.

HyACS surge precisamente para intervenir en ese punto crítico. El software transforma una imagen de una caja de Petri en un inventario cuantitativo de organismos detectados y de sus características morfológicas. De este modo, la imagen deja de ser simplemente un soporte visual y pasa a ser una fuente sistemática de medición. El reto no consiste solamente en contar objetos pequeños, sino en reconocer individuos con formas curvas, distribuciones espaciales irregulares y posibles contactos parciales entre sí, y además extraer variables útiles para el análisis biológico [1], [4].

Desde la perspectiva del seminario Geolimna–GEPAR, este capítulo representa bien la convergencia entre una necesidad ambiental concreta y una solución computacional trazable. HyACS no es solo un contador automático; es una herramienta que convierte una práctica tradicional de laboratorio en un procedimiento digital, reproducible y escalable, con valor tanto académico como aplicado.

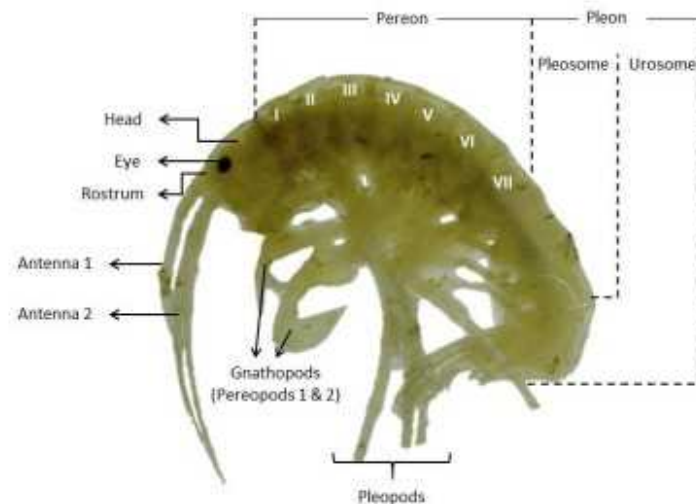


Figura 11.1: Anatomía general de *Hyalella*. La figura es útil porque sitúa visualmente al lector frente al organismo de interés y contextualiza la relevancia de medir rasgos morfológicos sobre una estructura corporal claramente curvada y segmentada.

## 11.2. Planteamiento del problema

El problema que da origen a HyACS puede formularse como una tensión entre importancia biológica y dificultad operativa. Por un lado, *Hyalella* es un organismo de interés para estudios de calidad ambiental, dinámica poblacional y ecotoxicología. Por otro, la metodología tradicional de conteo y medición consume mucho tiempo, exige observación sostenida bajo estereomicroscopio y produce resultados afectados por errores humanos derivados de fatiga, ilusión visual, aburrimiento y subjetividad [1], [4].

La situación se vuelve más exigente cuando se desea obtener no solo el número de individuos, sino también características morfológicas individuales y grupales. Una cosa es decir cuántos organismos hay en una muestra; otra, mucho más rica, es poder estimar para cada individuo medidas como longitud, espesor máximo, excentricidad, longitud de arco, perímetro y área, y luego producir estadísticas promedio y de dispersión para el conjunto [2], [4]. En el procedimiento manual, esta ampliación del nivel descriptivo suele implicar todavía más tiempo y más posibilidad de error.

Formalmente, el problema puede expresarse así: dada una imagen digital  $I$  de una caja de Petri que contiene un conjunto de organismos del género *Hyalella*, se requiere identificar automáticamente un conjunto de individuos  $\mathcal{H} = \{H_1, H_2, \dots, H_n\}$ , contar su cardinalidad  $n$  y extraer para cada uno de ellos un conjunto de descriptores morfológicos  $\mathbf{m}_i$ , de modo que el sistema entregue tanto resultados individuales como resultados agregados útiles para análisis ambiental y biológico [1], [2].

La Figura 11.2 ilustra el tipo de escena sobre el cual opera HyACS: una caja de Petri que contiene múltiples organismos distribuidos en un fondo relativamente homogéneo. Esta imagen de entrada resume la dificultad del problema: los individuos son numerosos, pueden estar próximos unos a otros y presentan curvaturas corporales diversas, lo que impide reducir el análisis a una simple detección geométrica elemental.



Figura 11.2: Imagen de entrada típica para HyACS: caja de Petri con múltiples individuos de *Hyalella*. A partir de esta escena el sistema debe localizar organismos, contarlos y extraer sus métricas morfológicas.

### 11.3. Objetivo general y objetivos específicos

La documentación del proyecto y del software permite reconstruir con claridad la intención del desarrollo: no solamente automatizar el conteo, sino también sistematizar la extracción de características morfológicas útiles para la descripción de la población y para la interpretación ambiental [1], [4].

#### Objetivo general

Establecer automáticamente el número total de individuos de *Hyalella* contenidos en una imagen de caja de Petri, así como el valor individual y grupal de sus principales características morfológicas observables.

## Objetivos específicos

1. Determinar un método de individualización de especímenes en imágenes de cajas de Petri.
2. Implementar un aplicativo útil para el usuario de laboratorio que permita obtener de forma ágil y gráfica las características individuales y grupales de los organismos detectados.
3. Extraer variables morfológicas relevantes como longitud, ancho, longitud de arco, perímetro, área y excentricidad.
4. Generar resultados tanto individuales como promedios, además de un archivo de salida con trazabilidad de la información.
5. Consolidar el desarrollo como software registrado y comunicable a la comunidad científica.

Estos objetivos dejan ver que el problema fue asumido en dos planos complementarios. El primero es algorítmico: detectar e individualizar organismos. El segundo es instrumental: hacer que el análisis sea útil, visualizable y exportable para un usuario real de laboratorio.

### 11.4. Modelo del problema y formulación básica

Sea  $I : \Omega \rightarrow \mathbb{R}^c$  una imagen digital de una caja de Petri, donde  $\Omega \subset \mathbb{Z}^2$  representa el dominio discreto de píxeles y  $c$  el número de canales de color. El propósito de HyACS es obtener a partir de  $I$  un conjunto de detecciones

$$\mathcal{H} = \{H_1, H_2, \dots, H_n\},$$

donde cada  $H_i$  corresponde a un individuo detectado del género *Hyalella*.

El conteo automático del sistema puede expresarse simplemente como

$$N = |\mathcal{H}|,$$

donde  $N$  es el número total de individuos identificados en la imagen.

Para cada individuo  $H_i$ , el sistema calcula un vector de características morfológicas

$$\mathbf{m}_i = (L_i, W_i, A_i, P_i, E_i, S_i),$$

donde:

- $L_i$  es una longitud asociada al individuo,

- $W_i$  es el ancho o espesor máximo,
- $A_i$  es el área,
- $P_i$  es el perímetro,
- $E_i$  es la excentricidad,
- $S_i$  es la longitud de arco.

A partir de este conjunto de mediciones individuales, HyACS calcula también métricas promedio para la población detectada. Si se considera una característica genérica  $x_i$  sobre los  $N$  individuos, su promedio puede expresarse como

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i.$$

De este modo se obtienen, por ejemplo, área promedio, longitud promedio, ancho promedio, excentricidad promedio y longitud de arco promedio [2].

Una característica especialmente interesante del sistema es la longitud de arco. A diferencia de una longitud lineal simple, esta variable intenta seguir mejor la curvatura corporal del organismo, lo cual resulta relevante en *Hyalella* por la forma arqueada de su cuerpo [5]. En términos conceptuales, si la silueta principal del organismo se representa mediante una curva discreta  $\gamma_i$ , la longitud de arco puede entenderse como

$$S_i = \int_{\gamma_i} ds,$$

o, en su versión discreta, como una suma de distancias elementales sobre la trayectoria estimada del cuerpo.

La importancia de esta formulación es que el sistema no se limita a detectar cajas o regiones, sino que transforma la detección en una representación cuantitativa del individuo y del conjunto. El resultado final es simultáneamente un conteo y una descripción morfométrica.

## 11.5. Metodología funcional de HyACS

HyACS fue concebido como una herramienta de laboratorio con una secuencia de uso sencilla y visual. El usuario importa una imagen de caja de Petri, puede hacer zoom sobre la escena, inicia la ejecución del algoritmo y posteriormente accede a resultados individuales, promedios y archivos exportados [2]. Esta simplicidad operativa es importante, porque evita que la potencia del algoritmo quede restringida a expertos en programación.

De manera funcional, el flujo de uso puede resumirse así:

1. importar una imagen válida de caja de Petri;
2. visualizarla y, si se desea, explorar regiones mediante *zoom*;
3. ejecutar el procesamiento automático;
4. observar la imagen procesada con las detecciones;
5. consultar resultados individuales;
6. consultar resultados promedio;
7. revisar los archivos exportados.

La Figura 11.3 resume la interfaz principal del sistema. En ella se reconocen los botones de importación, ejecución, consulta de individuos, consulta de resultados promedio y salida del programa. Esta organización revela un diseño orientado al usuario final, donde la lógica algorítmica compleja queda encapsulada en una experiencia de uso clara y guiada.



Figura 11.3: Interfaz principal de HyACS. El usuario puede importar una imagen, iniciar la ejecución, consultar resultados individuales, visualizar resultados promedio y cerrar el sistema.

## 11.6. Implementación computacional y arquitectura del software

La documentación técnica indica que HyACS fue desarrollado en Python 3.7.3, utilizando PyQt5 5.12.3 para la interfaz gráfica, OpenCV 4.1.0 para el procesamiento de imágenes y Qt Designer 5.9.7 para el diseño visual de la interfaz [1]. El entorno de prueba y desarrollo incluyó un computador con procesador Intel Core i5-8250U, 8 GB de RAM y sistema operativo Windows 10 [1].

Desde el punto de vista estructural, el sistema fue desarrollado de manera modular y se organizó alrededor de cuatro clases principales:

1. `MainWindow()`,
2. `Worker()`,
3. `Individuals()`,
4. `Results()`.

Esta organización es importante porque muestra que HyACS no fue escrito como un script lineal, sino como una aplicación estructurada. La clase principal administra la interfaz y la interacción del usuario; la clase `Worker()` permite ejecutar procesos en segundo plano usando *threads*; y las clases asociadas a resultados organizan la visualización individual y grupal de la información [1].

La documentación reporta además que el método `Start_execution()` se encarga de iniciar el proceso de detección y extracción de características, utilizando un modelo obtenido por YOLO para identificar individuos y luego calcular medidas como longitud, ancho, área, perímetro, longitud de arco y excentricidad [1]. Esta combinación entre aprendizaje profundo para detección y procesamiento geométrico posterior para medición constituye uno de los núcleos metodológicos del sistema.

En la etapa de entrenamiento, el documento de materiales y métodos señala tres acciones principales: selección de regiones útiles, aumento de datos mediante rotaciones y cambios de tamaño, y homogenización del tamaño de entrada de las imágenes [6]. Esta parte es relevante porque hace explícito que el sistema no depende únicamente de reglas manuales, sino de un proceso previo de entrenamiento y generalización del detector.

### 11.7. Resultados del sistema

Una vez completada la ejecución, HyACS muestra sobre la imagen original múltiples rectángulos verdes que indican la ubicación de los organismos detectados, así como la delimitación del borde de la caja de Petri [2]. Esta salida es importante porque permite una auditoría visual inmediata del conteo: el usuario no recibe solo un número, sino una evidencia gráfica de dónde están los individuos reconocidos por el sistema.

La Figura 11.4 ilustra este resultado general. En ella se observa cómo la imagen inicial se transforma en una escena etiquetada, donde cada organismo es localizado y el conjunto completo queda preparado para el cálculo morfométrico posterior.



Figura 11.4: Resultado general del procesamiento en HyACS. Los rectángulos verdes indican la localización de individuos detectados dentro de la caja de Petri, permitiendo una auditoría visual del conteo automático.

Además del etiquetado global, el software permite explorar resultados individuales. Para cada organismo detectado, se muestra una imagen aislada o recortada y un conjunto de métricas asociadas, entre ellas longitud de arco, longitud, ancho, área, excentricidad y perímetro [2]. Esta capacidad es particularmente valiosa porque conecta el conteo con la morfometría, y permite pasar del inventario grupal a la inspección específica de cada espécimen.

De manera complementaria, el sistema presenta una ventana de resultados promedio, donde aparece el número total de individuos detectados y las medias de las variables morfológicas calculadas [2]. Además, HyACS exporta una carpeta de resultados que contiene un archivo `Extracted_features.xlsx` y una imagen `labels.tif`. El archivo Excel incluye dos hojas: una con la información de cada organismo detectado y otra con los promedios y el conteo total [2].

La literatura anexa al desarrollo reporta además que el conteo automático redujo el tiempo del proceso aproximadamente a una cuarta parte respecto al conteo visual, con tiempos automáticos del orden de 22 a 58 s y una media cercana a 35 s, frente a un rango de 46 a 409 s y una media de 150 s en inspección manual. También se reportan métricas de desempeño superiores al 90% en exactitud y *recall* para la identificación correcta de individuos [5]. Estos datos no solo sustentan la utilidad del sistema, sino que lo sitúan como una herramienta eficiente y confiable.

## 11.8. Discusión

HyACS representa una transición importante en la forma de estudiar poblaciones de *Hyalella*. En lugar de depender exclusivamente de una observación manual prolongada, el sistema transforma la muestra en un objeto de procesamiento cuantitativo. Esta transición tiene al menos tres implicaciones. La primera es temporal: el análisis se acelera de forma significativa. La segunda es epistemológica: la medición deja de depender únicamente del ojo humano y adquiere mayor objetividad. La tercera es biológica: se vuelve posible incorporar variables morfológicas más ricas, como la longitud de arco, que consideran la curvatura real del organismo [5].

Un aspecto especialmente valioso del sistema es que no se limita a replicar el conteo tradicional más rápido. También amplía lo que se mide. Mientras que algunos procedimientos convencionales privilegian longitudes lineales aproximadas, HyACS incorpora descriptores que capturan mejor la forma del individuo. Esto es relevante porque en organismos curvos como *Hyalella*, la longitud lineal puede subestimar o distorsionar la dimensión corporal efectiva.

Otra fortaleza del desarrollo es su anclaje en necesidades reales de la investigación ambiental. La documentación del proyecto muestra que el interés por medir abundancia y morfología no era abstracto, sino que respondía a preguntas sobre calidad ambiental, dinámica poblacional y bioindicación en ecosistemas acuáticos [4]. En ese sentido, el software no es una aplicación aislada de visión artificial, sino una herramienta construida para dialogar con problemas concretos de limnología y ecología.

Como limitación razonable, debe reconocerse que la calidad del análisis depende de la calidad de la imagen de entrada, del proceso de detección y del comportamiento de los organismos en la muestra. También es cierto que la documentación funcional del sistema es más detallada en la operación del software que en la exposición matemática exhaustiva del núcleo algorítmico. Por ello, en este capítulo se ha optado por una formulación prudente, suficiente para expresar la lógica del método sin atribuir detalles no completamente explicitados en los manuales.

## 11.9. Conclusiones

Se presentó HyACS como un sistema de conteo automático y análisis morfométrico de *Hyalella*, desarrollado para identificar organismos en imágenes de cajas de Petri, contarlos y extraer variables individuales y grupales de interés ecológico y ambiental [1], [2]. El capítulo mostró que el aporte del software no consiste solo en automatizar una tarea

repetitiva, sino en convertir una práctica tradicional de laboratorio en un procedimiento digital, reproducible y cuantitativamente más rico.

La formulación básica permitió expresar con claridad la lógica del sistema: detectar un conjunto de individuos, contar su cardinalidad y calcular para cada uno un vector de características morfológicas, a partir del cual se obtienen también medidas promedio del grupo. Entre estas variables, la longitud de arco destaca como una contribución particularmente relevante para representar mejor la curvatura corporal del organismo.

Desde el punto de vista aplicado, HyACS reduce el tiempo de procesamiento, mejora la consistencia del conteo y entrega información morfométrica útil para estudios de dinámica poblacional, ecotoxicología y calidad ambiental. Desde el punto de vista académico, integra visión artificial, aprendizaje profundo, procesamiento geométrico e interfaz gráfica en una herramienta coherente y usable. Desde el punto de vista institucional, deja un software certificado, registrado y transferible, con respaldo de los grupos GEOLIMNA y GEPAR y de la Universidad de Antioquia.

Como trabajo futuro, sería valioso profundizar en la validación frente a nuevas condiciones de captura, ampliar la base de entrenamiento, explorar métricas morfológicas adicionales y extender la lógica del sistema a otros macroinvertebrados de interés como bioindicadores.

## Agradecimientos

Se agradece a la Universidad de Antioquia y a los grupos de investigación GEOLIMNA, GEPAR y GAIA por el contexto académico y experimental que hizo posible este desarrollo. Asimismo, se reconoce la participación de los autores, estudiantes e investigadores que contribuyeron al diseño, implementación, validación, documentación y uso institucional del software [5], [7].

## Referencias

- [1] Gepar, GeoLimna y Gaia, *HyACS: Hyalella Automatic Counting System. Service Manual*, Universidad de Antioquia, Medellín, Colombia, 2021.
- [2] Gepar, GeoLimna y Gaia, *HyACS: Hyalella Automatic Counting System. User's Guide*, Universidad de Antioquia, Medellín, Colombia, 2021.
- [3] D. S. F. M. Cann et al., *Solicitud para asignación de puntos por desarrollo de software: HyACS*, Documento institucional, Universidad de Antioquia, 13 de diciembre de 2021, 2021.
- [4] Grupo GEPAR y Grupo GAIA, *Método para estimar las características morfológicas de una Hyalella, útiles a la determinación de la calidad ambiental de una cuenca hidrográfica*, Documento de formulación de proyecto de investigación, 2017.
- [5] L. Pineda et al., "HyACS: Hyalella Automatic Counting System," *Environmental Science and Pollution Research*, vol. 30, págs. 123 603-123 615, 2023, Artículo reportado en la documentación anexa del proyecto.
- [6] Proyecto Hyalella, *Materiales y métodos Hyalella*, Documento interno sobre base de datos, entrenamiento y predicción con YOLO, 2021.
- [7] F. de Jesús Vélez Macías y D. S. F. M. Cann, *Certificación de uso del software HyACS*, Certificación institucional de uso por parte de los grupos GEOLIMNA y GEPAR, Universidad de Antioquia, 9 de diciembre de 2021, 2021.

# Capítulo 12

## Sistema de cuantificación y aforo de nubes a partir de imágenes de cielo

### Autores

Laura Carolina Sandoval Acosta<sup>1</sup>; Julián David Longas Arteaga<sup>2</sup>; David Fernández Mc Cann<sup>3</sup>; Mauricio Andrés Correa Ochoa<sup>4</sup>.

Grupos G-LIMA y GEPAR, Universidad de Antioquia

### Resumen

La observación de nubes ha sido tradicionalmente una práctica visual ligada a la meteorología, al monitoreo atmosférico y al análisis del estado del cielo. Sin embargo, transformar esa observación en variables cuantitativas comparables —como porcentaje nuboso, dirección media y velocidad aparente de desplazamiento— exige metodologías sistemáticas que reduzcan la subjetividad del observador y permitan registrar información con mayor trazabilidad. En este capítulo se presenta *SICUANU* (*Sistema de Cuantificación y Aforo de Nubes*), un software desarrollado para implementar un método que calcula el porcentaje nuboso y la dirección media de la velocidad de las nubes a partir de imágenes de una porción de cielo capturada con una cámara convencional dispuesta perpendicularmente al

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. laura.sandoval@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. julian.longas@udea.edu.co

<sup>3</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

<sup>4</sup>Profesor asociado, Departamento de Ingeniería Ambiental y Sanitaria. mauricio.correa@udea.edu.co

campo de observación [1]. El desarrollo cuenta con registro de software ante la Dirección Nacional de Derecho de Autor, certificación institucional de uso por parte de los grupos G-LIMA y GEPAR, y se inscribe en una línea de trabajo donde la visión artificial se pone al servicio del análisis ambiental y atmosférico [1], [2]. El capítulo reconstruye el problema, formaliza sus variables principales y propone una descripción metodológica prudente del sistema a partir de la información documental disponible. Aunque los documentos anexos privilegian la dimensión registral e institucional del desarrollo más que el detalle exhaustivo del algoritmo, permiten establecer con claridad la finalidad del software: convertir imágenes del cielo en medidas objetivas del estado nuboso y del movimiento aparente de las nubes.

**Palabras clave:** nubes; porcentaje nuboso; velocidad aparente; dirección media; visión artificial; meteorología; monitoreo atmosférico; SICUANU.

## 12.1. Introducción

Mirar el cielo ha sido, durante siglos, una forma elemental de anticipar cambios en el ambiente. Mucho antes de la existencia de sensores remotos, radares y satélites, las personas ya interpretaban la forma, densidad y desplazamiento de las nubes como indicadores del estado atmosférico. Sin embargo, esa observación, aunque útil, era profundamente dependiente del ojo humano y de la experiencia del observador. La meteorología moderna transformó esa intuición en medición, pero aún hoy persiste una distancia importante entre ver el cielo y cuantificarlo.

En contextos de investigación ambiental, esa distancia adquiere valor técnico. No basta con afirmar que “había muchas nubes” o que “el cielo estaba parcialmente cubierto”; se requiere estimar cuánto del campo visual está cubierto por nubosidad, hacia qué dirección parece desplazarse y con qué magnitud relativa. Estas variables son relevantes para el análisis de radiación, balance energético, dinámica local de la atmósfera y caracterización de condiciones ambientales en campañas de campo.

SICUANU surge precisamente en ese punto de encuentro entre observación del cielo y cuantificación computacional. Su propósito es convertir una imagen de una porción de cielo en variables medibles asociadas a la cobertura nubosa y al movimiento aparente de las nubes. El hecho de que el sistema utilice una cámara convencional orientada perpendicularmente al cielo observado sugiere una apuesta metodológica importante: construir una herramienta de análisis atmosférico apoyada en adquisición visual relativamente accesible, sin depender necesariamente de instrumentación especializada de alto costo [1].

Desde la perspectiva del seminario Geolimna–GEPAR, SICUANU es un desarrollo es-

pecialmente interesante porque extiende la visión artificial hacia el dominio atmosférico. Mientras otros capítulos del libro se concentran en agua, organismos o cultivos, aquí la escena de interés es el cielo mismo. Esto obliga a pensar la imagen no como un objeto estático, sino como una superficie cambiante donde textura, brillo, forma y desplazamiento deben interpretarse como información meteorológica.



Figura 12.1: Escena general de observación del cielo para el análisis nuboso. La figura sitúa el contexto físico del problema: una porción del cielo observada mediante una cámara convencional, sobre la cual se pretende estimar cobertura nubosa y movimiento aparente de las nubes.

### 12.2. Planteamiento del problema

La cuantificación del estado nuboso a partir de observación visual enfrenta dos dificultades principales. La primera es descriptiva: el porcentaje de cielo cubierto por nubes suele estimarse de manera subjetiva cuando se hace a simple vista. La segunda es dinámica: aun si la cobertura pudiera estimarse razonablemente, la dirección y velocidad aparente del movimiento de las nubes requieren comparar imágenes en el tiempo, lo que sobrepasa la capacidad de una observación instantánea y demanda una metodología explícita de análisis.

El certificado de registro del software indica que SICUANU hace posible la implementación de un método que calcula el porcentaje nuboso y la dirección media de la velocidad, incluyendo datos como la velocidad y el ángulo de dirección de las nubes, a partir de una porción de cielo tomada con una cámara convencional ubicada perpendicularmente al escenario de observación [1]. Esta descripción permite formular el problema de manera clara: dada una secuencia o conjunto de imágenes del cielo, se requiere separar computacionalmente la fracción asociada a nubes respecto al fondo visible y estimar, a partir del cambio temporal entre observaciones, un campo o una medida representativa del mo-

vimiento nuboso.

Formalmente, si se denota por  $I_t(x, y)$  la imagen del cielo en el instante  $t$  sobre un dominio discreto  $\Omega \subset \mathbb{Z}^2$ , entonces el problema consiste en estimar al menos dos tipos de variables:

1. una medida escalar de cobertura nubosa asociada a la fracción del dominio ocupada por nubes;
2. una medida vectorial o angular asociada al desplazamiento promedio aparente de estructuras nubosas entre instantes consecutivos.

La dificultad central del problema radica en que el cielo no es una escena rígida. Las nubes cambian de forma, densidad y textura, y la iluminación puede variar por razones independientes del movimiento. Por tanto, el desarrollo de SICUANU debe entenderse como una tentativa de extraer regularidad cuantitativa a partir de una escena altamente cambiante.



Figura 12.2: Ejemplo conceptual de secuencia de imágenes del cielo. La comparación entre cuadros permite estimar tanto la fracción nubosa como el desplazamiento aparente de las estructuras de nube en el tiempo.

### 12.3. Objetivo general y objetivos específicos

A partir de la descripción registral del software y de su finalidad funcional, el propósito general del desarrollo puede expresarse como la construcción de una herramienta computacional que cuantifique variables atmosféricas observables directamente en imágenes del cielo.

## Objetivo general

Desarrollar un software que permita cuantificar el porcentaje nuboso y estimar la dirección media y la velocidad aparente del movimiento de las nubes a partir de imágenes del cielo tomadas con una cámara convencional.

## Objetivos específicos

1. Adquirir imágenes de una porción del cielo bajo una configuración de cámara adecuada para su análisis.
2. Segmentar o discriminar computacionalmente regiones asociadas a nubosidad respecto al fondo del cielo.
3. Calcular una medida de cobertura nubosa como proporción del área observada.
4. Estimar la dirección media del desplazamiento aparente de las nubes a partir de variaciones temporales en la escena.
5. Consolidar los resultados en una aplicación de software usable por los grupos de investigación involucrados.

Estos objetivos muestran que el sistema no se reduce a una sola operación algorítmica. En realidad, articula adquisición, separación visual de regiones, análisis temporal y consolidación de resultados en una herramienta institucionalmente usable.

## 12.4. Modelo del problema y formulación básica

Sea  $I_t : \Omega \rightarrow \mathbb{R}^c$  una imagen digital del cielo adquirida en el instante  $t$ , donde  $\Omega$  representa el dominio de píxeles y  $c$  el número de canales de color. El objetivo inicial del sistema es estimar la fracción del dominio que corresponde a nubosidad.

Para ello, puede definirse una máscara binaria de nubes

$$B_t(x, y) = \begin{cases} 1, & \text{si el píxel } (x, y) \text{ es clasificado como nube,} \\ 0, & \text{en otro caso.} \end{cases}$$

A partir de esta máscara, el porcentaje nuboso  $C_t$  en el instante  $t$  puede expresarse como

$$C_t = 100 \frac{\sum_{(x,y) \in \Omega} B_t(x, y)}{|\Omega|},$$

donde  $|\Omega|$  representa el número total de píxeles de la región observada. Esta expresión

resume la idea central de cobertura: qué proporción del cielo visible se encuentra ocupada por nubes.

Para estudiar el movimiento aparente, considérese ahora un par de imágenes consecutivas  $I_t$  e  $I_{t+\Delta t}$ . Si a cada estructura nubosa o a cada punto relevante de la imagen puede asociársele un desplazamiento bidimensional

$$\mathbf{v}_k = (u_k, v_k),$$

entonces la dirección media del movimiento puede representarse mediante un vector promedio

$$\bar{\mathbf{v}} = \frac{1}{M} \sum_{k=1}^M \mathbf{v}_k,$$

donde  $M$  es el número de vectores de desplazamiento considerados.

El ángulo medio de dirección puede obtenerse a partir de dicho vector promedio como

$$\theta = \text{atan2}(\bar{v}, \bar{u}),$$

donde  $\bar{u}$  y  $\bar{v}$  son las componentes horizontal y vertical de  $\bar{\mathbf{v}}$ . De manera análoga, una magnitud media de velocidad aparente puede expresarse como

$$V = \|\bar{\mathbf{v}}\|.$$

Estas ecuaciones no pretenden describir todos los detalles internos del algoritmo de SICUANU, sino formalizar con claridad las variables que el registro del software afirma que el sistema calcula: porcentaje nuboso, ángulo de dirección y velocidad asociada al movimiento de las nubes [1].

## 12.5. Arquitectura funcional del sistema

A partir de los documentos disponibles, la arquitectura funcional de SICUANU puede entenderse como una secuencia de cuatro etapas. Primero, se adquiere una imagen o una secuencia de imágenes del cielo con una cámara convencional instalada en una orientación definida respecto a la porción de cielo observada. Segundo, se determina qué regiones de la imagen corresponden a nubes. Tercero, se calcula el porcentaje de cobertura de dichas regiones sobre el total del área observada. Cuarto, se analiza el desplazamiento temporal de las estructuras nubosas para estimar dirección media y velocidad aparente.

Esta arquitectura puede resumirse así:

1. adquisición de imágenes del cielo;
2. separación nube/fondo;
3. cuantificación de cobertura;
4. estimación del movimiento aparente.

Lo importante aquí es notar que SICUANU no se limita a un problema de segmentación estática. Su valor técnico radica en combinar una variable espacial —la cobertura— con una variable dinámica —el movimiento—. Esto lo acerca más a una herramienta de análisis atmosférico que a un simple clasificador de imágenes.

La Figura 12.3 resume esta lógica general. Incluso sin contar en esta tanda con un manual técnico extenso del software, el registro funcional disponible permite sostener con seguridad esta secuencia conceptual de operación.



Figura 12.3: Pipeline conceptual de SICUANU: adquisición de imágenes del cielo, discriminación de nubosidad, cálculo del porcentaje nuboso y estimación de dirección media y velocidad aparente del movimiento de las nubes.

## 12.6. Implementación y contexto institucional del desarrollo

Aunque la documentación anexa no aporta en esta tanda una memoria técnica detallada de librerías, clases o módulos internos del software, sí deja claramente establecido el estatus institucional del desarrollo. El certificado de registro ante la DNDA identifica a

SICUANU como una obra de software inédita, originaria y colectiva, registrada el 20 de agosto de 2021 bajo la partida 13-86-47, con titular patrimonial y productor Universidad de Antioquia, y con autoría atribuida a David Fernández Mc Cann, Mauricio Andrés Correa Ochoa, Laura Carolina Sandoval Acosta y Julián David Longas Arteaga [1].

La certificación de uso institucional, por su parte, señala que los grupos G-LIMA y GEPAR cuentan con licencia de uso del software otorgada por la Universidad de Antioquia [2]. Este hecho es importante para el libro, porque confirma que SICUANU no es solo una idea o un prototipo aislado, sino un desarrollo efectivamente reconocido y usado por los grupos académicos asociados.

Desde el punto de vista de su inserción en investigación aplicada, el capítulo también puede leerse como evidencia de una línea de trabajo donde la visión artificial se orienta hacia problemas de monitoreo ambiental. Si en otros capítulos el objeto de interés es agua, organismos o cobertura biológica, aquí el objeto es el cielo: una superficie visual cambiante cuyo análisis exige traducir textura, forma y desplazamiento en variables cuantitativas útiles.

## 12.7. Resultados esperados y utilidad del sistema

La salida principal de SICUANU, de acuerdo con el registro del software, consiste en una medida del porcentaje nuboso y en variables asociadas al movimiento medio de las nubes, entre ellas velocidad y ángulo de dirección [1]. Incluso esta descripción básica permite entender el valor práctico del desarrollo.

En primer lugar, el porcentaje nuboso proporciona una medida sintética del estado del cielo en un momento dado. En lugar de una apreciación cualitativa, el usuario obtiene un valor cuantificable y comparable entre observaciones. En segundo lugar, la dirección media y la velocidad aparente permiten incorporar una dimensión dinámica al análisis, haciendo posible describir no solo cuánto cielo está cubierto, sino también cómo se desplaza la nubosidad observada.

Desde una perspectiva aplicada, estas salidas pueden apoyar estudios de caracterización local del estado atmosférico, análisis de radiación incidente, seguimiento de variabilidad nubosa y procesos de monitoreo ambiental en campañas de observación. Desde una perspectiva metodológica, el desarrollo muestra que una cámara convencional puede convertirse, mediante software, en una fuente de variables ambientales cuantificables.

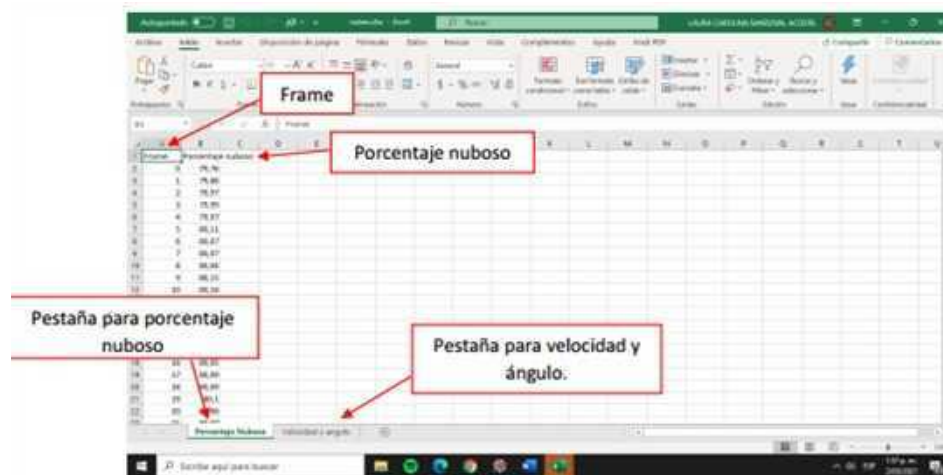


Figura 12.4: Esquema de resultados esperados en SICUANU: porcentaje de cobertura nubosa sobre la porción de cielo observada y vector medio de dirección/velocidad aparente del movimiento de las nubes.

## 12.8. Discusión

SICUANU resulta interesante porque traslada la visión artificial a una escena que, a primera vista, parece poco estructurada. El cielo no contiene objetos delimitados del mismo modo que una placa de Petri o una cama de plántulas; más bien presenta gradientes, texturas y formas difusas. Eso significa que cuantificar nubes no es simplemente detectar bordes, sino decidir cómo convertir una imagen atmosférica en una variable estable.

La combinación de porcentaje nuboso y movimiento aparente es conceptualmente fuerte. Muchas aproximaciones podrían limitarse a estimar cobertura, pero SICUANU incorpora además información dinámica. Esa decisión le da mayor valor científico al sistema, pues reconoce que la nubosidad no solo ocupa una fracción del cielo, sino que se desplaza y reorganiza en el tiempo.

Otra virtud del desarrollo es su accesibilidad instrumental implícita: el hecho de basarse en una cámara convencional sugiere una estrategia de bajo umbral de entrada para el monitoreo visual del cielo. Esto es especialmente valioso en contextos académicos o de investigación aplicada donde no siempre se dispone de sensores especializados de alto costo.

La principal limitación para este capítulo es documental: en esta tanda de insumos no contamos con un manual técnico completo del algoritmo, por lo que la reconstrucción metodológica debe mantenerse en un nivel prudente. Sin embargo, esta prudencia no

impide reconocer el valor del desarrollo. Los documentos disponibles permiten afirmar con seguridad la finalidad del software, sus variables de salida, su autoría, su estatus de registro y su uso institucional [1], [2]. Con ello se puede construir un capítulo fiel, serio y consistente con el espíritu del libro.

## 12.9. Conclusiones

Se presentó SICUANU como un sistema de cuantificación y aforo de nubes orientado a estimar porcentaje nuboso, dirección media y velocidad aparente del movimiento de las nubes a partir de imágenes del cielo tomadas con una cámara convencional [1]. El capítulo mostró que el desarrollo se ubica en la intersección entre visión artificial, monitoreo ambiental y análisis atmosférico, aportando una forma objetiva de convertir observaciones del cielo en variables cuantificables.

La formulación básica permitió expresar con claridad la lógica del problema: segmentar regiones nubosas, calcular su proporción respecto al dominio visual y estimar un desplazamiento medio aparente entre observaciones sucesivas. Aunque la documentación disponible no desarrolla exhaustivamente el algoritmo interno, sí sustenta de forma suficiente la finalidad funcional del software y su valor como herramienta de investigación aplicada.

Desde el punto de vista institucional, SICUANU cuenta con registro ante la DNDA y certificación de uso por parte de los grupos G-LIMA y GEPAR, lo que confirma su madurez como producto tecnológico universitario [1], [2]. Desde el punto de vista académico, representa una extensión natural de la línea Geolimna–GEPAR hacia problemas atmosféricos. Como trabajo futuro, sería deseable complementar este capítulo con una documentación técnica más detallada del algoritmo, sus criterios de segmentación y su procedimiento de estimación cinemática, de modo que el desarrollo pueda ser reproducido y evaluado con mayor profundidad.

## Agradecimientos

Se agradece a la Universidad de Antioquia y a los grupos G-LIMA y GEPAR por el contexto académico e institucional que dio origen a SICUANU, así como a los autores y participantes del desarrollo por su contribución a la construcción y consolidación del software [1], [2].

## Referencias

- [1] D. F. M. Cann, M. A. C. Ochoa, L. C. S. Acosta y J. D. L. Arteaga, *SICUANU: Sistema de Cuantificación y Aforo de Nubes*, Registro de soporte lógico–software, Dirección Nacional de Derecho de Autor, Libro–Tomo–Partida 13-86-47, fecha de registro: 20-ago.-2021, 2021.
- [2] M. A. C. Ochoa y D. S. F. M. Cann, *Certificación de uso del software SICUANU*, Certificación institucional de uso por parte de los grupos G-LIMA y GEPAR, Universidad de Antioquia, Medellín, 9 de diciembre de 2021, 2021.

# Capítulo 13

## Aplicación móvil para medición de variables ambientales en ecosistemas acuáticos

### Autores

John Fernando Ramírez Gómez<sup>1</sup>; Mateo Arroyave Cañas<sup>2</sup>; Fabio de Jesús Vélez Macías<sup>3</sup>; Néstor Jaime Aguirre Ramírez<sup>4</sup>; David Fernández Mc Cann<sup>5</sup>.  
Grupos Geolimna y Gepar, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

La obtención de variables fisicoquímicas en ecosistemas acuáticos suele requerir múltiples equipos independientes, operadores entrenados y condiciones de trabajo de campo que no siempre favorecen la seguridad del instrumento ni la continuidad del registro. En este

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. john.ramirez@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. mateo.arroyave@udea.edu.co

<sup>3</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. fabio.velez@udea.edu.co

<sup>4</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. nector.aguirre@udea.edu.co

<sup>5</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

<sup>1</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. fabio.velez@udea.edu.co

<sup>2</sup>Profesor titular, Departamento de Ingeniería Ambiental y Sanitaria. nector.aguirre@udea.edu.co

<sup>3</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

capítulo se presenta *SUCA*, una aplicación móvil Android desarrollada como parte de un sistema de medición superficial cautiva orientado a leer, acondicionar y almacenar datos ambientales enviados por un dispositivo electrónico multivariable a través de Bluetooth [1], [2]. El sistema fue concebido en el marco del proyecto SIMEVAM, asociado al seguimiento de variables ambientales en la ciénaga de Ayapel, y busca facilitar la caracterización de ecosistemas acuáticos de difícil acceso mediante una arquitectura portátil y robusta [1], [3]. La documentación disponible muestra una aplicación estructurada en modos de lectura, selección de variables y captura de datos, con almacenamiento en teléfono inteligente y exportación a hoja de cálculo [2]. En términos funcionales, el sistema permite medir al menos profundidad, temperatura, conductividad eléctrica, oxígeno disuelto y pH, mientras que la documentación técnica general también menciona velocidad del agua como parte de la arquitectura más amplia del dispositivo [1], [2]. El capítulo sintetiza el problema, la lógica de integración hardware–software, la formulación básica del sistema y su valor como herramienta de instrumentación ambiental portátil.

**Palabras clave:** monitoreo ambiental; Android; Bluetooth; ecosistemas acuáticos; sensores multivariables; ciénaga de Ayapel; instrumentación portátil; *SUCA*.

## 13.1. Introducción

La investigación ambiental en humedales, ciénagas y acuíferos depende con frecuencia de un conjunto de variables cuya medición simultánea resulta operativamente compleja. Profundidad, temperatura, conductividad eléctrica, pH y oxígeno disuelto no son, en la práctica de campo, simples números aislados: son piezas de una misma escena ambiental que idealmente deberían registrarse bajo las mismas condiciones espaciales y temporales. Sin embargo, el procedimiento convencional obliga muchas veces a transportar varios equipos, repetir maniobras de medición y exponer los instrumentos a ambientes hostiles, resbaladizos o de difícil acceso [1], [3].

En este contexto surge *SUCA*, una aplicación móvil que no se entiende de forma aislada, sino como parte de un sistema más amplio de sensado superficial cautivo. Su función consiste en recibir datos desde un dispositivo electrónico multivariable, visualizarlos en tiempo real, permitir la selección de variables y unidades, y almacenarlos localmente en un teléfono Android para uso posterior [2]. El sistema fue desarrollado para responder a necesidades reales de GEOLIMNA y GEPAR en el estudio de ecosistemas acuáticos, particularmente dentro del proyecto SIMEVAM, orientado al registro y seguimiento de variables ambientales en la ciénaga de Ayapel [1], [3].

Desde una mirada más amplia, *SUCA* representa un cambio de paradigma operativo. En

lugar de pensar cada variable como el dominio de un equipo independiente, el sistema busca integrar sensores, procesamiento embebido, comunicación inalámbrica y almacenamiento digital en una única plataforma de trabajo. Esta integración no solo reduce la carga instrumental del trabajo de campo, sino que también mejora la trazabilidad del dato y la ergonomía del proceso de medición.

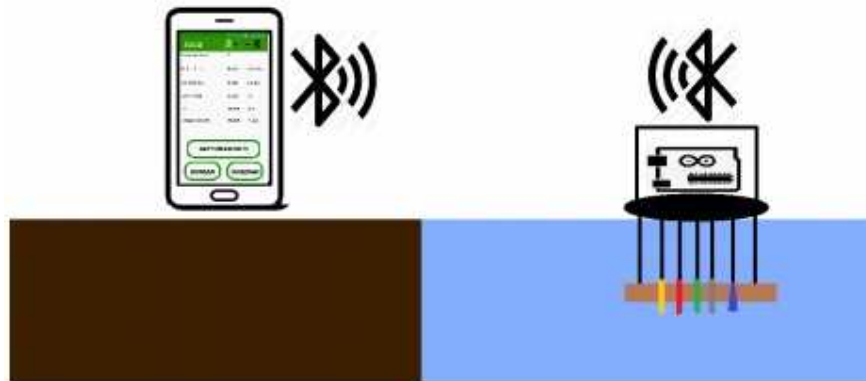


Figura 13.1: Contexto general de uso de SUCA en medición ambiental de ecosistemas acuáticos. El sistema integra sensores, electrónica de adquisición y aplicación móvil para facilitar campañas de campo en ambientes de difícil acceso.

### 13.2. Planteamiento del problema

El problema que da origen a SUCA puede describirse como una fragmentación instrumental del trabajo ambiental. Para caracterizar cuerpos de agua o acuíferos es común requerir varias variables fisicoquímicas y, con ello, varios equipos de medición, cada uno con sus propios procedimientos de lectura, calibración y almacenamiento. Esta fragmentación incrementa el tiempo de operación, dificulta la simultaneidad de las medidas y aumenta la vulnerabilidad del proceso cuando el entorno es inhóspito o de difícil acceso [1].

La solicitud de software es explícita en este punto: los sistemas de medición superficial cautiva suelen venir en formatos independientes para cada medición, mientras que SUCA permite concentrar la obtención de varias variables en un solo sistema, almacenando los datos en un celular y permitiendo acceso o transmisión en tiempo real [1]. Esta innovación no es meramente logística. En realidad, cambia la forma de relacionarse con el dato ambiental, porque lo vuelve más coherente, portátil y menos dependiente de la sincronización manual entre instrumentos.

Formalmente, el problema puede formularse así: se requiere un sistema capaz de adquirir y registrar un conjunto de variables ambientales  $\mathbf{x}(t)$  a partir de un dispositivo de sensado

superficial, transmitir las inalámbricamente hacia un teléfono inteligente y almacenarlas de manera estructurada para análisis posterior, minimizando el número de equipos, la exposición del hardware y la carga operativa del usuario. Si se representa el conjunto de variables medidas en un instante  $t$  mediante

$$\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_m(t)),$$

entonces el objetivo del sistema es garantizar que cada vector de observación sea recibido, visualizado y almacenado con consistencia temporal y con sus unidades de medición asociadas.

La documentación disponible reporta con claridad cinco variables en la interfaz de selección: profundidad, conductividad, temperatura, pH y oxígeno disuelto. Al mismo tiempo, la introducción técnica del manual menciona una arquitectura más amplia donde aparece también velocidad del agua. Por ello, en este capítulo se asume como núcleo seguro del sistema la medición de cinco variables en la app, dejando explícito que la arquitectura multivariable del dispositivo puede ser más amplia según la configuración del hardware.

### 13.3. Objetivo general y objetivos específicos

La documentación técnica del sistema plantea un objetivo general centrado en la construcción de una aplicación móvil Android como parte de un sistema de sensores multivariables que se comunican mediante Bluetooth [2].

#### Objetivo general

Realizar para los grupos de investigación GEPAR y GEOLIMNA una aplicación móvil para sistema operativo Android como parte del sistema de sensores multivariables, que se comunican mediante Bluetooth.

A partir de este objetivo general, el desarrollo persigue una lógica funcional clara: recibir los datos del sistema electrónico, mostrarlos al usuario, permitir configuraciones básicas de medición y almacenarlos de manera estructurada para su uso posterior.

## Objetivos específicos

1. Integrar en una aplicación móvil la lectura inalámbrica de variables ambientales provenientes de un sistema electrónico multivariable.
2. Permitir al usuario seleccionar variables y unidades de medición de acuerdo con las necesidades de la campaña.
3. Implementar modos de captura manual y automática para el almacenamiento de datos.
4. Guardar la información obtenida en el dispositivo móvil y exportarla a una hoja de cálculo.
5. Facilitar el trabajo de campo en ecosistemas acuáticos de difícil acceso mediante una herramienta portátil y robusta.

Estos objetivos revelan que SUCA no es solo una interfaz de visualización. En realidad, es una capa de mediación entre el sistema físico de sensores y el proceso posterior de análisis ambiental.

## 13.4. Arquitectura general del sistema

SUCA debe entenderse como uno de los componentes de una arquitectura mayor. La documentación técnica describe un sistema compuesto por un microcontrolador, múltiples sensores y un módulo de comunicación Bluetooth, cuyo bloque receptor es un teléfono Android donde corre la aplicación móvil [2]. Esta arquitectura integra, por tanto, cuatro niveles:

1. **Bloque de sensado:** conjunto de sensores ambientales.
2. **Bloque de adquisición y acondicionamiento:** microcontrolador y electrónica asociada.
3. **Bloque de transmisión:** módulo Bluetooth HC-05.
4. **Bloque de recepción, visualización y almacenamiento:** aplicación SUCA en Android.

Los planos electrónicos muestran explícitamente una configuración con Arduino Mega 2560, Bluetooth HC-05 y sensores asociados a profundidad, pH, oxígeno disuelto, conductividad eléctrica y temperatura, además del encapsulado y las conexiones requeridas para el sistema [2]. Esta descripción es importante porque evidencia que la aplicación no fue diseñada como software abstracto, sino como parte inseparable de una solución instrumental completa.

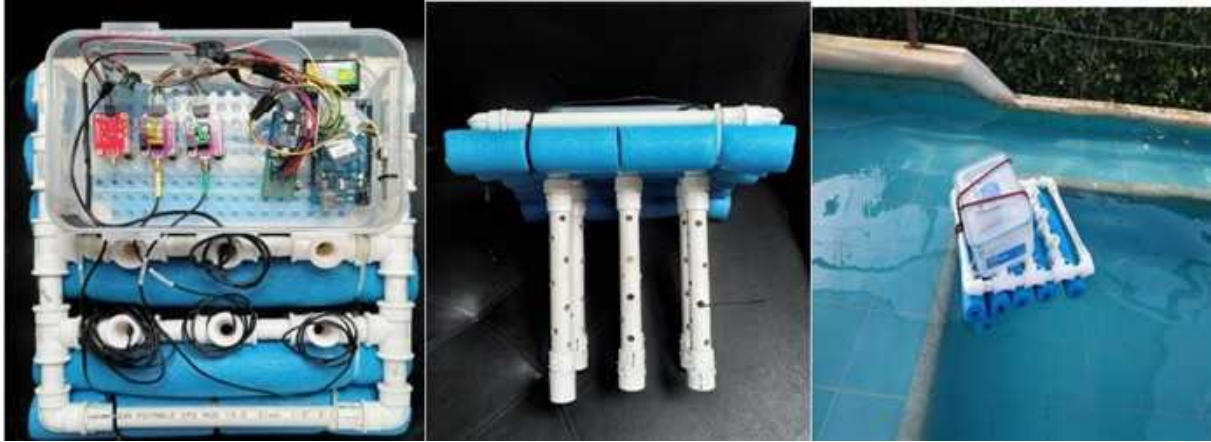


Figura 13.2: Arquitectura electrónica general del sistema SUCA con conexión Bluetooth. Se observan el microcontrolador, el módulo de comunicación y los sensores ambientales integrados al sistema de medición superficial cautiva.

### 13.5. Modelo funcional y formulación básica

Sea  $\mathbf{x}(t)$  el vector de variables recibido por la aplicación en el instante  $t$ :

$$\mathbf{x}(t) = (d(t), c(t), T(t), pH(t), OD(t)),$$

donde:

- $d(t)$  representa profundidad,
- $c(t)$  representa conductividad eléctrica,
- $T(t)$  representa temperatura,
- $pH(t)$  representa pH,
- $OD(t)$  representa oxígeno disuelto.

La aplicación permite asociar a cada componente una unidad seleccionada por el usuario. Por tanto, la medición efectiva puede representarse como un par ordenado

$$(\mathbf{x}(t), \mathbf{u}),$$

donde  $\mathbf{u}$  es el vector de unidades seleccionadas para cada variable. El manual muestra, por ejemplo, opciones de profundidad en cm, m, pulgadas o pies; temperatura en °C o °F; conductividad en  $\mu\text{S}/\text{cm}$  o  $1/\Omega\text{m}$ ; y oxígeno disuelto en % de saturación o mg/L [2].

El sistema soporta además dos modos de captura. En modo manual, cada muestra se

almacena solo cuando el usuario lo solicita. Si se denota por  $\tau_k$  el instante en que el usuario presiona el botón de captura, entonces la base de datos local queda formada por un conjunto discreto

$$\mathcal{D}_{\text{manual}} = \{\mathbf{x}(\tau_1), \mathbf{x}(\tau_2), \dots, \mathbf{x}(\tau_n)\}.$$

En modo automático, en cambio, la aplicación almacena las muestras de acuerdo con un intervalo programado, produciendo un conjunto

$$\mathcal{D}_{\text{auto}} = \{\mathbf{x}(t_0), \mathbf{x}(t_0 + \Delta t), \mathbf{x}(t_0 + 2\Delta t), \dots\}.$$

Esta formulación sencilla deja ver la lógica del sistema: SUCA no solo muestra datos, sino que los organiza como una secuencia de observaciones estructuradas y exportables.

## 13.6. Interfaz de usuario y lógica de operación

Uno de los rasgos más valiosos de SUCA es que su complejidad instrumental queda encapsulada en una experiencia de uso relativamente simple. La documentación del código reporta una secuencia de pantallas que comprende selección del modo de trabajo, establecimiento de la conexión Bluetooth, selección de sensores y unidades, elección del modo de captura y finalmente la interfaz de medición [2].

El sistema contempla, entre otras, las siguientes funcionalidades visibles para el usuario:

1. conexión Bluetooth con el dispositivo emisor;
2. selección de variables a mostrar;
3. selección de unidades de medición;
4. captura manual de datos;
5. captura automática con intervalo;
6. pausa, reinicio y borrado de datos;
7. carga de archivos y almacenamiento local.

La interfaz de selección de variables y unidades es especialmente importante, porque convierte la medición en una actividad configurable. No todas las campañas requieren las mismas variables ni las mismas unidades. Por ello, el software no impone un flujo rígido, sino que adapta la pantalla de medición a la selección hecha previamente por el usuario [2].



Figura 13.3: Interfaz de selección de variables y unidades en SUCA. El usuario puede escoger qué variables visualizar y en qué unidades recibirlas antes de iniciar la captura de datos.

Después de esta etapa, la aplicación conduce al modo de captura, donde el usuario decide entre operación manual o automática. Esta distinción es importante porque da flexibilidad al trabajo de campo: en algunos casos interesa registrar observaciones puntuales controladas por el operador; en otros, adquirir datos periódicos con mínima intervención humana [2]. La pantalla de medición muestra únicamente las variables seleccionadas, lo que mejora legibilidad y reduce carga visual.

## 13.7. Implementación de software y almacenamiento

La documentación técnica muestra que SUCA fue implementada en Java para Android, organizada en varias actividades como `MainActivity`, `ActivitySelectMode`, `MedirActivity`, `SelectSensor`, `modoMedicion`, `Measurements`, `ModeLeerActivity` y `CargarArchivo` [2]. Esta estructura modular es relevante porque deja ver que el software fue diseñado para crecer, incorporar funciones nuevas y mantener separadas las responsabilidades principales de conexión, lectura, selección y almacenamiento.

Desde el punto de vista operativo, la aplicación solicita permisos de lectura y escritura en almacenamiento interno, establece la conexión Bluetooth con dispositivos emparejados y procesa la cadena de datos recibida para distribuirla entre las variables activadas [2]. El

almacenamiento local de las mediciones y su exportación a una hoja de cálculo convierten al sistema en una herramienta de trazabilidad, no solo de visualización inmediata.

La solicitud de software enfatiza además varias propiedades de diseño: robustez en condiciones hostiles, operación en tiempo real, integridad del dato, extensibilidad y portabilidad a cualquier dispositivo Android. También menciona que el método para agregar nuevas características se describe en el manual técnico, lo que confirma que el sistema fue concebido como plataforma ampliable y no como aplicación cerrada de una sola versión [1].

### 13.8. Relación con el proyecto SIMEVAM

SUCA se desarrolló en el marco del proyecto SIIU 2020-37870 “Sistema de registro y seguimiento de variables ambientales ciénaga de Ayapel” (SIMEVAM) [1]. Esta relación es fundamental para entender el sentido del software. SUCA no nace como ejercicio abstracto de integración móvil, sino como respuesta a una necesidad concreta de monitoreo ambiental en un sistema cenagoso de alta importancia ecológica y social.

El documento del proyecto SIMEVAM describe la necesidad de registrar continuamente variables de la matriz agua, como altura de columna de agua, transparencia, temperatura, oxígeno disuelto, pH y conductividad eléctrica, en articulación con variables meteorológicas y con el pulso de inundación del sistema cenagoso de Ayapel [3]. En ese contexto, SUCA puede entenderse como una de las soluciones instrumentales que ayudan a cerrar la brecha entre necesidad de medición y disponibilidad de herramientas adecuadas para el terreno.

Esta conexión también refuerza el valor interdisciplinario del desarrollo. El software no se limita a informática móvil ni a electrónica aplicada; se inserta dentro de una pregunta ambiental más amplia: cómo medir mejor para comprender mejor la dinámica de un humedal tropical y sus procesos asociados.

### 13.9. Discusión

SUCA ilustra con claridad una de las apuestas más fértiles de la ingeniería aplicada al ambiente: integrar tecnologías existentes en una arquitectura nueva y útil. El carácter innovador del sistema no depende de inventar desde cero cada sensor o cada protocolo, sino de haber articulado móviles, Bluetooth, electrónica embebida y necesidades reales de campo en una solución operativamente superior a la suma de sus partes [1].

Una de sus principales fortalezas es la reducción de fragmentación instrumental. Al concentrar en una sola aplicación la recepción y almacenamiento de varias variables, disminuye la dependencia de equipos aislados y simplifica el trabajo de campo. Esto tiene consecuencias prácticas importantes: menor número de operadores, menor riesgo de daño en instrumentos expuestos, mayor coherencia temporal entre variables y mejor trazabilidad del dato recolectado.

También es destacable la forma en que la aplicación convierte la medición en una secuencia configurable. La selección de variables, las unidades y el modo de captura muestran que el sistema no fue diseñado para un único escenario rígido, sino para diferentes campañas y necesidades de observación. Esa flexibilidad es un rasgo de madurez del desarrollo.

Como limitación documental, debe reconocerse que los materiales anexos en esta tanda son más ricos en la descripción funcional y electrónica que en una exposición formal completa del algoritmo de acondicionamiento interno de los datos. Por ello, el capítulo adopta una formulación prudente, suficiente para expresar la lógica del sistema sin atribuir detalles no plenamente desarrollados en los documentos disponibles. Esa prudencia, sin embargo, no reduce el valor del capítulo: la finalidad, la arquitectura y la operación del sistema quedan claramente sustentadas por la documentación entregada. :contentReference[oaicite:9]index=9 :contentReference[oaicite:10]index=10

### 13.10. Conclusiones

Se presentó SUCA como una aplicación móvil para Android integrada a un sistema de medición superficial cautiva de variables ambientales en ecosistemas acuáticos, desarrollada para recibir, visualizar y almacenar datos provenientes de un dispositivo multivariable comunicado por Bluetooth [1], [2]. El capítulo mostró que el valor del desarrollo no radica solo en la lectura inalámbrica de sensores, sino en la construcción de una herramienta coherente para trabajo de campo, capaz de reducir fragmentación instrumental y mejorar la trazabilidad de la información.

La formulación propuesta permitió expresar la lógica funcional del sistema como una secuencia de observaciones multivariadas estructuradas, asociadas a unidades configurables y almacenadas en modo manual o automático. Esta representación ayuda a entender que SUCA no es simplemente una app de despliegue, sino un componente esencial del proceso de medición ambiental.

Desde el punto de vista aplicado, el desarrollo fortalece campañas de caracterización en humedales y acuíferos al reducir el número de equipos requeridos y permitir una operación más segura y eficiente. Desde el punto de vista académico, integra electrónica, comuni-

caciones, desarrollo móvil e instrumentación ambiental en un mismo proyecto. Desde el punto de vista institucional, se articula con SIMEVAM y con las necesidades reales de GEOLIMNA y GEPAR en la ciénaga de Ayapel.

Como trabajo futuro, sería deseable consolidar una memoria técnica específica del procesamiento interno de las señales, verificar y fijar el dato de propiedad intelectual correspondiente a SUCA, y explorar extensiones hacia sincronización remota, georreferenciación y monitoreo continuo en red.

## Agradecimientos

Se agradece a la Universidad de Antioquia y a los grupos de investigación GEOLIMNA y GEPAR por el contexto académico y ambiental que hizo posible el desarrollo de SUCA, así como al proyecto SIMEVAM por proporcionar el marco de investigación donde esta solución instrumental adquirió sentido y aplicación [1], [3], [4].

## Referencias

- [1] D. S. F. M. Cann, F. de Jesús Vélez Macías y N. A. Ramírez, *Solicitud para asignación de puntos por desarrollo de software: SUCA - Aplicación móvil de sistema de medición superficial cautiva*, Documento institucional, Universidad de Antioquia, 13 de diciembre de 2021, 2021.
- [2] Grupos Gepar y Geolimna, *SUCA: Hardware receptor y emisor de datos para sistema de medición superficial cautiva. Planos de diseño electrónico y documentación técnica*, Universidad de Antioquia, Medellín, Colombia, 2021.
- [3] Universidad de Antioquia, *Sistema de registro y seguimiento de variables ambientales ciénaga de Ayapel (SIMEVAM)*, Documento de proyecto SIIU 2020-37870, 2020.
- [4] F. de Jesús Vélez Macías y D. S. F. M. Cann, *Certificación de uso del software SUCA*, Certificación institucional de uso por parte de los grupos GEOLIMNA y GEPAR, Universidad de Antioquia, 9 de diciembre de 2021, 2021.

# Capítulo 14

## Identificación automática de irregularidades en pavimento

### Autores

Luis David Flórez Pareja<sup>1</sup>; Juan Pablo Escobar Arenas<sup>2</sup>; David Fernández Mc Cann<sup>3</sup>.  
Grupo Gepar y Geolimna, Facultad de Ingeniería, Universidad de Antioquia

### Resumen

La evaluación del estado del pavimento continúa realizándose en numerosos contextos mediante inspección visual manual, un procedimiento lento, subjetivo y vulnerable al error humano. En este capítulo se presenta *IDIRPA*, un software desarrollado para la identificación y cuantificación automática de irregularidades en pavimento —particularmente grietas, baches y piel de cocodrilo— a partir de imágenes digitales, mediante técnicas clásicas de procesamiento digital de imágenes [1], [2], [3]. El método se estructura en tres etapas funcionales dentro del software: extracción de métrica, transformación de perspectiva y extracción de características, y en el artículo publicado se formaliza además como una secuencia de once pasos que incluye filtrado promedio, binarización por triángulo, extracción de bordes mediante Canny y proyección de resultados sobre la imagen original [2], [3]. El sistema entrega medidas reales como área y perímetro, resaltado visual del

---

<sup>1</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.florez@udea.edu.co

<sup>2</sup>Estudiante, Departamento de Ingeniería Electrónica y Telecomunicaciones. juan.escobar@udea.edu.co

<sup>3</sup>Profesor titular, Departamento de Ingeniería Electrónica y Telecomunicaciones. david.fernandez@udea.edu.co

defecto y almacenamiento de resultados, convirtiendo una fotografía ordinaria del pavimento en un insumo cuantitativo útil para inspección y priorización de intervenciones [2], [4]. El artículo asociado reporta además la evaluación del método mediante la métrica de intersección sobre unión (IOU), con valores de 0.69, 0.87 y 0.79 para irregularidades como piel de cocodrilo, bache y grieta, respectivamente [3]. El capítulo sintetiza el problema, la formulación del método, su arquitectura de software y la relación entre el desarrollo tecnológico y su validación académica posterior.

**Palabras clave:** pavimento; grietas; baches; piel de cocodrilo; procesamiento digital de imágenes; transformación de perspectiva; intersección sobre unión; IDIRPA.

## 14.1. Introducción

El pavimento es una interfaz crítica entre infraestructura, movilidad y seguridad. Aunque su deterioro es un proceso esperado en el tiempo, la manera en que se detecta, documenta y cuantifica sigue condicionando la oportunidad y la calidad de las intervenciones. En muchos escenarios, la identificación de grietas, baches y piel de cocodrilo continúa dependiendo de inspecciones visuales realizadas por personal técnico, lo que convierte la evaluación en una tarea intensiva en tiempo, sensible a la experiencia del observador y propensa a variaciones subjetivas [3], [4].

En este contexto surge *IDIRPA*, un software orientado a transformar fotografías de pavimento en descripciones cuantitativas de irregularidades. Su valor radica en que no se limita a detectar visualmente la existencia de un defecto, sino que organiza un flujo de trabajo capaz de convertir la imagen en medidas reales como área, perímetro y diámetro equivalente, además de producir evidencias gráficas del defecto resaltado sobre la escena original [2], [3]. Esta capacidad es especialmente útil en contextos donde las concesiones viales y los entes de control requieren respaldo fotográfico y criterios cuantitativos para seguimiento y toma de decisiones [1].

El interés del desarrollo aumentó cuando el trabajo dio lugar a una publicación científica posterior, donde se sistematizó la metodología y se evaluó su desempeño mediante la métrica de intersección sobre unión. Esta articulación entre software y artículo publicado es importante para el libro del seminario, porque muestra un caso claro en el que un desarrollo tecnológico universitario no solo se construye y registra, sino que además se formaliza, discute y valida en un medio académico :contentReference[oaicite:2]index=2.



Figura 14.1: Contexto general del problema abordado por IDIRPA. A partir de imágenes de pavimento, el sistema identifica y cuantifica irregularidades como grietas, baches y piel de cocodrilo, con fines de inspección y apoyo a mantenimiento vial.

## 14.2. Planteamiento del problema

La detección de irregularidades en pavimento es una tarea necesaria para planificar mantenimiento y reparación de vías. Sin embargo, la medición manual resulta costosa tanto en tiempo como en recursos, y además introduce una componente subjetiva en la evaluación del daño [3]. El problema no es únicamente reconocer si existe una grieta o un bache, sino estimar con suficiente objetividad la extensión afectada y producir evidencia visual que respalde la valoración técnica.

La documentación del software y el artículo publicado coinciden en este punto: IDIRPA busca cuantificar irregularidades como grietas, baches y piel de cocodrilo a partir de imágenes de pavimento, ofreciendo resultados como área, perímetro y resaltado del defecto sobre la imagen original [2], [3]. La solicitud institucional añade que estos resultados pueden ser utilizados por inspectores viales como evidencia cuantitativa de la afectación, dentro de escenarios de supervisión y seguimiento de la calidad del pavimento [1].

Formalmente, sea  $I : \Omega \rightarrow \mathbb{R}^c$  una imagen digital de una superficie de pavimento. El objetivo del sistema es identificar una región  $R \subset \Omega$  asociada a una irregularidad, segmentarla del fondo y estimar propiedades geométricas reales de dicha región. Para lograrlo, el método requiere una relación métrica obtenida a partir de un objeto de referencia en la imagen, así como una transformación de perspectiva que permita trabajar sobre una representación más regular de la región de interés [2], [3].

La dificultad central radica en que las irregularidades suelen ser oscuras respecto al pavimento sano, pero no son las únicas regiones oscuras presentes en una escena. Sombras, manchas y pequeñas rocas incrustadas pueden confundirse con defectos reales, y el propio artículo señala que factores como la luz y la sombra influyen en los resultados del algoritmo [3]. Por tanto, IDIRPA debe entenderse como una solución eficiente y práctica dentro de un dominio de validez bien definido, no como un detector universal exento de condiciones de captura.

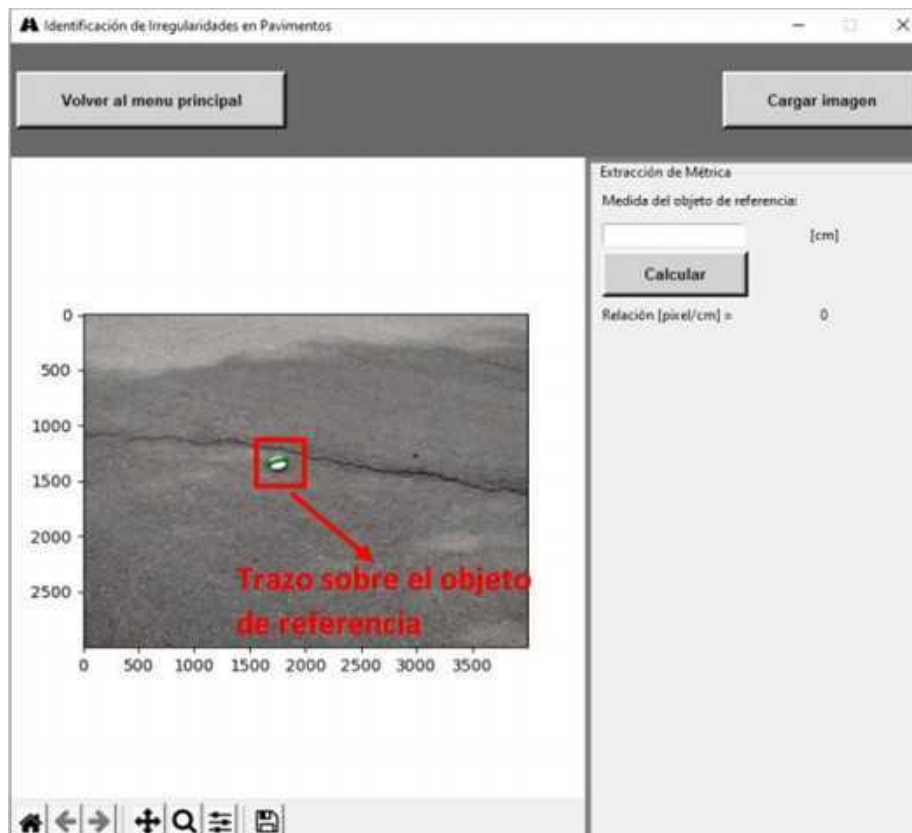


Figura 14.2: Ejemplo de imagen de entrada con una irregularidad en pavimento y un objeto de referencia. El objeto testigo permite establecer la relación métrica necesaria para convertir píxeles en unidades reales.

### 14.3. Objetivo general y objetivos específicos

La documentación técnica del software y el artículo publicado permiten expresar con claridad la intención del desarrollo: construir un método de procesamiento digital de imágenes que permita detectar irregularidades sobre pavimento y cuantificarlas en términos físicamente interpretables [3], [4].

## Objetivo general

Aplicar técnicas de procesamiento digital de imágenes para el diseño e implementación de un algoritmo que permita la identificación automática de irregularidades sobre pavimento a partir de fotografías, posibilitando su posterior cuantificación.

## Objetivos específicos

1. Determinar la relación métrica entre píxeles y longitud real a partir de un objeto de referencia.
2. Corregir la perspectiva de la región de interés mediante una transformación proyectiva.
3. Segmentar irregularidades como grietas, baches y piel de cocodrilo usando técnicas clásicas de procesamiento de imágenes.
4. Calcular características geométricas reales como área, perímetro y diámetro equivalente.
5. Consolidar el método en un software útil, portable y trazable para inspección vial.

### 14.4. Modelo del problema y formulación básica

La metodología formalizada en el artículo publicado puede describirse como una cadena de once pasos que transforma la imagen original en una representación cuantificable de la irregularidad [3]. La primera etapa consiste en obtener una relación métrica entre píxeles y dimensión real a partir de un testigo presente en la imagen. Si  $d_p$  es la longitud en píxeles del testigo y  $d_r$  su longitud real, entonces la relación métrica puede escribirse como

$$\mu = \frac{d_p}{d_r},$$

donde  $\mu$  representa la razón píxel por unidad de longitud.

Después, se selecciona una región de interés y se corrige su perspectiva. Si  $Q$  es el cuadrilátero definido sobre la imagen y  $T$  la transformación proyectiva correspondiente, entonces la imagen corregida puede representarse como

$$f'(x, y) = T[f(x, y)].$$

Sobre esta imagen corregida se realiza una conversión a escala de grises,

$$f'(x, y) \in [r, g, b] \rightarrow [g],$$

seguida de un filtrado promedio para homogeneizar la textura. El artículo expresa esta etapa como

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f'(x + s, y + t),$$

donde  $w(s, t)$  es el kernel del filtro promedio [3].

La binarización se realiza mediante el algoritmo del triángulo, obteniendo una máscara

$$h(x, y) = \begin{cases} 1, & \text{si el píxel pertenece a la irregularidad,} \\ 0, & \text{en otro caso.} \end{cases}$$

A partir de esta máscara, el área en píxeles se define como

$$A_p = \sum_{(x,y) \in R} h(x, y),$$

y luego se convierte a unidades reales con la métrica obtenida previamente:

$$A_r = \frac{A_p}{\mu^2}.$$

El artículo reporta además la extracción de bordes mediante el método de Canny,

$$j(x, y) = \text{Canny}[h(x, y)],$$

y su proyección inversa sobre la imagen original para resaltar visualmente la irregularidad [3]. En la práctica, esta secuencia transforma una fotografía ordinaria en una escena donde la región afectada aparece delimitada y acompañada de medidas físicas útiles.

El mismo artículo añade que, para ciertas irregularidades como baches y piel de cocodrilo, el cálculo del perímetro y del diámetro equivalente puede realizarse sobre la región rellenada delimitada por el contorno, usando expresiones como

$$D = \sqrt{\frac{4A}{\pi}}, \quad P = \pi D,$$

cuando se aproxima la región a un círculo equivalente [3]. Esta decisión metodológica es

importante porque conecta la segmentación con descriptores geométricos interpretables en contexto de mantenimiento vial.

## 14.5. Metodología funcional y secuencia de uso

El manual de usuario organiza el uso de IDIRPA en tres módulos secuenciales: extracción de métrica, extracción de perspectiva y extracción de características. Esta estructura coincide conceptualmente con la metodología detallada en el artículo publicado, aunque el artículo la desarrolla en mayor profundidad y la expande a once pasos de procesamiento [2], [3].

La secuencia funcional del software puede resumirse así:

1. cargar una imagen original con un objeto de referencia;
2. seleccionar dos puntos sobre el objeto para establecer la métrica;
3. seleccionar cuatro puntos sobre la región de interés para corregir perspectiva;
4. cargar la imagen ya transformada;
5. elegir el tipo de irregularidad;
6. ejecutar el procesamiento y consultar las medidas resultantes.

Esta estructura guiada es una fortaleza del sistema, porque traduce una metodología relativamente técnica en una experiencia de uso clara para el usuario final. Además, el manual enfatiza que la ejecución debe hacerse obligatoriamente en ese orden, lo cual muestra que la secuencia no es arbitraria, sino parte esencial de la lógica del método [2].



Figura 14.3: Menú principal de IDIRPA. El software organiza su funcionamiento en tres módulos secuenciales: extracción de métrica, extracción de perspectiva y extracción de características.

## 14.6. Resultados y validación del método

El aporte del artículo publicado es especialmente importante en esta sección, porque añade una validación cuantitativa del método que no aparece con el mismo detalle en los manuales. En particular, el desempeño del algoritmo se evalúa mediante la métrica de intersección sobre unión (IOU), definida como la razón entre el área de intersección y el área de unión entre una máscara real y una máscara de predicción [3]:

$$IOU = \frac{\text{Área de intersección}}{\text{Área de unión}}.$$

Para la evaluación, los autores capturaron cinco imágenes de pavimento con los tres tipos de deterioro considerados y compararon máscaras reales obtenidas manualmente con las máscaras de predicción generadas por el algoritmo. El artículo reporta valores de 0.69, 0.87 y 0.79 para irregularidades como piel de cocodrilo, bache y grietas, respectivamente, y en la tabla del artículo también se observan valores individuales de 0.69, 0.87, 0.79, 0.87 y 0.81 para las cinco imágenes analizadas :contentReference[oaicite:3]index=3.

Estos resultados son relevantes por dos razones. Primero, muestran que el software no

solo produce salidas visualmente plausibles, sino que fue sometido a una comparación explícita contra máscaras de referencia. Segundo, evidencian que una metodología basada exclusivamente en procesamiento digital de imágenes clásico puede alcanzar resultados aceptables sin recurrir necesariamente a flujos de entrenamiento propios de machine learning o deep learning [3].

La Figura 14.4 puede usarse para mostrar esta lógica de resultados: resaltado del defecto, máscara generada y medidas geométricas extraídas. En el libro, esta figura ayuda a conectar el flujo metodológico con la salida que finalmente observa el usuario.

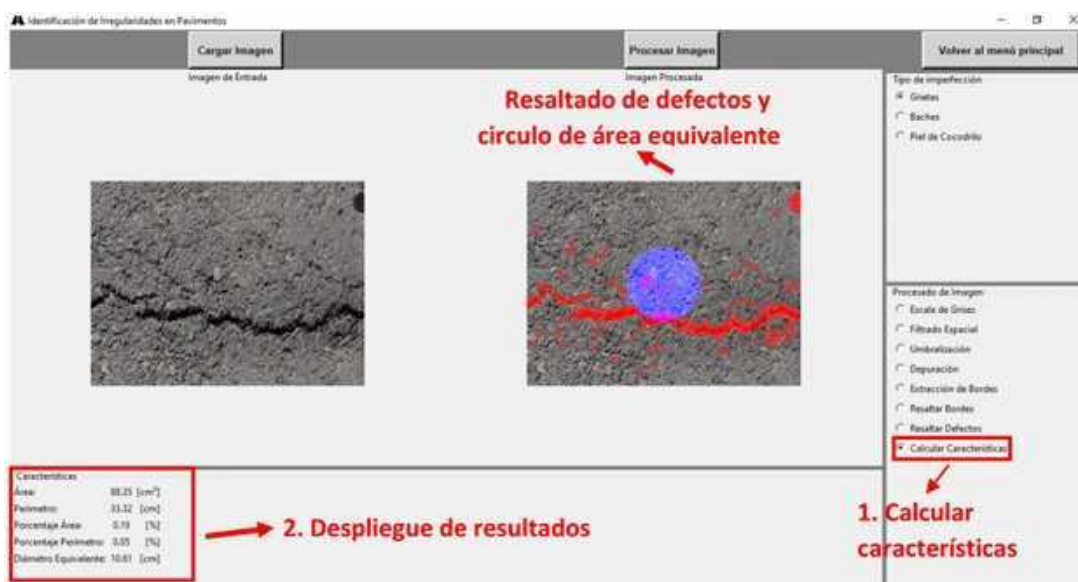


Figura 14.4: Ejemplo de resultados en IDIRPA. El sistema resalta el defecto sobre la imagen original, permite visualizar etapas intermedias del procesamiento y entrega medidas geométricas útiles para cuantificar la afectación del pavimento.

## 14.7. Implementación del software

La documentación técnica indica que IDIRPA fue desarrollado en Python 3.9.2 usando Spyder 4.1.5 del paquete Anaconda y librerías como OpenCV, NumPy, SciPy, Matplotlib, PIL, imutils, Pandas y Tkinter [4]. El sistema se estructura en varios módulos, entre ellos `IDIRPA.py`, `IrregularidadesPavimento.py`, `bwAreaOpen.py`, `RellenoDeContorno.py` y `byteIMG.py`.

Esta arquitectura modular tiene valor práctico porque separa la lógica de interfaz de la lógica algorítmica. El manual técnico muestra además cómo agregar nuevas funcionalidades al sistema, lo que confirma que el software fue concebido con una estructura extensible

[4]. La solicitud institucional refuerza esta idea al describir el sistema como robusto, extensible y portable sobre Windows o Linux, y útil para analistas de fallas en pavimento y empresas dedicadas a este tipo de estudios [1].

En retrospectiva, el artículo publicado confirma que esta arquitectura ligera y basada en procesamiento clásico tiene una ventaja adicional: evita etapas de entrenamiento, etiquetado y ajuste de hiperparámetros propias de enfoques de machine learning y deep learning, reduciendo complejidad computacional y favoreciendo velocidad de ejecución [3].

### 14.8. Discusión

La incorporación del artículo publicado permite leer IDIRPA con mayor profundidad. Ya no se trata solo de un software funcional documentado por sus autores, sino de un método que fue posteriormente sistematizado y sometido a validación académica. Esta relación entre desarrollo y publicación fortalece el valor del capítulo, porque permite sostener con evidencia externa al manual que el sistema produce resultados comparables con máscaras reales mediante una métrica explícita como IOU [3].

Otro punto importante del artículo es la defensa del procesamiento digital de imágenes clásico frente a soluciones más pesadas basadas en machine learning o deep learning. Los autores argumentan que la aproximación propuesta simplifica el flujo de desarrollo, reduce requerimientos computacionales y evita problemas de pérdida de generalización asociados al tiempo y a la variabilidad futura de los datos [3]. Este argumento es coherente con la filosofía de varios desarrollos del seminario Geolimna–GEPAR: cuando un problema puede resolverse bien con visión clásica y geometría explícita, esa solución puede ser más estable, interpretable y portable.

Naturalmente, el método tiene límites. El propio artículo y el manual reconocen que la luz, las sombras, las rocas oscuras incrustadas y las deformaciones producidas por algunas operaciones del algoritmo pueden hacer que la máscara predicha no coincida completamente con la real [3], [4]. Lejos de ser una debilidad narrativa, esta admisión fortalece la credibilidad del desarrollo, porque explicita las condiciones bajo las cuales la herramienta funciona mejor.

En conjunto, IDIRPA muestra un caso ejemplar de maduración tecnológica: parte como software universitario documentado y registrado, y luego se consolida también como artículo científico donde la metodología y los resultados quedan expuestos, discutidos y medidos.

## 14.9. Conclusiones

Se presentó IDIRPA como un software para la identificación automática de irregularidades en pavimento mediante procesamiento digital de imágenes, orientado a defectos como grietas, baches y piel de cocodrilo [1], [3], [4]. El capítulo mostró que el sistema organiza su funcionamiento en tres etapas principales dentro de la interfaz —métrica, perspectiva y características—, pero que a nivel metodológico puede describirse como una cadena más rica de once pasos, formalizada posteriormente en el artículo publicado.

La incorporación del paper fortalece especialmente la dimensión científica del capítulo, ya que aporta una métrica explícita de validación basada en intersección sobre unión y reporta resultados cuantitativos aceptables para los tipos de deterioro analizados. De este modo, el capítulo ya no descansa solo en la documentación técnica interna, sino también en una publicación que respalda el método y su evaluación.

Desde el punto de vista aplicado, IDIRPA ofrece una herramienta útil para inspección vial, priorización de mantenimiento y documentación objetiva del deterioro del pavimento. Desde el punto de vista académico, integra geometría proyectiva, filtrado, umbralización, extracción de bordes e interfaz gráfica en un desarrollo ligero y extensible. Desde el punto de vista institucional, cuenta con certificación de uso por parte del grupo GEPAR y con registro ante la DNDA bajo la partida 13-87-277 [5], [6].

Como trabajo futuro, el mismo artículo sugiere explorar enfoques de machine learning y deep learning, así como herramientas de procesamiento más robustas frente a variaciones de luminosidad y escenarios de captura más controlados [3]. Esa proyección confirma que IDIRPA no es un cierre, sino un punto de partida para nuevas iteraciones metodológicas.

## Agradecimientos

Se agradece a la Universidad de Antioquia y al grupo de investigación GEPAR por el contexto académico e institucional que hizo posible el desarrollo de IDIRPA, así como a los autores y colaboradores que participaron en su construcción, documentación, validación y posterior publicación científica [1], [3], [5].

## Referencias

- [1] D. S. F. M. Cann, *Solicitud para asignación de puntos por desarrollo de software: IDIRPA*, Documento institucional, Grupo GEPAR, Universidad de Antioquia, 13 de diciembre de 2021, 2021.
- [2] Universidad de Antioquia, *IDIRPA: Software para la identificación de irregularidades en pavimento. Manual de usuario*, Universidad de Antioquia, Medellín, Colombia, 2021.
- [3] J. P. Escobar-Arenas, L. D. Flórez-Pareja y D. S. Fernandez-Mc Cann, “Estimación de irregularidades en pavimentos mediante técnicas de procesamiento digital de imágenes,” *Revista Politécnica*, vol. 19, n.º 37, págs. 20-28, 2023. DOI: 10.33571/rpolitec.v19n37a2.
- [4] Universidad de Antioquia, *IDIRPA: Software para la identificación de irregularidades en pavimento. Manual técnico*, Universidad de Antioquia, Medellín, Colombia, 2021.
- [5] D. S. F. M. Cann, *Certificación de uso del software IDIRPA*, Certificación institucional de uso por parte del grupo GEPAR, Universidad de Antioquia, Medellín, 16 de diciembre de 2021, 2021.
- [6] D. F. M. Cann, L. D. F. Pareja y J. P. E. Arenas, *IDIRPA: Software para la identificación de irregularidades en pavimento*, Registro de soporte lógico–software, Dirección Nacional de Derecho de Autor, Libro–Tomo–Partida 13-87-277, fecha de registro: 17-nov.-2021, 2021.

# *Investigaciones 2019–2021*

*Grupos de investigación GEOLIMNA y GEPAR*



Este libro reúne una selección de *desarrollos, investigaciones y soluciones tecnológicas construidas* entre 2019 y 2021 en la convergencia entre *vision artificial, instrumentación, analítica de datos, automatización y monitoreo ambiental*.

Los capítulos aquí reunidos muestran *cómo problemas reales del entorno pueden traducirse en sistemas medibles, algoritmos verificables, dispositivos funcionales y herramientas útiles* para la investigación, la *docencia y la transferencia de conocimiento*.

La obra refleja una forma de trabajo interdisciplinaria en la que la ingeniería dialoga con la *limnología, la ecología, la agricultura de precisión, la infraestructura y el análisis ambiental*. En ese cruce, los grupos GEOLIMNA y GEPAR *han consolidado una línea de producción académica y tecnológica orientada tanto a la comprensión de fenómenos como a la construcción de soluciones concretas*.

**Universidad de Antioquia**

**Facultad de Ingeniería**

**Grupos de investigación GEOLIMNA y GEPAR**

**Medellín, Colombia**

**2026**

*Espacio reservado para ISBN, código de barras o sello editorial*