



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**AUTOMATIZACIÓN DEL ANÁLISIS CONDUCTUAL  
DE CRUSTÁCEOS ACUÁTICOS**

**TESIS**

Que para obtener el título de

**Ingeniero Eléctrico Electrónico**

**P R E S E N T A**

Mario Eduardo Sepúlveda Hernández

**DIRECTORA DE TESIS**

Dra. Karina Mendoza Ángeles



**Ciudad Universitaria, Cd. Mx., 2024**



*A la Dra. Karina Mendoza Ángeles, por disponer de su tiempo y enseñanzas para la realización de este manuscrito y confiar en mí*

*Al Dr. Jesús Hernández Falcón y al laboratorio de Redes Neuronales de la Facultad de Medicina*

*A mis padres, María del Rosario Hernández Arreola y Mario Sepúlveda Cruz por su apoyo y paciencia todos estos años*

*A Clara Analleli Hernández Arreola y Marco Antonio Jiménez Ramírez*

*A mi hermana y hermanos, Adriana Berenice Sepúlveda Hernández, Elías Sepúlveda Hernández y Aldo Patricio Sepúlveda Hernández  
A mis abuelos Inocencio Hernández Montero y a Elías Sepúlveda Arriaga.*

*A mis compañeras y compañeros de laboratorio Aidee, Iván, Laura, Mireya y Rodrigo*

*A Josseline, Karen y Fernando por su amistad en los últimos semestres de la carrera*

*A Natalia*

*A la UNAM y a la Facultad de Ingeniería por darme todas las herramientas para desarrollar esta tesis*

*"The world be reminding me it's danger"  
Kendrick Lamar, 2022*

*A Rosa Arreola Servín y Eulalia Cruz Aldana*





# Reconocimientos

---

Investigación realizada gracias al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) de la UNAM IN228523. Agradezco a la DGAPA-UNAM la beca recibida



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	3
1.2. Justificación . . . . .	5
1.3. Transformada de Hough . . . . .	5
1.4. Transformada Hough para detectar círculos . . . . .	7
1.4.1. Radio $r$ conocido . . . . .	7
1.4.2. Radio $r$ desconocido . . . . .	8
1.5. Búsqueda de círculos bajo el agua . . . . .	
1.6. Histograma de una imagen . . . . .	10
1.6.1. Expansión del histograma . . . . .	15
1.6.2. Ecuilización del histograma . . . . .	16
1.6.3. Igualación de histogramas . . . . .	17
<b>2. Objetivos</b>	<b>19</b>
2.1. Objetivo general . . . . .	19
2.2. Objetivos particulares . . . . .	19
<b>3. Materiales y métodos</b>	<b>21</b>
3.1. Marcas de pintura . . . . .	21
3.2. Obtención de registros . . . . .	22
<b>4. Desarrollo del problema</b>	<b>25</b>
4.1. Segmentación de videos para análisis individual y colectivo de imágenes . . . . .	25
4.1.1. Proceso general . . . . .	29
4.1.2. Negativo de la imagen . . . . .	31
4.1.3. Selección de áreas de estudio . . . . .	32

4.2.	Detección de círculos en puntos de interés . . . . .	34
4.2.1.	Trayectoria . . . . .	34
4.2.2.	Posición de lado . . . . .	40
4.2.3.	Hipnograma . . . . .	44
<b>5.</b>	<b>Resultados</b>	<b>47</b>
5.1.	Validación del hipnograma . . . . .	47
5.2.	Validación de trayectoria . . . . .	50
<b>6.</b>	<b>Discusión de resultados</b>	<b>53</b>
<b>7.</b>	<b>Conclusiones</b>	<b>57</b>
<b>8.</b>	<b>Trabajo a futuro</b>	<b>59</b>
<b>A.</b>	<b>Hipnogramas automatizados</b>	<b>61</b>
<b>B.</b>	<b>Anatomía externa del acocil</b>	<b>65</b>
<b>C.</b>	<b>Archivos obtenidos del programa</b>	<b>67</b>
<b>D.</b>	<b>Manual de usuario</b>	<b>69</b>
D.1.	Archivo leer.m . . . . .	69
D.2.	Programa principal . . . . .	71
D.2.1.	Detección de círculos para luz roja y blanca . . . . .	72
D.2.2.	Número de acociles de análisis . . . . .	72
D.2.3.	Número de carpetas de análisis . . . . .	72
D.2.4.	Umbral de binarización . . . . .	73
D.3.	Programa ejecutándose . . . . .	73
<b>E.</b>	<b>Código</b>	<b>77</b>
E.1.	Funciones . . . . .	85
	<b>Referencias</b>	<b>109</b>

# Capítulo 1

## Introducción

---

El estudio de algunas funciones neurobiológicas usualmente busca relacionar la actividad cerebral con una conducta específica. Una función de interés es el sueño (**Miyazaki et al., 2017**).

El sueño se ha descrito mediante una serie de condiciones que se presentan cuando un animal duerme, entre ellas (**Piéron, 1913**):

- Actividad cerebral específica y distinta de la de vigilia
- Postura estereotípica
- Reducción de actividad motora
- Un lugar preferido

En mamíferos, y algunos invertebrados (**Miyazaki et al., 2017**), esta conducta ha sido ampliamente estudiada y caracterizada, de manera que se puede distinguir la gran diversidad de posturas estereotípicas que presenta cada especie.

En la Figura 1.1 se muestran 4 ejemplos de cómo se presenta la postura estereotípica y su área preferida para descansar en diferentes animales e insectos; es por eso que resulta interesante el estudio de esta conducta.

Si bien el sueño ya ha sido ampliamente explorado en mamíferos, su estudio en invertebrados ha sido escaso; por ejemplo, se ha descrito en la mosca de la fruta (**Hendricks et al., 2000**), la abeja (**Helfrich-Förster, 2018**), el pulpo (**Pophale et al., 2023**) y el acocil (**Ramón et al., 2004**).

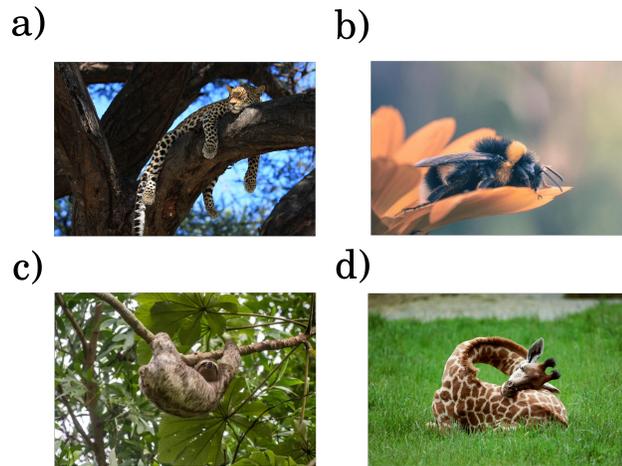


Figura 1.1: Posturas estereotípicas para distintas especies como **a)** Leopardo descansando en la rama de un árbol. **b)** Abeja descansando en el pétalo de una flor. **c)** Oso perezoso colgado de la rama de un árbol **d)** Jirafa flexionando su cuello sobre su espalda para descansar (**Brown, 2018; Helfrich-Förster, 2018**).

Para este trabajo, el modelo biológico que se estudió fue el acocil, un crustáceo acuático decápodo que muestra conductas de interés para la neurobiología, como la agresividad (**Jiménez-Morales et al., 2018**), la exploración y el sueño (**Ramón et al., 2004**), siendo esta última a la que me enfoqué en este estudio.

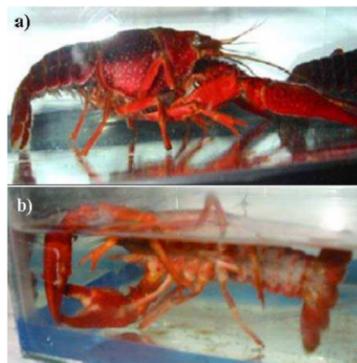


Figura 1.2: Posturas de un acocil en su acuario **a)** parado y **b)** de lado (**Mendoza-Ángeles et al., 2005**).

Cuando el acocil duerme, la postura estereotípica implica colocarse de lado con un tren de patas ambulatorias apoyado sobre el fondo del acuario y el otro sobre una de las paredes, como se muestra en la Figura 1.2b. Esta conducta aparece en cualquier hora del día, pero se ha reportado que duermen más de noche (**Ramón et al., 2004**). Esto implica que, para estudiar la conducta de los acociles, es necesario obtener registros de video que abarquen extensos periodos.

Para el estudio de conducta en el acocil, usualmente un investigador analiza los videos segmento a segmento y anota la conducta que se presenta y el momento en que ocurre. En el caso del sueño, se realizan hipnogramas; diagramas que permiten representar las fases de sueño y vigilia a través del tiempo. Para un acocil, las conductas que se reportan son:

- **Exploración:** El acocil está parado o deambulando en su acuario.
- **De lado:** El acocil adopta la postura estereotípica de sueño.

Al ser graficadas en un hipnograma, el resultado es el de la Figura 1.3:

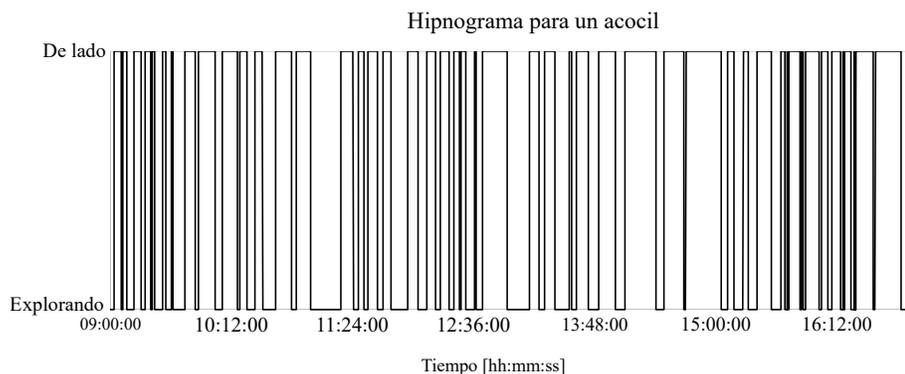


Figura 1.3: Hipnograma de la conducta de un acocil obtenido de un registro de 8 horas. El valor 0 representa **Exploración** y 1 representa **Posición de lado** (modificado de **Osorio-Palacios et al., 2021**).

## 1.1. Planteamiento del problema

Estudiar la conducta de un acocil puede ser fatigante y requiere de mucha paciencia pues hay ocasiones en las que un investigador debe reportar la conducta de 2 o más acociles para un mismo experimento y estos pueden tener duraciones incluso mayores de 24 horas.

Otra vicisitud ocurre cuando dos o más observadores estudian el mismo registro de conducta, lo que provoca inconsistencias entre ellos, dificultando el análisis, además de la fatiga y el estrés causados por revisar largos registros. Es por eso que los investigadores recurren al procesamiento digital de imágenes, para evitar decisiones erróneas durante el análisis de los registros provocadas por el cansancio.

Actualmente los algoritmos de procesamiento digital de imágenes son implementados como un apoyo para acelerar tareas estresantes que necesiten de una persona presente, que analice y reporte durante tiempos prolongados, como lo son los estudios de conducta.

Recientemente se han desarrollado sistemas para el estudio de la conducta que no requieren que una persona supervise. Dichos sistemas se basan en el Deep Learning, como DeepLabCut (**Nath et al., 2019**) o SLEAP (**Pereira et al., 2022**); estos programas requieren ser entrenados para identificar la postura que se quiere analizar.

Otros sistemas como LoliTrack® (**Loligo Systems, 2021**) o ANY-maze (**Corral, 2020**) se utilizan para estudiar la trayectoria y los momentos de inmovilidad que un animal de estudio puede presentar. La desventaja que presentan es que sólo estudian el centroide del animal, dejando de lado la postura, esta situación obliga a los investigadores a revisar el registro completo.

Si bien estos sistemas son buenas opciones para el estudio de conducta de animales en laboratorios, no solucionan el problema que plantea este trabajo: detectar cuando un crustáceo acuático cambia su postura durante registros de conducta sin que un observador esté reportando.

Implementar un sistema que permita analizar la conducta de acociles de manera recurrente, sin que se necesite entrenar el programa y que permita obtener un hipnograma para cada registro realizado, puede ser valioso para el estudio del sueño en acociles pues permitirá obtener resultados confiables y reproducibles pues evitará errores en las anotaciones que dos personas puedan describir del estudio de conducta.

Con ayuda del procesamiento digital de imágenes se desarrolló un programa que permite analizar la conducta de los acociles sin que haya una persona presente supervisando.

Las tareas que se optimizaron fueron el seguimiento de los acociles explo-

rando su acuario, y la obtención del hipnograma, (Figura 1.3) con el fin de que ya no sea necesario realizar anotaciones de las transiciones de la postura del acocil, facilitando al observador el análisis de las áreas que recorrió y los cambios de postura que tuvo durante el registro.

## 1.2. Justificación

Una de las principales dificultades de analizar imágenes de crustáceos acuáticos es que la imagen se deforma por el movimiento del agua (**Liang et al., 2024; Patullo et al., 2007**), lo que impide que se detecten partes de su cuerpo que son de interés para analizar su movimiento.

Programas como DeepLabCut o ANY-maze han sido usados con mayor enfoque en mamíferos terrestres; pocos estudios se han realizado con animales acuáticos, pues la distorsión producida por el agua resulta en detecciones erróneas.

Para contrarrestar el problema de la distorsión del agua, se optó por la identificación de líneas y círculos pues no sufren deformación en imágenes ruidosas como las que son tomadas debajo del agua (**Duda y Hart, 1972; Nixon y Aguado, 2020**). El programa está basado en la Transformada de Hough, un algoritmo que permite la detección de círculos en imágenes.

## 1.3. Transformada de Hough

La transformada de **Hough, 1962**, es un algoritmo que permite el reconocimiento de patrones complejos; se aplica en procesamiento digital de imágenes y permite detectar líneas, círculos o elipses de manera precisa en imágenes binarias a las que se les aplicó un algoritmo de detección de borde (**Cherabit et al., 2012; Duda y Hart, 1972; Nixon y Aguado, 2020**).

En una imagen binaria, los píxeles se representan como un sistema coordenado  $(x, y)$ ; cuando varios píxeles forman una línea, se pueden representar con la ecuación 1.1:

$$y = mx + b \tag{1.1}$$

donde  $m$  es la pendiente de la línea, y  $b$  su ordenada al origen.

En coordenadas polares, cuando se representa una recta como un radio  $\rho$  con un ángulo  $\theta$ , usualmente se realiza con la ecuación 1.2:

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (1.2)$$

donde  $\rho$  representa el radio que va desde el origen (0,0) (o el primer píxel de la imagen), hacia el píxel  $(x,y)$  de interés, como se muestra en la Figura 1.4a.

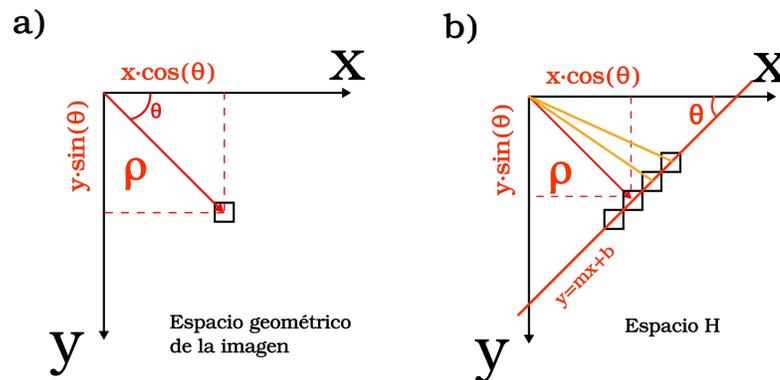


Figura 1.4: Ángulo que se forma para **a)** Un punto  $(x_i, y_i)$  y su parametrización en el espacio  $(\rho, \theta)$  y **b)** Una recta de píxeles  $y = mx + b$  que coinciden con el ángulo  $\theta$

La ecuación 1.2 tiene un periodo de  $2\pi$  de manera que, al representarla gráficamente, se producen curvas sinusoidales.

Al observar las curvas producidas por los parámetros  $(\rho, \theta)$ , se producen sinusoidales; los cruces donde las curvas intersectan son los ángulos que relacionan dos o más píxeles a una misma recta (Figura 1.4b)(Gonzales y Woods, 2018; Nixon y Aguado, 2020).

## 1.4. Transformada Hough para detectar círculos

La ecuación característica para representar un círculo es la siguiente:

$$(x - a)^2 + (y - b)^2 = r^2 \quad (1.3)$$

donde los valores  $(a, b)$  son el centro de la circunferencia, y  $r$  es el radio. Al convertir la ecuación 1.3 a coordenadas polares tendremos que

$$\begin{aligned} x &= a + r \cdot \cos(\theta) \\ y &= b + r \cdot \sin(\theta) \end{aligned} \quad (1.4)$$

El conjunto de ecuaciones 1.4 tienen el objetivo de encontrar la tripleta de valores  $(a, b, r)$  por lo que se presentarán dos casos:

### 1.4.1. Radio $r$ conocido

La búsqueda de parámetros se reduce a encontrar valores  $(a, b)$  que son el centro del círculo; el perímetro se delimita por una serie de valores  $(x_i, y_i)$ ; al transformar los valores con la ecuación 1.4, se obtienen circunferencias en el espacio H (**Rhody, 2005**).

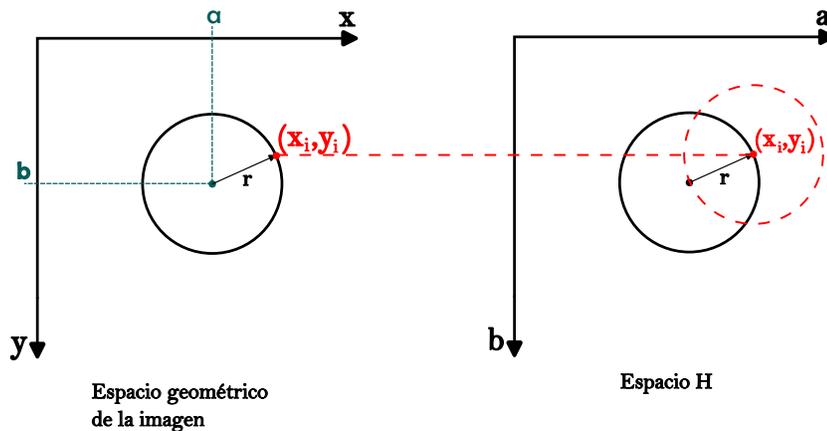


Figura 1.5: Representación de un punto  $(x_i, y_i)$  que pertenece a una circunferencia con centro en  $(a, b)$ ; en la derecha se muestra su transformación del punto  $(x_i, y_i)$  en el espacio H como centro de una circunferencia que intersecta en el punto  $(a, b)$ .

Al seguir transformando puntos  $(x_i, y_i)$ , se producen más circunferencias que intersectan en el punto  $(a, b)$ ; las coordenadas con mayor número de coincidencias son consideradas como el centro del círculo de interés.

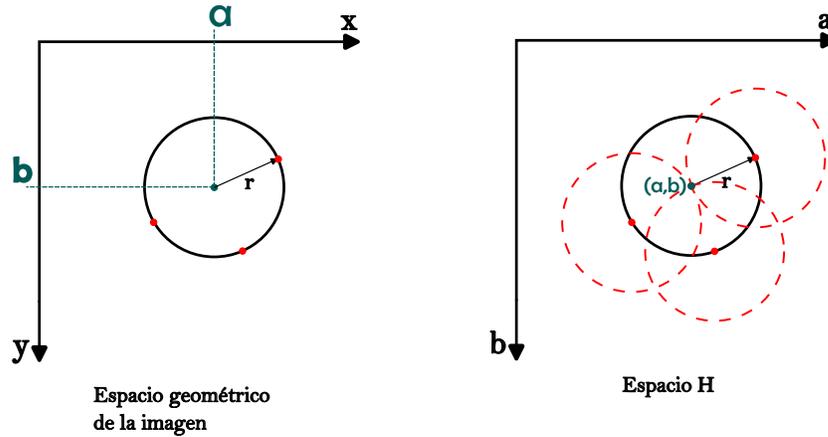


Figura 1.6: Transformada de Hough para varios puntos  $(x_i, y_i)$ ; en la imagen de la derecha se producen varias circunferencias que intersectan en el punto  $(a, b)$ , lo que indica que los puntos  $(x_i, y_i)$  pertenecen a la circunferencia de interés.

En la Figura 1.6 se muestran las circunferencias que intersectan en el punto  $(a, b)$  que será el centro del círculo de interés. Cuando en el acumulador de intersecciones en  $(a, b)$  se encuentren varias coincidencias, en el plano H se mostrarán varias circunferencias.

La transformada de Hough funciona también para encontrar varias circunferencias en una misma imagen e incluso circunferencias incompletas.

### 1.4.2. Radio $r$ desconocido

En este caso, se tendrá un vector de coincidencias de 3 dimensiones  $(a_0, b_0, r_0)$ ; la transformada de valores  $(x_i, y_i)$  formarán circunferencias que coinciden con el centro  $(a, b)$ ; la búsqueda del radio  $r_0$  dará la tercera dimensión; el valor se irá incrementando de 0 hasta el valor de  $r$  esperado.

La búsqueda de estos tres parámetros dará, en el espacio tridimensional, un cono con su pico en  $(a, b)$  y su altura será dada por el radio  $r$ . Cuando la búsqueda arroja un centro  $(a, b)$  y un radio  $r$ , la matriz de coincidencias se incrementa, indicando que se ha detectado un círculo con las dimensiones esperadas.

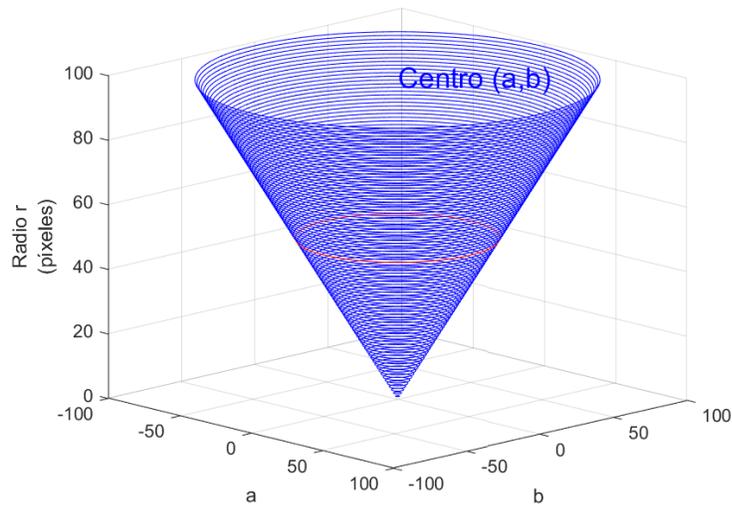


Figura 1.7: Transformada de Hough para buscar  $(a_0, b_0, r_0)$  (Rhody, 2005).

## 1.5. Búsqueda de círculos bajo el agua

Al analizar imágenes de objetos sumergidos en agua, la vista aérea suele presentar una distorsión en su forma, debido a los movimientos del agua o por la distorsión del lente (Liang et al., 2024; Patullo et al., 2007).

Para el trabajo presentado, se optó por la búsqueda de círculos con la Transformada de Hough debido a que la forma circular sufre una deformación mínima en imágenes distorsionadas (Duda y Hart, 1972) como las que son tomadas debajo del agua.

La aplicación de la Transformada de Hough en una imagen de interés se realizó por medio de la función pre-programada de MatLab (MathWorks, 2024):

```
imfindcircles(im, [rmin rmax])
```

donde **im** es una imagen binaria que se obtuvo a través de un procesamiento de imágenes, y **[rmin rmax]** son el rango, en píxeles, de los radios que eran de interés.

## 1.6. Histograma de una imagen

La representación de imágenes suele hacerse por medio de un histograma que nos permite representar una distribución de los niveles de intensidad (Escalante, 2006; Gonzales y Woods, 2018):

$$P(r) = \frac{N(r)}{M \times N} \quad (1.5)$$

donde  $P(r)$  representa la probabilidad de ocurrencia de un nivel de intensidad (0 a 1);  $N(r)$  representa el número de píxeles que tienen un nivel de intensidad en común, y  $M \times N$  son las dimensiones de la imagen. Además, la ecuación 1.5 está normalizada a 1.

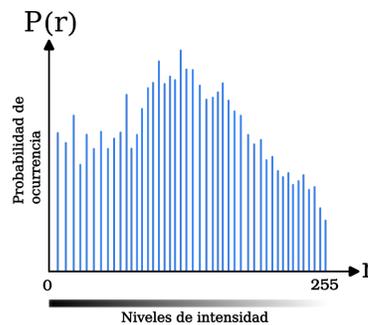


Figura 1.8: Representación de un histograma para una imagen donde  $r$  representa los posibles niveles de intensidad (0 a 255) y  $P(r)$  representa la probabilidad de que se produzca un nivel de intensidad (0 a 1).

En la Figura 1.8 se tiene una representación general para una imagen, en forma de histograma; la variable  $r$  (niveles de intensidad) está representado en una escala de 0 a 255, para una imagen en escala de grises, la barra de color está desde 0, siendo los niveles de intensidad oscuros, hasta 255, que representa los colores blancos.

Las imágenes que permiten a los observadores percibir la mayor información posible suelen tener histogramas con una distribución de niveles de intensidad como la de la Figura 1.8, pero se pueden producir condiciones externas que provoquen imágenes no tan agradables a la vista, lo que se resume en 4 principales casos:

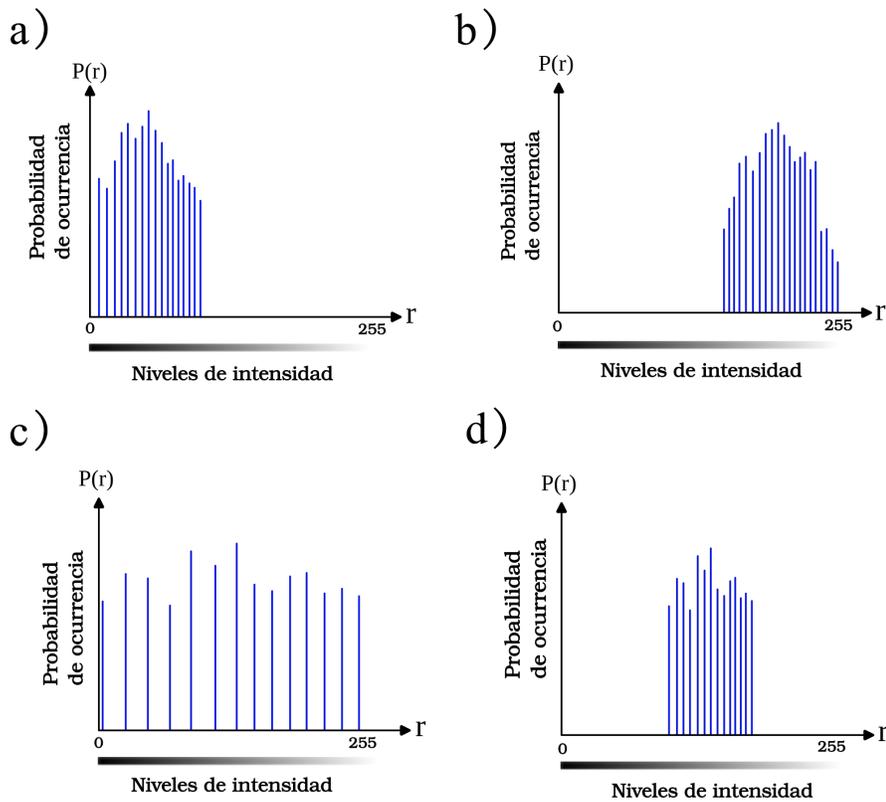


Figura 1.9: Histogramas que representan cómo se distribuyen los niveles de intensidad en cuatro diferentes imágenes: **a)** oscura. **b)** brillante. **c)** alto contraste. **d)** bajo contraste.

El histograma de la Figura 1.9a representa una imagen con niveles de intensidad bajos, lo que para un observador es una imagen **oscura**. El histograma de la Figura 1.9b representa una imagen con altos niveles de intensidad, lo que para un observador es una imagen **brillante**.

Otro de los problemas que puede presentar una imagen es cuando se tiene un bajo contraste, como se muestra en la Figura 1.9d; se puede observar que los niveles de intensidad están comprimidos en la parte central del histograma. Las imágenes que son más agradables a la vista, o que permiten reconocer mayor información de ellas, son las que presentan un alto contraste, como se observa en la Figura 1.9c, debido a que los niveles de intensidad están distribuidos en toda la escala del mapa de colores.

La representación ideal de una imagen es cuando un histograma tiene una función de densidad  $P(r)$  constante, es decir, todos los niveles de intensidad contienen el mismo número de píxeles.

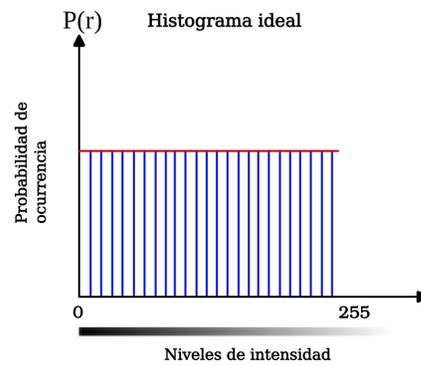


Figura 1.10: Histograma ideal

La Figura 1.10 representa una función de densidad constante para todos los niveles de intensidad. Cuando se trabaja con imágenes reales, es complicado obtener histogramas con una función de densidad igual o casi igual, debido a las condiciones externas con las que se tomó la imagen.

a)



b)

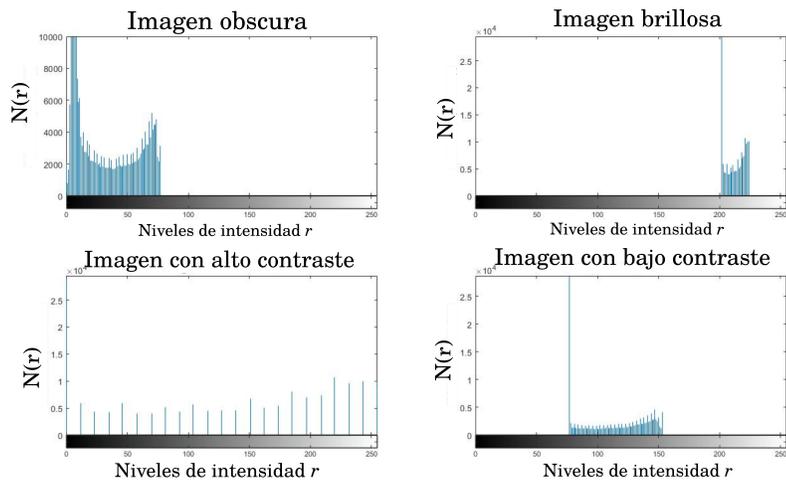


Figura 1.11: Imágenes tomadas en contextos reales, con diferentes contrastes y brillos, para la película Macario (Gavaldón, 1959) (a) Y su representación a través de su histograma. (b). Cada histograma representa una imagen oscura, brillante, de alto contraste y de bajo contraste, respectivamente.

La Figura 1.11 es un ejemplo de lo que sucede con imágenes tomadas en contextos reales. Los niveles de intensidad pueden variar por las condiciones del entorno en la que se tomó la fotografía.

En la Figura 1.11a se muestra una representación, en imágenes, de los histogramas de las Figuras 1.11b. Se observa que la distribución de niveles de intensidad es importante para que un observador perciba la mayor cantidad de información posible.

Para evitar estos casos y presentar una imagen nítida, esperando acercarnos a un histograma ideal, como se muestra en la Figura 1.10, se realizan operaciones lineales que permitan el mejoramiento de una imagen.

Las operaciones más utilizadas son de realce puntual como la **expansión del histograma** y la **ecualización del histograma**.

### 1.6.1. Expansión del histograma

La expansión de los niveles de intensidad consiste en manipular el histograma original y “estirarlo” hacia ambas direcciones de la barra de niveles de intensidad y tratar de hacer que la distribución sea uniforme para todos los niveles (Gonzales y Woods, 2018).

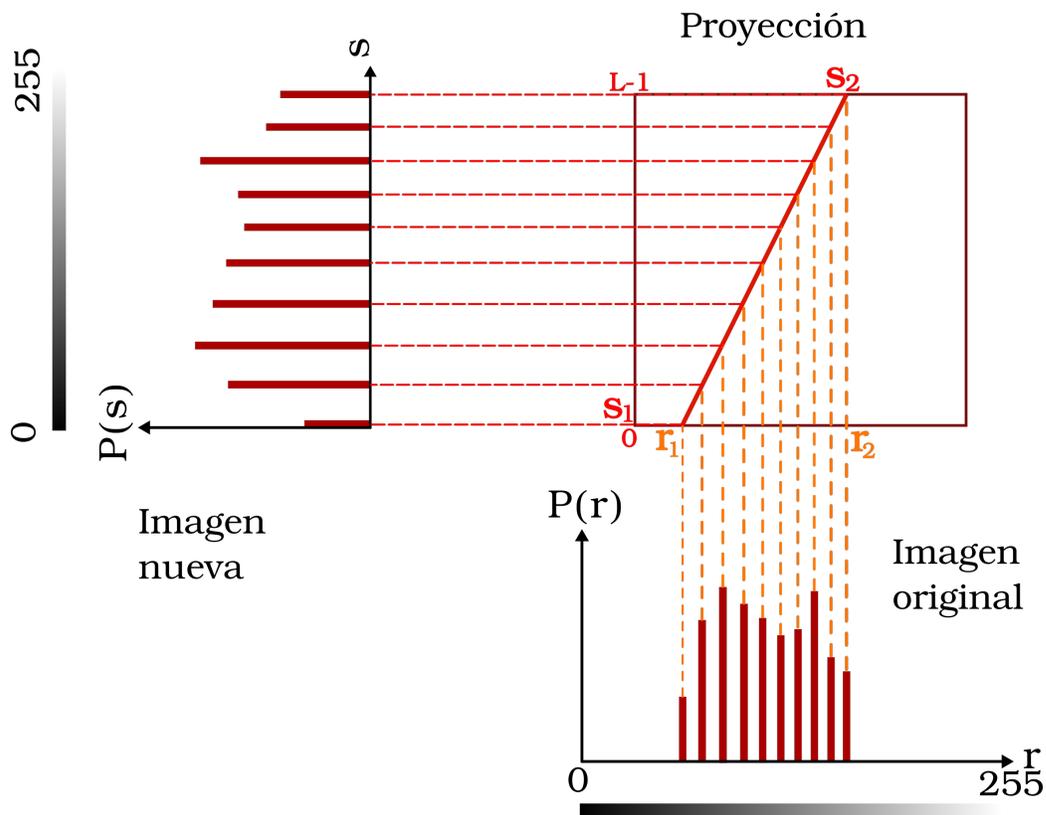


Figura 1.12: Expansión de histograma

La expresión algebraica que representa la transformación lineal de la Figura 1.12 será a partir de la ecuación 1.6:

$$\frac{s_2 - s_1}{r_2 - r_1} = \frac{S - s_1}{R - r_1} \quad (1.6)$$

La ecuación 1.6 representa una recta a partir de dos puntos, pero adecuada para una imagen; la variable  $S$  representa la nueva imagen que se obtiene de ex-

pandir el histograma; la variable  $R$  representa la imagen original que será evaluada para la obtención de la imagen  $S$ ; al despejar la imagen  $S$  tenemos que:

$$\frac{s_2 - s_1}{r_2 - r_1} (R - r_1) + s_1 = S \quad (1.7)$$

El término  $\frac{s_2 - s_1}{r_2 - r_1}$  representa la pendiente de la recta que se utiliza para la expansión del histograma; el valor  $s_2 - s_1$  representa los límites de la discretización de los niveles de color (generalmente en escala de grises). El nivel máximo es  $s_2 = L - 1 = 255$ ; y  $s_1 = 0$ , donde  $L$  representa el número de posibles niveles de intensidad.

Los valores  $r_2$  y  $r_1$  representan el nivel de intensidad máximo y mínimo, respectivamente, de la imagen original, estos se obtienen del histograma de la imagen  $R$ , como se observa en la Figura 1.12.

Retomando la ecuación 1.7 y evaluando los valores  $s_2$  y  $s_1$  tendremos que la imagen con el histograma expandido será:

$$S = \frac{255}{r_2 - r_1} (R - r_1) \quad (1.8)$$

### 1.6.2. Ecuación del histograma

El histograma, definido por la ecuación 1.5, que representa los niveles de intensidad contenidos en una imagen, se puede interpretar como una señal discreta. Para localizar el nivel de intensidad que representa una probabilidad de ocurrencia se utiliza la siguiente transformación:

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k p_r(r_j); k = 0, 1, 2, \dots, L - 1 \quad (1.9)$$

La ecuación 1.9 realiza la localización de las nuevas probabilidades para los niveles de intensidad que representarán la imagen de salida. Este procedimiento permite agrupar niveles de intensidad con poca probabilidad de ocurrencia con altas probabilidades de ocurrencia; además de realzar las características de una imagen para así, mejorar su contraste (**Escalante, 2006; Gonzales y Woods, 2018**).

La ecualización de un histograma ideal (Figura 1.10) busca que todos los valores de intensidad representen una muestra  $s_k$  constante, pero en la realidad es muy difícil conseguir ese efecto.

La forma de corregir este proceso es reagrupar el histograma obtenido en la ecuación 1.9, con una igualación de histogramas; es decir, usar un histograma de referencia para representar la nueva imagen (**Gonzales y Woods, 2018**).

### 1.6.3. Igualación de histogramas

Debido a que el procedimiento de ecualizar el histograma no siempre mejora el contraste de la imagen, se procede a realizar una igualación de histogramas, este proceso busca una nueva agrupación de niveles de intensidad para mejorar el contraste de una imagen a partir de una imagen referencia con un contraste alto (**Gonzales y Woods, 2018**).

Para obtener una igualación de histogramas se realiza la siguiente operación:

$$G(z_q) = (L-1) \sum_{i=0}^q p_z(z_i); \quad q = 0, 1, 2, \dots, L-1 \quad (1.10)$$

donde  $p_z(z_i)$  es el valor  $i$ -ésimo de nivel de intensidad del histograma de referencia.

La ecuación requiere un valor de  $q$  tal que la probabilidad de ocurrencia  $s_k$  mas alta del histograma original sea igual a la de la imagen de referencia:

$$G(z_q) = s_k \quad (1.11)$$

De manera homóloga, se necesita un valor de  $k$  que permita encontrar el valor de probabilidad de ocurrencia más alta de la imagen de original, que sea igual a la probabilidad más alta de la imagen de referencia:

$$z_q = G^{-1}(s_k) \quad (1.12)$$

En la práctica, no es necesario encontrar la función inversa de  $G$  ya que, al trabajar con valores discretos, sólo es necesario trabajar con valores enteros finitos.

La igualación de histogramas se resume en los siguientes pasos:

- Se evalúan las probabilidades de ocurrencia de la imagen de referencia con la ecuación 1.10.
- Los resultados se redondean a valores enteros en el rango  $[0, L - 1]$ .
- Los valores de  $s_k$  obtenidos de la ecuación 1.9 se igualan al valor más cercano de  $z_q$  obtenido en el paso anterior.

De esta manera, las probabilidades de ocurrencia de la imagen original toman una nueva agrupación, lo que resulta en un histograma que tiene niveles de intensidad parecidos al histograma de la imagen referencia; los valores máximos de  $s_k$  serán los mismos que  $z_q$ , y los niveles de intensidad de la imagen original se expandirán, de manera que el nivel más alto será el mismo que el de la imagen de referencia.

Para el desarrollo de este trabajo, al ser una gran cantidad de imágenes las que se analizaron, fue necesario realizar la igualación de histogramas ya que las condiciones de cada cuadro que entraron en el programa tenían características y distribuciones de niveles de intensidad distintas, por lo que fue necesario hacer una semejanza entre imágenes para poder obtener la detección esperada.

Los procedimientos mostrados se aplicaron a las imágenes que se obtuvieron durante los registros de conducta.

El programa presentado se implementó con base en el realce puntual de imágenes y en la detección de círculos, lo que resultó en altos niveles de confianza.

# Capítulo 2

## Objetivos

---

### 2.1. Objetivo general

Diseñar e implementar un sistema de monitoreo continuo, basado en el procesamiento digital de imágenes, que permita automatizar el análisis conductual de crustáceos acuáticos.

### 2.2. Objetivos particulares

- Automatizar la detección del cambio de postura que presentan los acociles.
- Reconstruir un hipnograma que represente los momentos de todo un registro en los que el acocil está de lado.
- Implementar el sistema de detección en MatLab.



# Capítulo 3

## Materiales y métodos

---

El modelo biológico empleado en los experimentos fue el acocil *Procamburus clarkii* (3 machos y 2 hembras). Cada acocil se colocó en un acuario individual y se tomaron registros de conducta en los que se cambió la iluminación del cuarto con 12 horas de luz blanca y 12 horas de luz roja.

### 3.1. Marcas de pintura

Debido a la distorsión que se produce por el agua (**Patullo et al., 2007**), se decidió pintar círculos en los cuerpos de los acociles para estudiar su conducta debido a que, esta forma geométrica presenta distorsión mínima en imágenes ruidosas (**Duda y Hart, 1972**), permitiendo así la detección de trayectoria y su cambio de postura.

En la Figura 3.1a se muestran las marcas que fueron detectadas cuando el acocil está parado: tres puntos en su cefalotórax y un punto en cada quela (consultar Anexo B para ver la anatomía externa del acocil). La pintura que se utilizó es de color blanco o de un tono rosa muy tenue; estos colores permiten que se detecten los círculos tanto con luz blanca, como con luz roja.

La Figura 3.1b se muestra el punto de color verde que se pintó en los costados del acocil. El fin de poner un punto de pintura verde es que la marca sólo apareció cuando el acocil estaba de lado, permitiendo así detectar el cambio de postura.

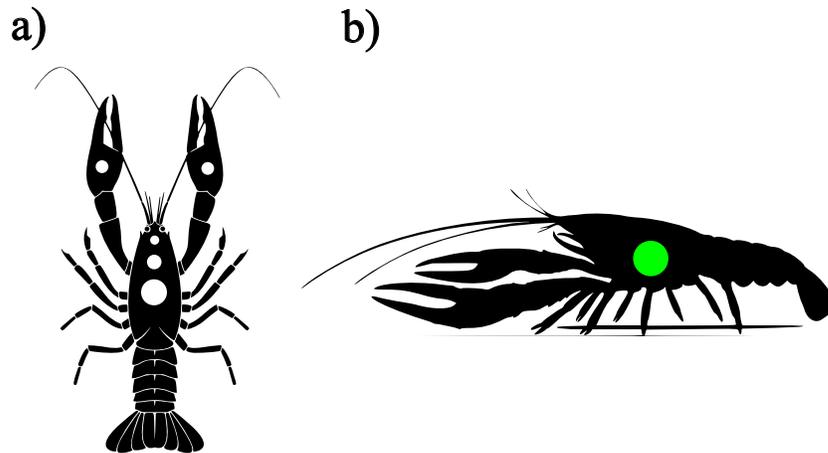


Figura 3.1: Marcas que fueron pintadas en los acociles. a) Puntos blancos en el cefalotórax dorsal para la detección de trayectoria. b) Punto verde en el cefalotórax lateral derecho e izquierdo para la detección de la postura de lado.

### 3.2. Obtención de registros

Después de pintar los círculos en los acociles, se colocaron en acuarios individuales de  $28 \times 22 \times 15$  [cm]. La conducta se registró con una cámara SONY HDR-CX405 para después procesar el video con el algoritmo propuesto.

Los registros tuvieron una duración de 24 horas, alternando la luz con la que se iluminaron; es decir, 12 horas con luz blanca (registros en el día) y 12 horas con luz roja (registros nocturnos). En cada registro se analizaron de 2 a 3 acociles, como máximo, esto permitió obtener los mejores resultados en la detección de los círculos marcados en su cuerpo.

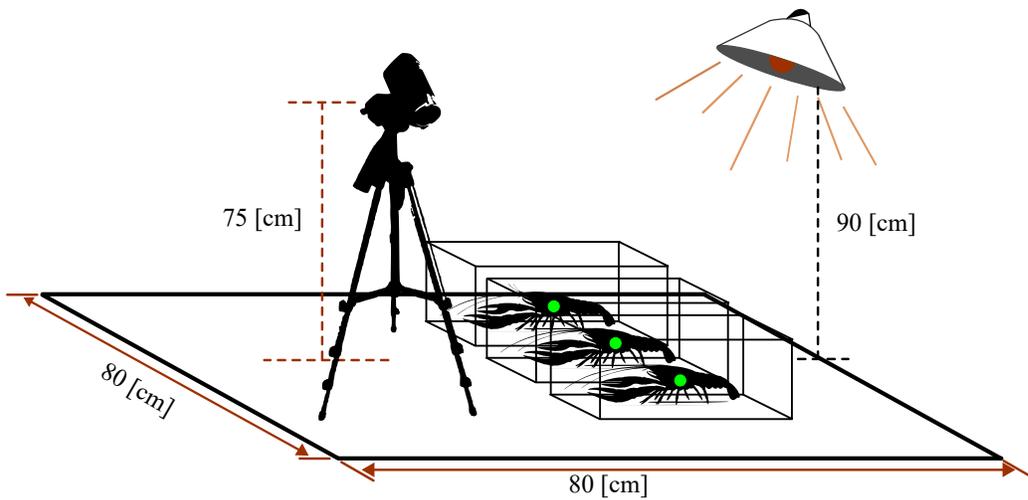


Figura 3.2: Colocación de acociles para registrar su conducta.

En la Figura 3.2 se muestra un esquema de cómo se registró la conducta; la cámara se colocó a una altura de 75 [cm] desde la mesa hasta el soporte. La lámpara fue colocada 90 [cm] arriba de la mesa.

La conducta que se esperó detectar es el cambio de postura de los acociles en la que se colocan apoyados en las paredes del acuario, como se muestra en la Figura 3.3.

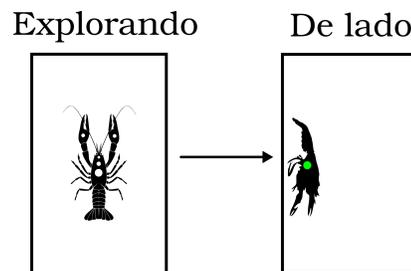


Figura 3.3: Cambio de postura.

Al finalizar el análisis de conducta de los acociles, se graficaron todas las coordenadas del estado **Explorando** y en la misma gráfica se dibujaron las coordenadas del punto verde para ver las zonas donde el acocil cambió su postura.



# Capítulo 4

## Desarrollo del problema

---

En este capítulo se explicará el procesamiento de imágenes que se realizó para el estudio de conducta, como un proceso general.

Se muestra también cómo se manejó el cambio de luz en las imágenes para poder manipularlas y que el proceso general no distinga, en su mayoría, si es luz roja o blanca, así como las gráficas que se pueden obtener de un registro de conducta.

### **4.1. Segmentación de videos para análisis individual y colectivo de imágenes**

Para que los resultados del procesamiento fueran normalizados, se presentó una serie de pasos que tratan de mejorar las imágenes con luz blanca y roja, esto permitió que se entregara un sólo código que abarque los dos tipos de luz pero que no altere la obtención de información una de otra.

Al obtener las imágenes del video de conducta, se realizó una muestra cada 50 o 25 cuadros (dependiendo de la configuración de la cámara); en tiempo real, es equivalente a tomar una muestra cada segundo de video.

El procesamiento comenzó con la iluminación de la imagen; al tener dos escenarios de registro, uno con luz blanca y otro con luz roja, se perdió información al cambiar de iluminación por lo que fue necesario contemplar un mejoramiento de contraste a través del histograma.

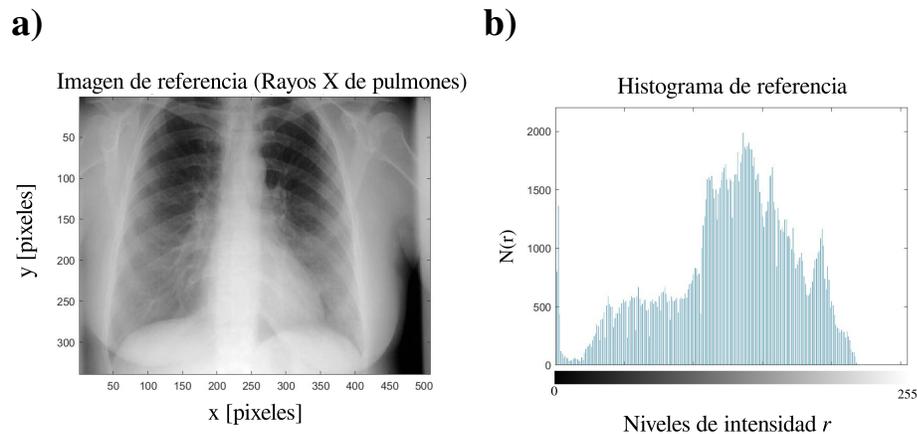


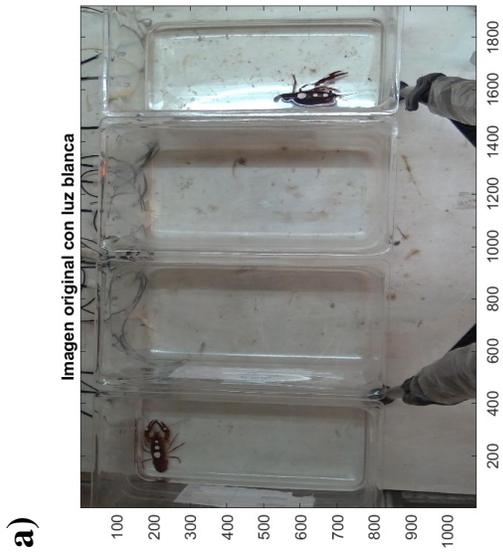
Figura 4.1: **a)** Imagen referencia y **b)** su histograma para realizar la ecualización.

La Figura 4.1 tiene como objetivo ser el histograma referencia para los dos escenarios, luz blanca y luz roja. Los resultados de ecualizar las imágenes se presentan en las Figuras 4.2 y 4.3.

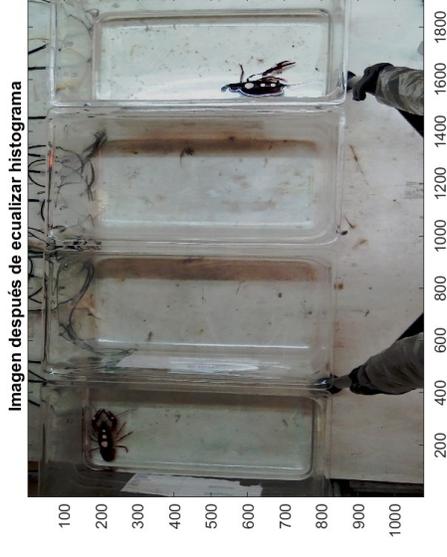
Con luz blanca, los acociles tienen sus marcas visibles, la ecualización del histograma presentó una nueva distribución de niveles de intensidad, lo que disminuyó los niveles de intensidad cercanos a 255, como se muestra en la Figura 4.2d. Al comparar la Figura 4.2a con 4.2c, se observa una disminución en las áreas brillosas que rodean a los acociles.

Por otro lado, en la captura de video con luz roja, se ve un mejoramiento del brillo de la Figura 4.3a a la Figura 4.3c, lo que permitió determinar los puntos de interés del acocil.

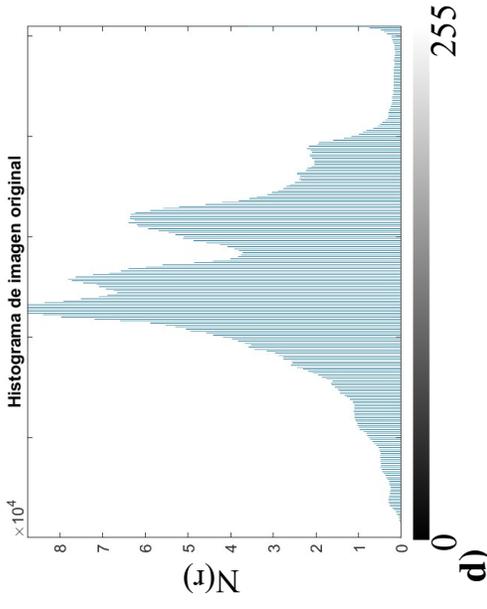
La Figura 4.3 refleja el efecto de ecualizar el histograma con luz roja; los niveles de intensidad de la Figura 4.3d ahora presentan una distribución similar a la de la imagen de referencia (Figura 4.1b) y no sólo en la parte izquierda, como en la Figura 4.3b, lo que permitió detectar los círculos de interés bajo luz roja.



**c)**



**b)**



**d)**

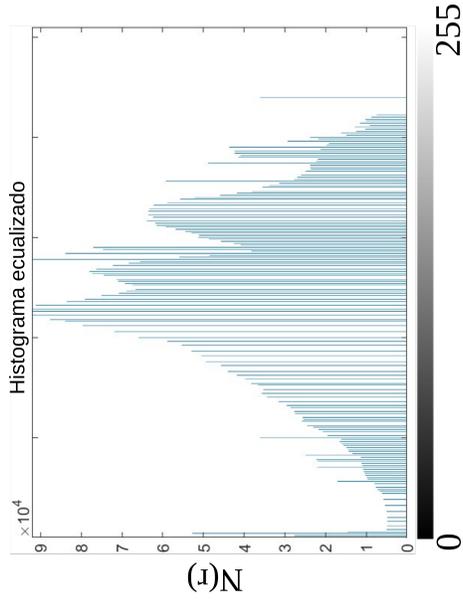


Figura 4.2: Proceso de ecualización para muestras de video con luz blanca. **a)** Imagen original (luz blanca). **b)** Histograma de la Imagen 4.2a. **c)** Imagen después de proceso de ecualización. **d)** Histograma de la Figura 4.2c.

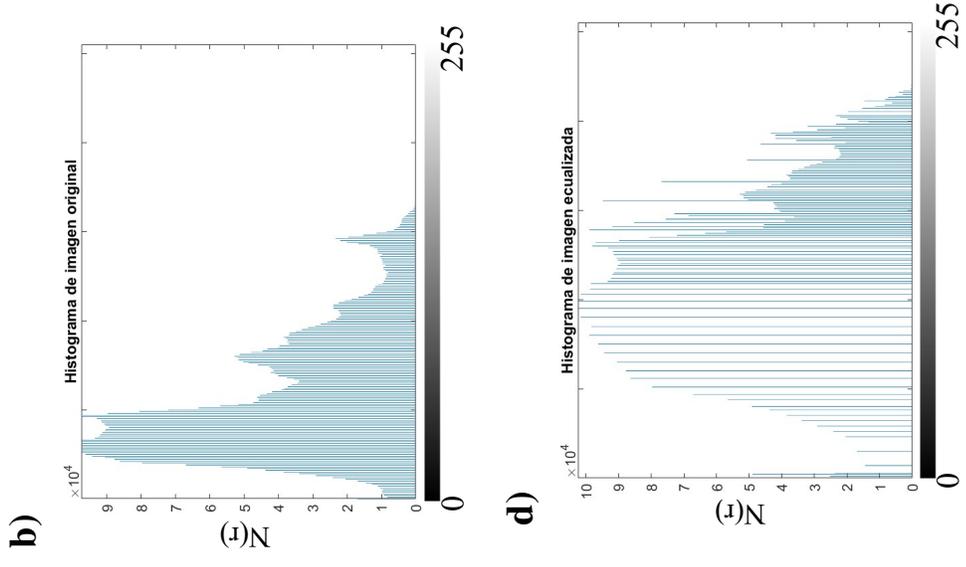
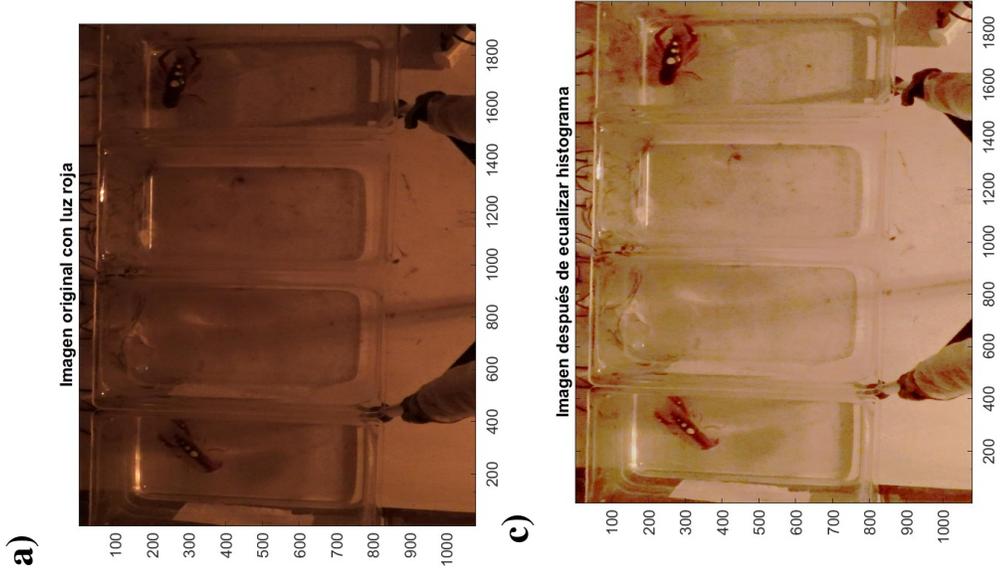


Figura 4.3: Proceso de ecualización de histograma para muestras de imagen con luz roja. **a)** Muestra de video con luz roja. **b)** Histograma de la Figura 4.3a. **c)** Imagen 4.3a ecualizada. **d)** Histograma ecualizado de la Figura 4.3b.

### 4.1.1. Proceso general

Una vez ecualizadas las imágenes, se aplicaron una serie de procesos para detectar los puntos de interés. El primer paso fue convertir la imagen de RGB a una escala de grises.

$$imGray = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B \quad (4.1)$$

La ecuación 4.1 contempla los coeficientes que se necesitan multiplicar para obtener una sola imagen en escalas de grises a partir de las tres bandas RGB (Gonzales y Woods, 2018).

a)



b)



Figura 4.4: Conversión de imágenes de RGB a escala de grises. a) Figura 4.2c en escala de grises. b) Figura 4.3c en escala de grises.

A partir de la Figura 4.4, se obtuvo una imagen binaria que tenga valores de intensidad de 0 y 1, siendo 0 el negro y 1 el blanco, con la ecuación 4.2:

$$binaria = imGray < umbral \quad (4.2)$$

La ecuación 4.2 implica una operación lógica que comparó los valores de intensidad de la imagen; si los valores de intensidad son menores al umbral, se les asignó un 1; el caso contrario es que, si son mayores al umbral, se les asignó un 0.

Los valores del umbral se obtuvieron de manera empírica, de manera que los círculos de interés mantuvieran su forma geométrica durante el proceso (Tabla 6.1). Al completar la operación se obtuvo una imagen binaria como la de la Figura 4.5

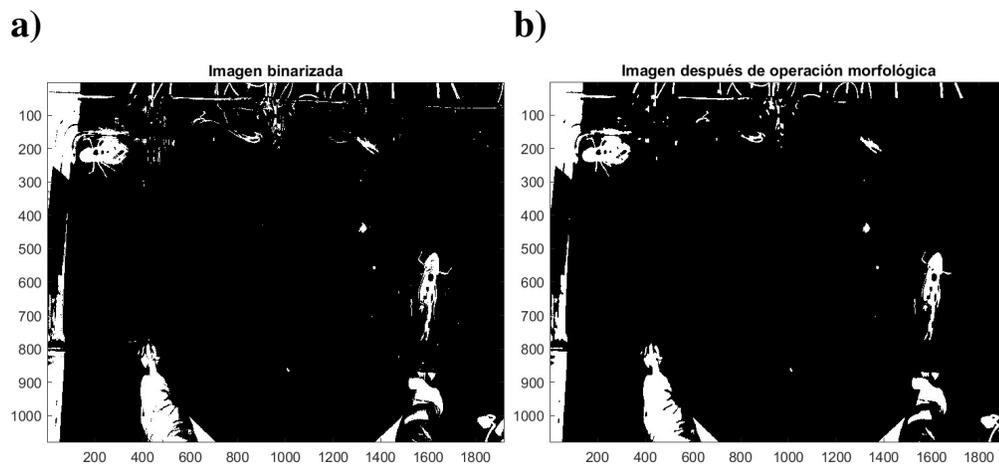


Figura 4.5: Binarización de imagen y operación morfológica. a) Imagen 4.4a binarizada y b) Imagen 4.5a después de operación morfológica

En la Figura [4.5a] se muestra la Figura 4.4a después de evaluarla en la ecuación 4.2, y en la Figura 4.5b se muestra la Figura 4.5a con el llenado de huecos con una operación morfológica; el llenado de huecos consiste en “rellenar” pixeles aislados con los valores predominantes de sus vecinos.

En MatLab se llamó a la función `bwmorph(im, operación, n)` donde **im** es la imagen binaria obtenida de la Ecuación 4.2 ; **operación** es el tipo de algoritmo de decisión para el llenado de 0; en este caso se utilizó la operación ‘majority’ y **n** es el número de veces que se repitió el algoritmo; usar un valor muy alto puede hacer que la imagen ya no cambie pero consuma tiempo.

La operación “**majority**” analiza el píxel central de una vecindad de  $3 \times 3$  para determinar si el valor se debe rellenar o no (MathWorks, 2024); el valor del píxel central será 1 si hay cinco o más píxeles que lo rodean con valor igual a 1; de lo contrario el píxel pasará a 0.

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & \mathbf{0} & 0 \\ 0 & 1 & 1 \end{bmatrix} \implies \begin{bmatrix} 1 & 1 & 1 \\ 1 & \mathbf{1} & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

La matriz anterior es un ejemplo de cómo se aplica la operación “**majority**” en una vecindad de  $3 \times 3$  píxeles. Esta operación permitió preservar la forma circular de las marcas en los acociles para así obtener su detección en un proceso posterior.

#### 4.1.2. Negativo de la imagen

La detección de círculos que hace MatLab suele hacerla con el fondo blanco y un frente negro, también círculos en negro con frente blanco; es por eso que se debe aplicar el negativo de la imagen que encerrará la circunferencia en un entorno de píxeles con valor 0.

$$\text{negativo} = 1 - \text{binaria} \quad (4.3)$$

En la ecuación 4.3 se invierten los valores binarios de la Figura 4.5b para que así, se pueda proceder a encontrar los círculos de interés.

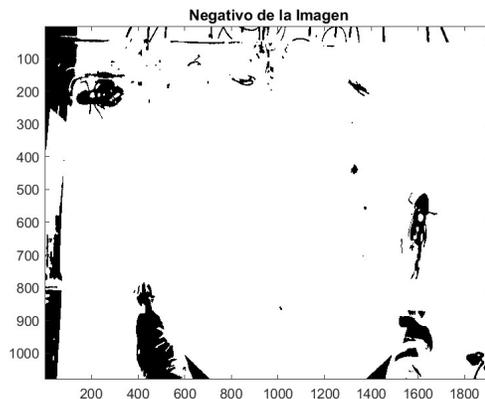


Figura 4.6: Negativo de la Figura 4.5b

En la Figura 4.6 se muestra el negativo de la Figura 4.5b donde ahora los acociles tienen un color negro y su fondo es blanco. Para que el algoritmo sólo analice áreas especificadas, se recortó el área de sondeo para que sólo tome las regiones delimitadas y no toda la imagen.

### 4.1.3. Selección de áreas de estudio

Las dimensiones de cada acuario son seleccionadas por el usuario, de manera que sólo se ingresó el número de acociles que se analizaron para después, seleccionar el área total donde se encuentran los acociles que se estudian, los puntos se deben seleccionar de manera que el área sea la máxima para cada detección individual.

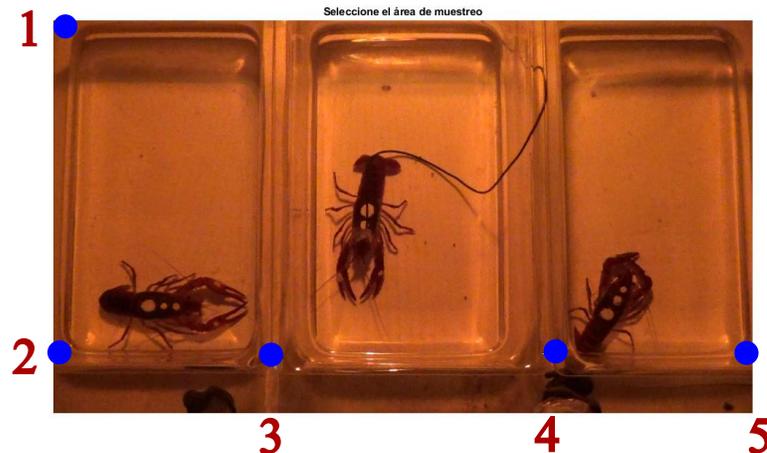


Figura 4.7: Selección de área de estudio por el usuario.

En la Figura 4.7 se muestra el orden en que se colocaron los puntos para seleccionar el área de estudio de tres acociles. Los puntos marcados serán siempre  $n + 2$  donde  $n$  es el número de acociles de estudio. El algoritmo puede estudiar hasta cuatro acociles a la vez, por lo que el proceso general de detección muestra cuatro acuarios; sin embargo, para la sección de Resultados, sólo se hablará de hasta tres acociles. La forma en que se tomaron las dimensiones es por medio del siguiente comando:

```
[x,y]=getpts;  
x=uint16(x);  
y=uint16(y);
```

En la variable  $x$  se almacenan las coordenadas de las abscisas para cada acuario; en la variable  $y$  sucede lo mismo pero no se utilizarán; sólo serán tomadas en cuenta la coordenada  $y$  de los puntos 1 y 2.

Una vez obtenidas las dimensiones para cada acuario, se utilizaron estas para dividir la imagen negativa (Figura 4.6) en el número de acociles que fue proporcionado anteriormente. Cada acuario será una imagen que se almacenará como se muestra en la Figura 4.8:

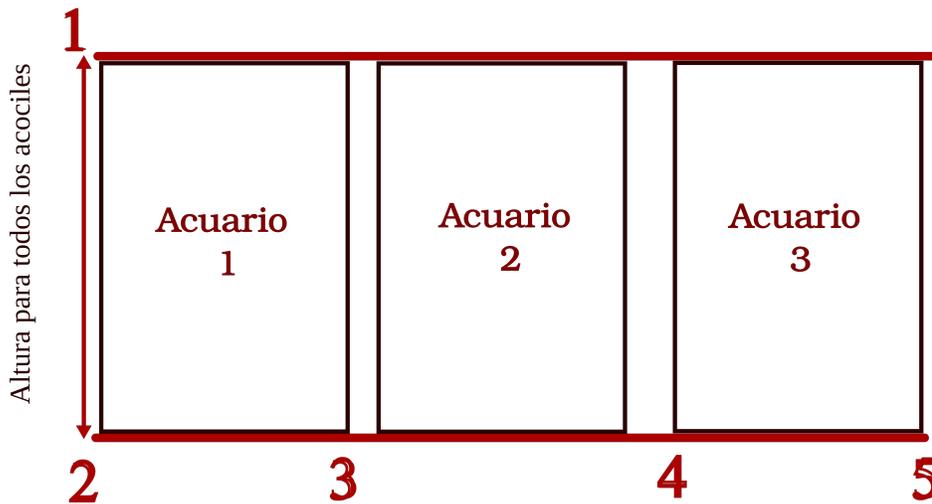


Figura 4.8: División de áreas que tomará el algoritmo.

Las dimensiones para cada acuario son tomadas de la siguiente manera:

```
acuاريو1=A(y(1):y(2),x(2):x(3),:);
acuاريو2=A(y(1):y(2),x(3):x(4),:);
acuاريو3=A(y(1):y(2),x(4):x(5),:);
```

donde  $A$  es la imagen original antes de cualquier proceso; los índices  $y$  representan las ordenadas de los puntos 1 y 2; estos índices fueron los mismos para todas las imágenes. Los índices en la variable  $x$  representan las abscisas para cada imagen; estos fueron variando de acuerdo a cuántos acociles se analizaron.

La Figura 4.9 es la representación para el análisis de tres acociles. En la parte de la derecha de cada imagen se graficará cada punto de trayectoria detectado para cada acuario.

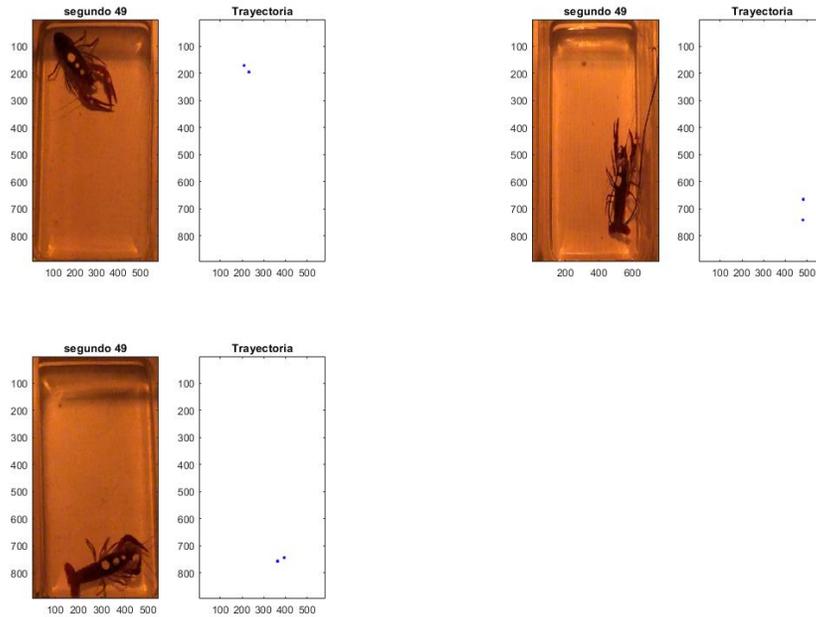


Figura 4.9: Selección de áreas para tres acociles.

## 4.2. Detección de círculos en puntos de interés

El análisis de conducta se realizó a través de dos algoritmos: Trayectoria y Posición de lado. El primero está enfocado en la detección de los círculos que representan la posición en su acuario(Figura 3.1a); el segundo está enfocado en la detección del círculo verde de su costado (Figura 3.1b) que representa el cambio de postura.

### 4.2.1. Trayectoria

Al seleccionar el área de la imagen a una más pequeña, que sólo muestre el acuario de interés, las coordenadas de los centros se ajustaron a las dimensiones de la nueva imagen, de manera que sólo habrá puntos dentro del espacio que representa el acuario.

El análisis de círculos como puntos de interés se realizó a partir de la Figura 4.6; a la cuál se aplicó la Transformada de Hough (sección 1.3) para detectar

circunferencias.

En MatLab se aplicó la función `imfindcircles(im, [rmin rmax]);` al ingresar la imagen de la Figura 4.6 seccionada en áreas a la función, se obtuvieron los centros y radios de los círculos que detectó; al dibujarlos sobre la imagen original con `viscircles(centro, radio)` tenemos los resultados de la Figura 4.10.

Una vez obtenidos los centros y radios de los círculos, se realizó un proceso de selección para tomar un círculo del arreglo y así, se identifique como la ubicación que tomó el acocil en su acuario. Para obtener el círculo de mayor radio se utilizó el siguiente código:

```
if length(centro)>1
    [x1,y1]=find(radios==max(max((radios))) );
    plot(centro(x1,1),centro(x1,2),'b. ');
    title('Trayectoria');
    xlim([1 480]),ylim([1 720]),hold on;
    axis ij;
end
```

El código presentado tiene como objetivo encontrar el radio de mayor longitud y, al identificarlo, se relacionó el radio con el centro al que pertenece, una vez asociados, las coordenadas del centro se graficaron en un área que representa el acuario, como un punto  $(x,y)$ .

Para la reconstrucción de la trayectoria se utilizaron los centros previamente detectados, con el fin de graficar los puntos  $(x,y)$  obtenidos en forma de una línea que sigue al acocil.

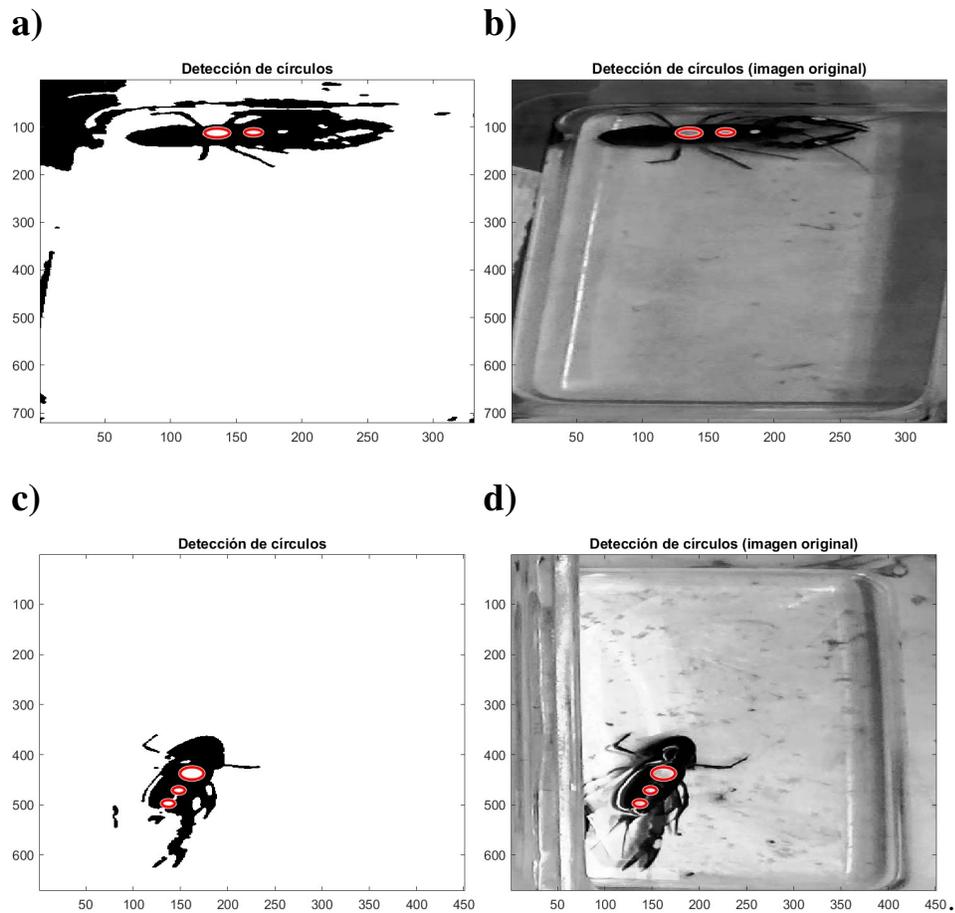


Figura 4.10: Detección de círculos en puntos de interés para los acociles en los acuarios 1 y 4. **a)** Círculos detectados para la Figura 4.6 del **Acuario 1**. **b)** Círculos en la imagen original. **c)** Círculos detectados para la Figura 4.6 del **Acuario 4**. **d)** Círculos obtenidos y marcados en la imagen original.

Los círculos detectados siempre mostraron el frente del acocil por lo que, al buscar su trayectoria, siempre se graficó un punto que identificó la vista al frente del crustáceo.

## Círculos falsos

Como son varias imágenes las que se analizaron, el algoritmo encontró círculos “falsos” que se guardaron como una posición errónea en la trayectoria. Cada que el acocil se desplazaba, también cambió la posición de que las o el mismo movimiento del agua creaba áreas brillosas que distorsionaron la imagen del acocil.

En esos cambios de luz aparecieron regiones cóncavas alrededor del cuerpo del acocil, como en la Figura 4.11, lo que provocó que aparecieran círculos que no eran de interés. Estos errores se contabilizaron y se registraron en los resultados.

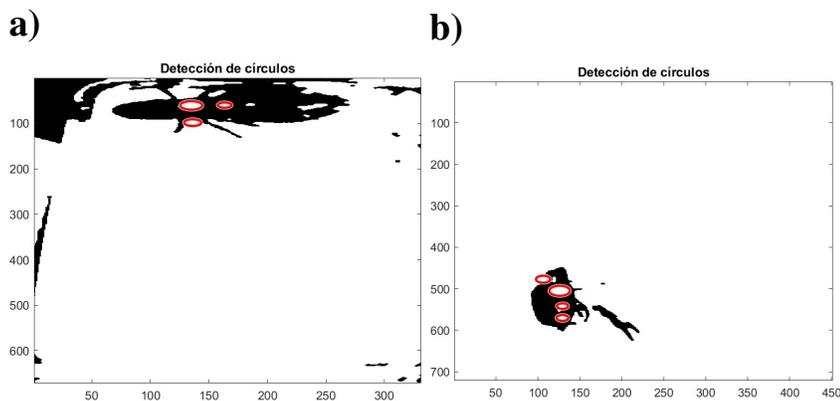


Figura 4.11: Detección de círculos falsos: **a)** Círculos falsos en imagen del **Acuario 1**. **b)** Círculos falsos en imagen del **Acuario 4**.

## Inmovilidad

Durante el análisis de imágenes, las coordenadas del círculo de mayor radio se almacenaron en un archivo externo, lo que permitió realizar otro análisis: inmovilidad. Para determinar la distancia que se mueve un acocil de una muestra a otra, se realizó una medición de distancia del tipo Euclidiana:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad [\text{Px}] \quad (4.4)$$

La ecuación 4.4 determinará la distancia, en pixeles, que recorrió un acocil de una muestra a otra, dentro de su acuario; para determinar la cantidad de movimiento se analizaron muestras cada 5 minutos; es decir, cada 300 cuadros se graficó la suma de las distancias recorridas por el acocil, en pixeles.

#### Movimiento

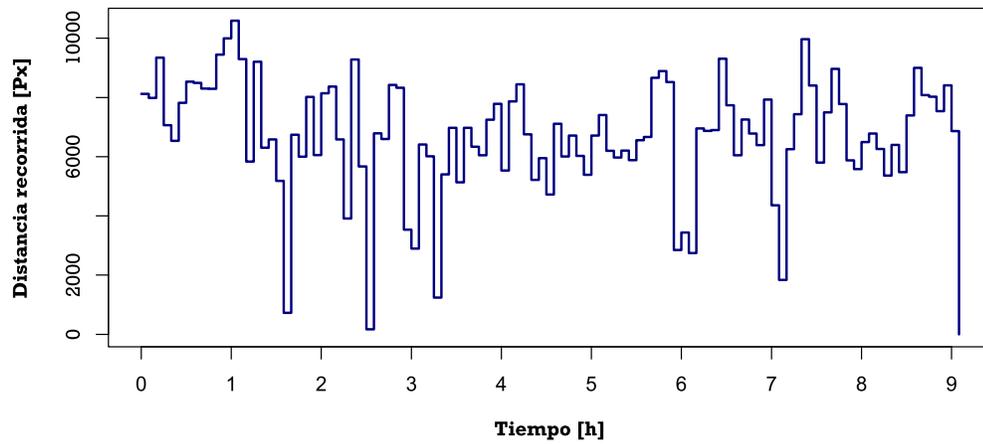


Figura 4.12: Cantidad de movimiento registrada para un análisis de conducta de un acocil con luz blanca.

#### Movimiento

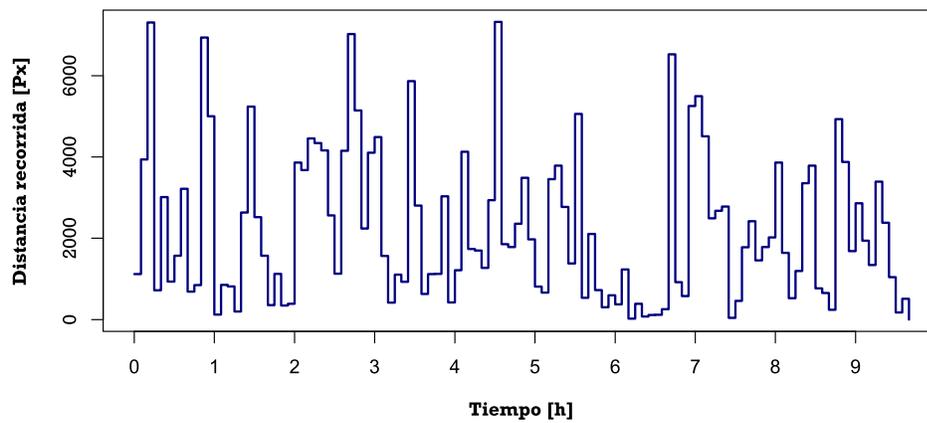


Figura 4.13: Cantidad de movimiento para un análisis de conducta de un acocil con luz roja

En la Figura 4.12 se representaron las cantidades de movimiento recorridas

por el acocil cada 5 minutos, en un registro de luz blanca; en el eje  $x$  se representaron las horas que duró el registro.

Comparando la cantidad de movimiento en un registro realizado con luz roja, se observa cómo la cantidad de movimiento se redujo.

Las Figuras 4.12 y 4.13 permiten, a un observador, determinar momentos en los que el acocil se mantuvo inmóvil; esta información es relevante debido a que se ha reportado que el acocil también puede dormir cuando está inmóvil (parado) (**Osorio-Palacios et al., 2021**), por lo que es otra forma de registrar los posibles momentos en los que el acocil presenta la conducta de sueño, aunque no esté de lado.

### 4.2.2. Posición de lado

El cambio de postura se detectó con el punto verde que se marcó en los costados del acocil (Figura 3.1). Por medio de una segmentación de imagen, se separaron las tres bandas RGB en imágenes individuales para aislar la banda G; una vez logrado ese paso, a la imagen original se le extrajo la banda G, dando como resultado una imagen que sólo muestra las áreas verdes que detectó la banda.

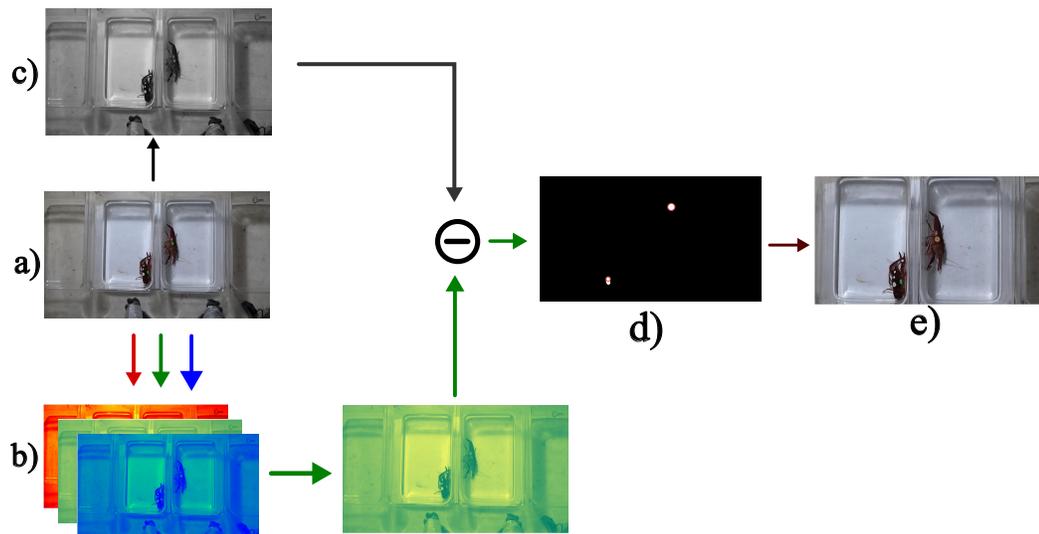


Figura 4.14: Proceso para detectar puntos verdes. **a)** Cuadro n-ésimo tomado del video. **b)** Separación de bandas RGB para sólo trabajar con la banda G. **c)** Conversión de RGB a escala de grises. **d)** Substracción de áreas verdes (ecuación 4.5). **e)** Círculos verdes detectados en la imagen.

### Separación de bandas RGB

Para almacenar las bandas RGB en imágenes individuales (Figura 4.14b), se realizó el siguiente proceso donde **imagen** es el cuadro n-ésimo original tomado del video.

```
R=imagen(:,:,1);  
G=imagen(:,:,2);  
B=imagen(:,:,3);
```

La notación  $(:, :, x)$  son las dimensiones que seleccionamos para la nueva imagen; dicho de otra manera, se tomaron todas las filas, todas las columnas y  $x$  fue la banda que se tomó de la imagen original (R, G o B). Cuando se separaron las tres bandas, sólo se tomó la banda G; después, con la ecuación 4.1, se convirtió la imagen original a una escala de grises.

Para separar las áreas verdes de la imagen original, se realizó una substracción de valores de intensidad como se muestra en la Figura 4.14d.

$$imgV = G - imGray \quad (4.5)$$

La ecuación 4.5 elimina de la banda G todos los valores de intensidad que no sean las áreas verdes, de manera que la imagen **imgV** sólo guardó los píxeles que corresponden a los tonos verdes que detectó la banda G en la imagen original.

### Detección de círculos a través de áreas verdes.

Para poder manipular la imagen **imgV** se realizó una operación de binarización con la función de MATLAB `imbinarize(im, "método")` en la que **im** es la imagen, en escala de grises, que se obtuvo en la ecuación 4.5 y '**método**' es la forma de binarizar la imagen.

Para este caso se utilizó el método '**global**' que se basa en el método de Otsu (**Otsu, 1979**) que determina el umbral óptimo de los niveles de intensidad de una imagen en escala de grises y así, realizar la binarización con dicho valor, determinando un 'fondo' y un 'frente'.

Al aplicar la binarización, nuestro frente son las áreas de verde detectadas (valor 1) y el fondo es el resto de la imagen (valor en 0). Cuando se tiene la imagen binaria, se le aplicó una operación morfológica para delimitar los límites de las áreas verdes.

$$bwmorph(im, "majority", 50)$$

Al terminar la operación, se aplicó la transformada de Hough para detectar los círculos que representan las áreas verdes. Cuando se obtuvieron los centros y los radios, se dibujaron encima de la imagen original (Figura 4.15), detectando así el verde que se marcó en el costado del acocil.

Cuando se detectaron círculos verdes, se marcó la postura del acocil como posición de lado; cuando no se detectaron círculos, el estado era alerta o explorando su acuario.

### Detección de áreas verdes con luz roja.

Para poder detectar el área verde con luz roja, se manipuló la cámara con las siguientes condiciones considerando la luz roja como poca luz:

- Low Lux.
- 25 fps.
- f1/8 (apertura de la cámara).
- Balance de blancos para interiores.

Durante el procesamiento se aplicó la función `imlocalbrighten()` que aumenta el brillo de imágenes que fueron tomadas con poca luz, como lo son los videos con luz roja. Esta función sólo se tomó en cuenta cuando la imagen tenía luz roja, con luz blanca se utilizó la imagen original sin ninguna modificación.

**a)**



**b)**



Figura 4.15: Detección de acocil de lado. **a)** Detección de acocil de lado con luz blanca. **b)** Detección de acocil de lado con luz roja.

En la Figura 4.15a se muestra la detección del cambio de postura de un acocil iluminado con luz blanca; la ventaja es que se utiliza la imagen original sin ningún procesamiento.

Para la Figura 4.15b se muestra la detección del punto verde cuando estaba de lado con luz roja; para este reconocimiento se debe utilizar `imlocalbrighten()` que recibe como argumento la imagen original, lo que subirá el valor de los tonos verdes; la desventaja de utilizar dicha función es que las sombras comenzaron a crear círculos falsos, por lo que se pueden detectar momentos en los que no esté de lado.

### Trayectoria y posición de lado

Con las coordenadas obtenidas en trayectoria y en la identificación de la postura de lado, se realizó una gráfica que superpone los valores de los dos casos: la trayectoria en color azul y la detección del cambio de postura en verde, de esta manera se puede apreciar en qué zonas de su acuario el acocil cambió su postura.

La Figura 4.16 permite a un observador identificar las zonas en las que el acocil se pone de lado y si tiene una zona preferida.

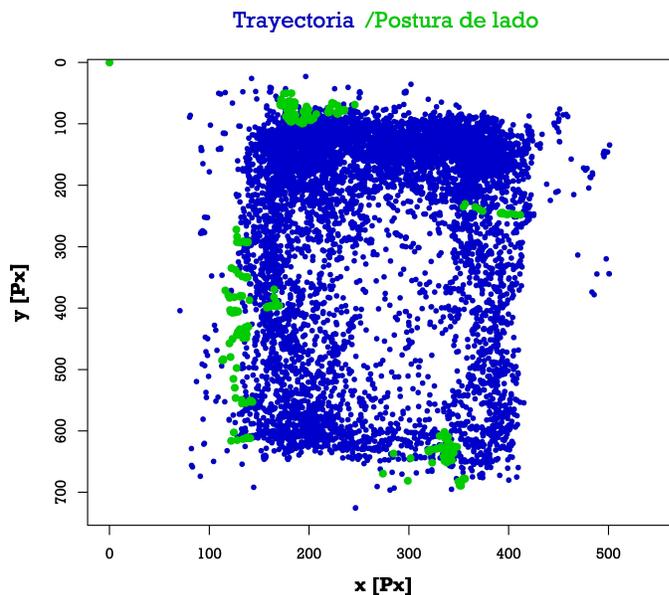


Figura 4.16: Superposición de coordenadas de trayectoria y coordenadas del cambio de postura.

### 4.2.3. Hipnograma

La detección del punto verde se registró en un archivo de texto externo para poder graficar el tiempo que estuvo de lado; se marcó **0** si el acocil estaba alerta o explorando el acuario; por el contrario se marcó **1** si el acocil estaba de lado.

Al final del procesamiento de imágenes, se obtuvo una gráfica de valores binarios que representa, en el eje de las abscisas, las horas que duró el registro, y en el eje de las ordenadas se muestra el estado lógico que representa la postura en la que se encontraba el acocil.

El fin de obtener esta gráfica es mostrar las horas totales que el acocil estuvo de lado y en qué hora del registro ocurrió; es decir, un hipnograma. En la Figura

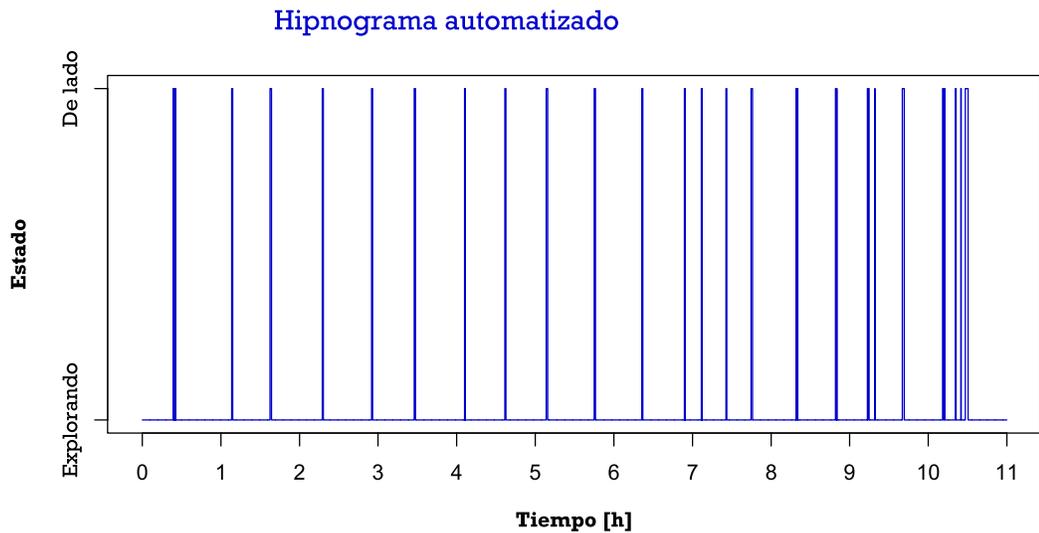


Figura 4.17: Registro de un acocil grabado por 10 horas con luz blanca.

4.17 se muestra el registro tomado a un acocil con luz blanca; el hipnograma optimizado con el programa propuesto concuerda con la actividad de día en los acociles; de acuerdo con Ramón et al., 2004, los acociles tienen mayor actividad en el día, por lo que su registro de posición de lado muestra menos estados en 1.

## Barrido de datos

Durante el análisis de postura puede ocurrir que sucedan falsas detecciones del área verde, lo que implica almacenar un estado lógico 1 **falso**. Al observar el archivo creado por el programa, se pudo localizar estos estados lógicos falsos, de manera que se recurrió a un barrido de datos:

- Caso 1: Si el valor central es igual a 1 y los dos valores anteriores y los dos valores posteriores son iguales a cero, el valor central cambiará a **0**.

$$[0 \ 0 \ 1 \ 0 \ 0] \implies [0 \ 0 \ 0 \ 0 \ 0]$$

- Caso 2: Si el valor central es igual a 1 y los dos valores anteriores son iguales a 0 y si alguno de los dos valores posteriores es igual a 1, el valor central cambiará su valor a 0.

$$[0 \ 0 \ 1 \ 1 \ 0] \implies [0 \ 0 \ 0 \ 1 \ 0]$$

$$[0 \ 0 \ 1 \ 0 \ 1] \implies [0 \ 0 \ 0 \ 0 \ 1]$$

- Caso 3: Si el valor central es igual a 0 y los dos valores anteriores y los dos posteriores son iguales a 1, el valor central cambiará a 1.

$$[1 \ 1 \ 0 \ 1 \ 1] \implies [1 \ 1 \ 1 \ 1 \ 1]$$

Cada caso se repitió al menos dos veces para cada hipnograma, teniendo un resultado como el siguiente.

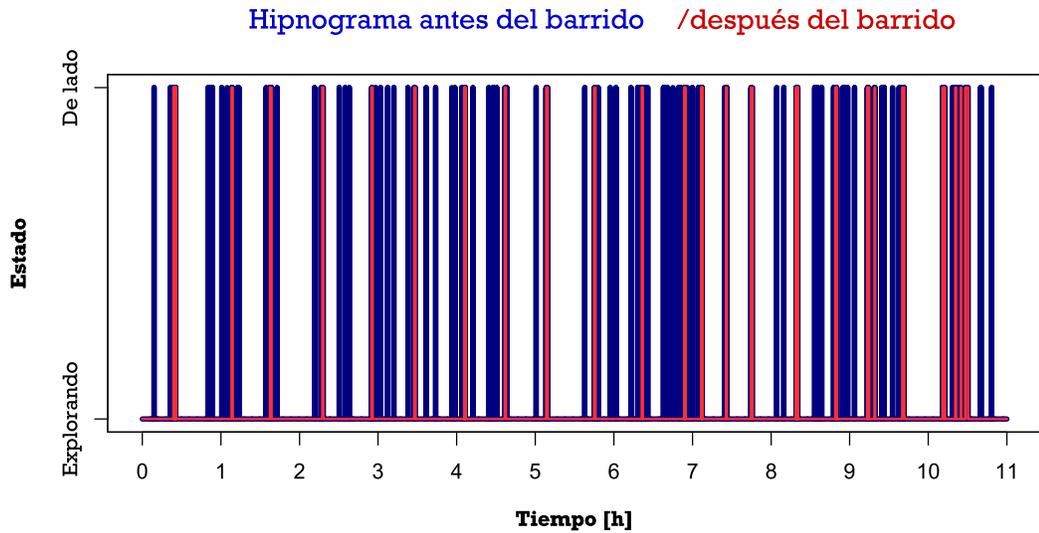


Figura 4.18: Comparación del hipnograma de la Figura 4.17

En la Figura 4.18 se muestra, de color azul, el hipnograma original que se obtiene del programa, antes del barrido de estados lógicos. De color rojo se representa el hipnograma después del barrido, que es el hipnograma que se presenta a los observadores como resultado final.

El hipnograma de color rojo se validó por medio de un observador experto que revisó todo el registro de conducta, lo que resultó en el nivel de confianza que el programa puede presentar como automatización de detectar el cambio de postura.

# Capítulo 5

## Resultados

---

El nivel de confianza del programa se evaluó por medio de una validación visual; es decir, un observador experto revisó todos los registros realizados en los experimentos para comprobar que el algoritmo realizó la detección correcta, tanto de trayectoria como del cambio de postura.

### 5.1. Validación del hipnograma

Para determinar si el programa realizó bien la detección del cambio de postura, un observador experto realizó la misma tarea, de manera que se graficó un hipnograma automatizado (Figura 4.17) y un hipnograma hecho por una persona.

Al superponer los dos hipnogramas en una misma gráfica, se obtuvo una comparación cuadro a cuadro de cuántas veces o en cuántos periodos el programa se equivocó en la detección del punto verde. En la Figura 5.1 se muestra, de color azul, el hipnograma automatizado que nos entregó el programa; de color rojo se muestra la validación que hizo un observador del registro.

## Hipnograma automatizado /Validación de un observador

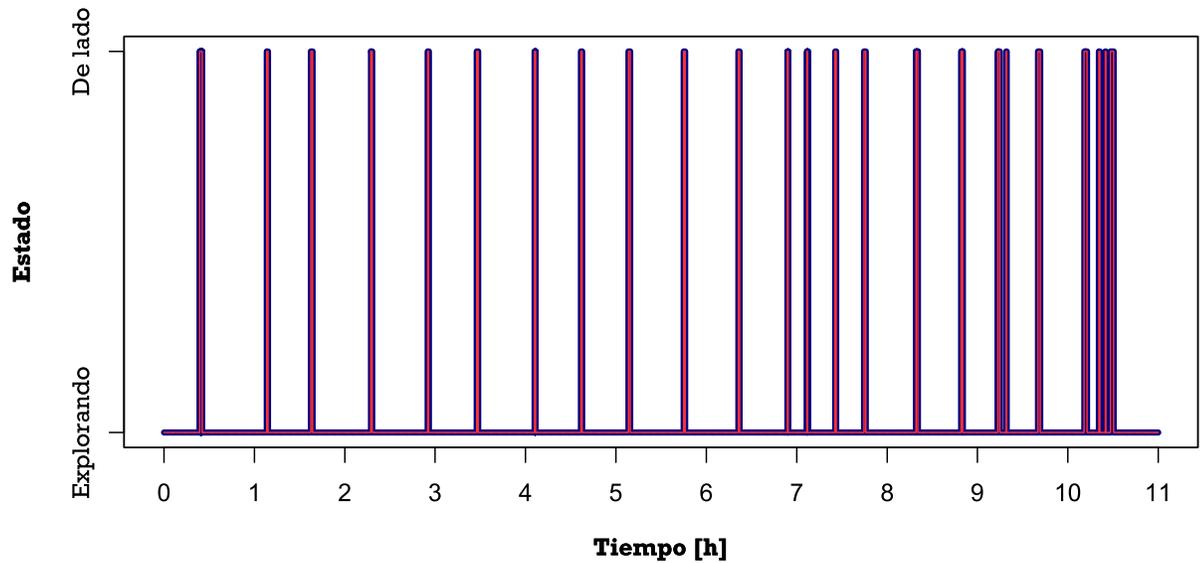


Figura 5.1: Registro de un acil grabado por 11 horas con luz blanca.

Para la validación se tomaron en cuenta los siguientes criterios:

- Si el programa detectó un punto verde y el observador también detectó un punto verde, indicando que el acil estaba de lado, se consideró como una lectura correcta: **(1)**.
- Si el programa no detectó puntos verdes y el observador no detectó que el acil estaba de lado, la lectura es correcta pero no se marcó: **(0)**
- Si el acil estaba explorando pero el algoritmo detectó áreas verdes se tomó como una lectura incorrecta y tampoco se marcó: **(0)**.

Después de la validación visual, se realizó una operación de substracción en la cuál al arreglo de valores lógicos que obtuvo el programa, se le restó el arreglo de valores que obtuvo el observador experto y se tomó el valor absoluto. En un arreglo posterior se guardaron los resultados de la operación.

En el arreglo **Resultado** se analizaron los valores que son iguales a 1, que representan una detección falsa de la postura.

$$|\text{Hipnograma original} - \text{Validación}| = \text{Resultado}$$

$$\begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Para determinar en qué medida el programa tuvo errores de detección, se tomaron muestras de 100 cuadros por segmentos hasta llegar a 585 muestras de error ( $58550 \text{frames} / 100$ ). Al validar 5 hipnogramas con 58500 cuadros cada uno, se obtuvieron los resultados de la Tabla 5.1.

Hipnograma	No. de acociles	Acociles analizados	Iluminación	EE %
1	2	1	Luz blanca	0.02
2	2	1	Luz roja	33.28
3	2	1	Luz roja	1.311
4	3	1	Luz roja	7.89
5	3	1	Luz roja	2.01

Tabla 5.1: Porcentajes de error para los hipnogramas obtenidos.

En la Tabla 5.1 se muestran los porcentajes de error (EE %) para 5 hipnogramas, teniendo como mejor resultado en luz blanca 0.02 % y en luz roja 1.311 % de error en la detección. Los hipnogramas se pueden consultar en el Anexo A.

Para analizar los resultados de la tabla 5.1 se debe tomar en cuenta que las condiciones en cada registro variaba de acocil en acocil, como el zoom de la cámara, la iluminación que recibe cada estanque, la distorsión del lente, etc.

## 5.2. Validación de trayectoria

La validación de trayectoria se elaboró tomando como lectura correcta cuando el programa detectó uno de los cinco puntos marcados en su cuerpo (Figura 5.2). Cuando la detección del círculo se marcó fuera del acocil o el programa no detectó ninguno de los 5 círculos cuando estaba explorando, se tomó como una lectura incorrecta.

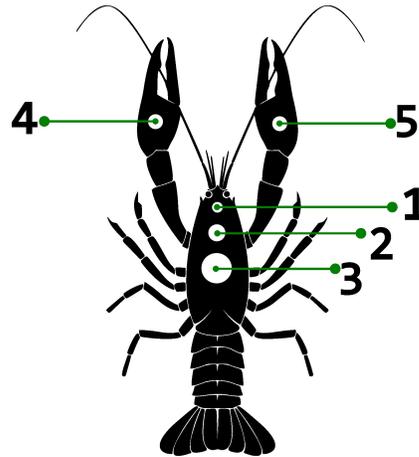


Figura 5.2: Selección de puntos para la trayectoria.

Se analizaron 10 muestras aleatorias de 5 minutos (300 cuadros) en las cuáles un acocil estaba explorando su acuario, cada muestra de la Tabla 5.3 representa la validación de trayectoria para un acocil de manera individual, pero los porcentajes totales de la Tabla 5.2 representan, de manera global, el nivel de confiabilidad tanto para 2 acociles registrados simultáneamente, como a 3.

	Detección de trayectoria					
	Punto 1	Punto 2	Punto 3	Punto 4	Punto 5	Nada
<b>Luz blanca</b>						
<b>2 acociles</b>						
Muestra 1	0	0.61	0.3733	0	0.1	0.0066
Muestra 2	0	0.78333	0.2166	0	0	0
<b>Luz roja</b>						
<b>2 acociles</b>						
Muestra 3	0.20666	0.16	0.58333	0	0.0133	0.0366
Muestra 4	0.06333	0.8866	0.0266	0	0	0.0233
Muestra 5	0.06333	0.9066	0.0166	0	0	0.01333
Muestra 6	0.87666	0.09	0.0333	0	0	0
<b>Luz roja</b>						
<b>3 acociles</b>						
Muestra 7	0.14	0.82	0.03	0.0066	0	0.0033
Muestra 8	0.1466	0.67	0.1733	0	0.0033	0.0066
Muestra 9	0.53	0.4	0.07	0	0	0
Muestra 10	0.5266	0.4366	0.0366	0	0	0

Tabla 5.2: Porcentajes de confiabilidad

En la Tabla 5.2 se registró la cantidad de veces que se detectó cada punto en cada prueba (Número de detecciones/300) para determinar cuál de los 5 puntos era más probable de ser detectado.

Al ser muestras aleatorias, hay momentos en los que los acociles cambiaron su postura, provocando que no se detectaran los puntos de trayectoria; esos eventos se dejaron fuera de la validación de trayectoria pues el cambio de postura se representó con el hipnograma.

En la Tabla 5.3 se realizó la suma de los porcentajes de detección de cada punto; en la segunda columna se muestra el porcentaje de confianza para la

detección de los círculos de trayectoria.

Muestra	% Aciertos	% Error
1	99.33	0.666
2	99.9	0
3	96.33	3.666
4	97.65	2.333
5	98.65	1.333
6	99.9	0
7	99.33	0.666
8	99.66	0.333
9	99.9	0
10	99.9	0

Tabla 5.3: Confiabilidad en trayectoria y error.

Después de realizar el promedio del porcentaje de éxito para todas las muestras, se encontró una confianza del 99.09% y un 0.8997% de error.

# Capítulo 6

## Discusión de resultados

Para explicar las variaciones del porcentaje de error de la Tabla 5.1, se analizaron las condiciones en las que los acociles fueron grabados; al analizar los registros se encontraron variaciones en la iluminación de cada acuario, el zoom de la cámara, la distorsión que sufrió la imagen por el lente de la cámara, etc.

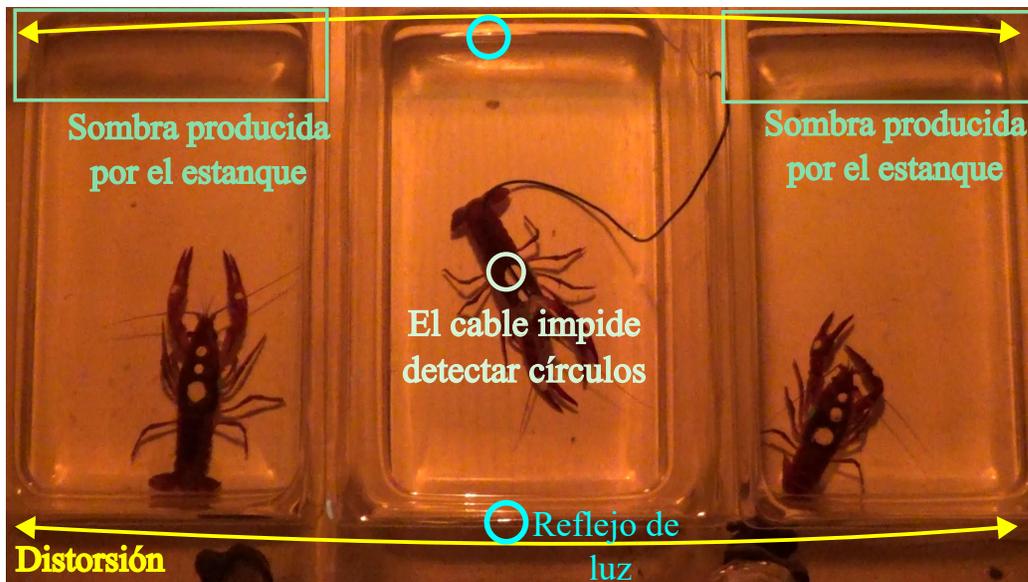


Figura 6.1: Problemas en la captura de imagen.

En la Figura 6.1 se muestran algunas de las condiciones que generaron una mala detección durante un registro de conducta. Los reflejos de luz y las sombras producidas por el acuario ocasionaron falsas detecciones tanto de los círculos de trayectoria, como el círculo verde, ocasionando que se almacenaran posiciones falsas o una postura equivocada.

Otro evento desfavorable es cuando los acociles cambiaron su postura lentamente de estar parados a estar de lado; esto ocasionó que la detección del punto verde fuera ambigua y presente variaciones en la lectura, provocando que no se grafiquen ventanas continuas de detección.

Por último, la lámpara de luz roja no logró iluminar a los acociles de manera uniforme de manera que los extremos de la imagen perdieron iluminación.

Estas situaciones estuvieron presentes para todos los registros de luz roja; en el caso de luz blanca la situación que causaba problemas eran los reflejos de luz pues crean formas que, para la detección de trayectoria y postura, resultaron en círculos falsos.

### **Confianza en trayectoria y postura**

Como se mencionó antes, la validación de trayectoria de la Tabla 5.2 fue analizada únicamente con segmentos de video en los que el acocil estaba parado y explorando debido a que el cambio de postura puede ocasionar que los puntos para el estudio de exploración no sean visibles para la cámara y no sea posible detectarlos.

Esta situación no implica que se dejaran de almacenar las coordenadas de posición del acocil puesto que, cuando cambió de postura y se detectaba el círculo verde, las coordenadas del centro también se interpretaron como la posición del acocil en su acuario, sólo que ahora indicaban que estaba de lado.

El porcentaje de confianza se interpretó de la siguiente manera: cuando se detectó uno de los cinco círculos pintados en su cuerpo, el círculo seleccionado como posición en el acuario era 99.09 % uno de ellos; el error se etiquetó como un círculo fuera del acocil o que no se detectó ningún círculo.

Para el error que presentaron los hipnogramas, se debe tomar en cuenta que, para luz blanca, el 0.02 % de error es debido a que no se debe procesar la imagen y se puede trabajar con el cuadro original. Es decir, los tonos de verde no sufrieron alteraciones.

El error de 33.28 % sucedió debido a que los acociles cambiaban su postura lentamente, lo que provocó que no se detectara la marca verde de manera continua o, por otro lado, el acocil seguía explorando pero dejaba visible su marca verde.

### Recomendaciones para el programa

Durante la elaboración del algoritmo se encontraron varias recomendaciones para correr el programa. La Tabla 6.1 contiene la manipulación de la cámara

Característica	Luz blanca	Luz roja
Valor de <i>umbral</i> (ec. 4.2)	55	105
Configuración de cámara	Automático	<ul style="list-style-type: none"> <li>■ Low lux activado</li> <li>■ 25 fps</li> <li>■ Balance de blancos para interiores</li> </ul>
Código adicional	Ninguno	<code>imlocalbrighten()</code>
Área de registro	No tener objetos de color verde cerca del lugar de registro.	
Rango [ <i>rmin</i> <i>rmax</i> ]	[7 – 13]	[5 – 16]

Tabla 6.1: Recomendaciones de toma de captura para el registro.

y de variables que se tomaron en cuenta para los experimentos, de manera que los resultados fueran los mejores posibles. De este modo, el usuario puede determinar si las condiciones le parecen útiles en experimentos a futuro o si requiere hacer ajustes para sus registros.



# Capítulo 7

## Conclusiones

---

El objetivo de la tesis se cumplió debido a que se desarrolló, implementó y validó un programa capaz de detectar el cambio de postura del acocil y su trayectoria de manera automatizada.

Los niveles de confianza en la trayectoria y en la detección del cambio de postura son altos y, considerando que un observador experto necesita de hasta 3 a 5 días para analizar un registro de conducta (de 24 horas de duración que incluye a 2 acociles) el programa resulta ser de alta confianza debido a que realiza la misma tarea en un rango de 8 a 10 horas, generando además el hipnograma, la reconstrucción de trayectoria y la serie de tiempo con los momentos de inmovilidad.

Además, el análisis de inmovilidad puede resultar especialmente útil para el estudio del sueño pues se ha reportado que los acociles también duermen cuando están parados (**Osorio-Palacios et al., 2021**), lo que para un observador resulta en largos análisis para detectar inmovilidad. El programa permitirá que el investigador sólo requiera de analizar la actividad eléctrica cerebral correspondiente a la inmovilidad si desea corroborar que el animal está durmiendo, evitando analizar registros completos, lo que permitirá ahorrar tiempo, recursos computacionales y energía.



# Capítulo 8

## Trabajo a futuro

---

El trabajo presentado tiene distintas áreas en las que se puede seguir trabajando para poder estudiar la conducta en acociles. Por ejemplo, una conducta de interés para la investigación es la interacción social en acociles; donde se colocan 2 a 3 en el mismo acuario. Poder describir la trayectoria y cuántas veces un acocil interactúa con otro puede ser de mucha ayuda para poder estudiar los encuentros agonistas que despliegan estos crustáceos (**Jiménez-Morales et al., 2018; Suryanto et al., 2023**).

Otra forma de seguir trabajando con la conducta de acociles es buscar nuevos métodos de procesamiento de imágenes, esto con el fin de implementar métodos que no requieran el pintado de acociles de manera frecuente. Una posibilidad es trabajar con redes convolucionales entrenadas con las posturas del acocil.

La formación del “esqueleto” del acocil, que pueda dar información sobre el movimiento de las patas y las quelas, sería útil para el análisis conductual y de marcha. Una forma en la que este problema se puede abordar es mediante la reconstrucción de un modelo 3D del acocil, otra es mediante diferentes métodos de procesamiento de imágenes que permitan aislar el tórax del acocil de sus patas (**Plum et al., 2023**).



# Apéndice A

## Hipnogramas automatizados

Los hipnogramas obtenidos en los experimentos realizados se pueden consultar en esta sección. De color azul se representa el hipnograma automatizado y de rojo el hipnograma que obtuvo un observador experto, esto con el fin de poder comparar los resultados.

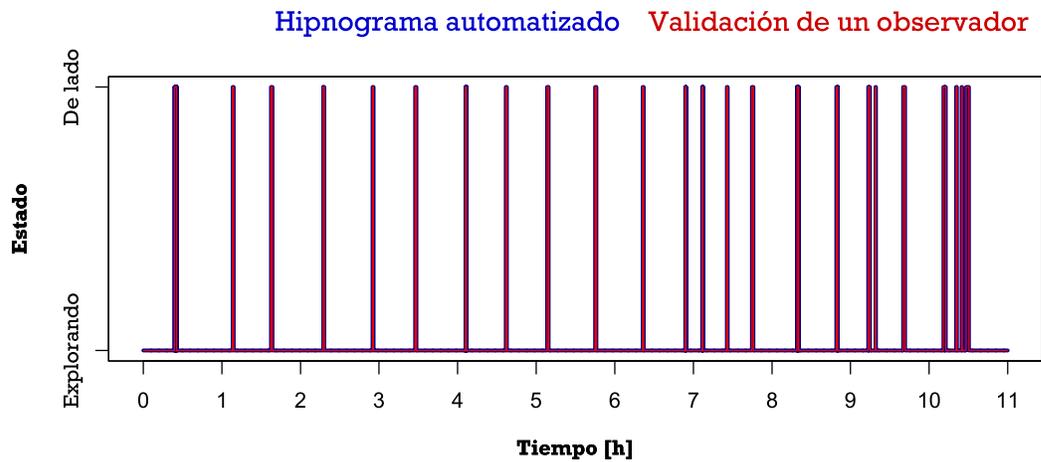


Figura A.1: Registro de un acocil grabado por 10 horas con luz blanca.

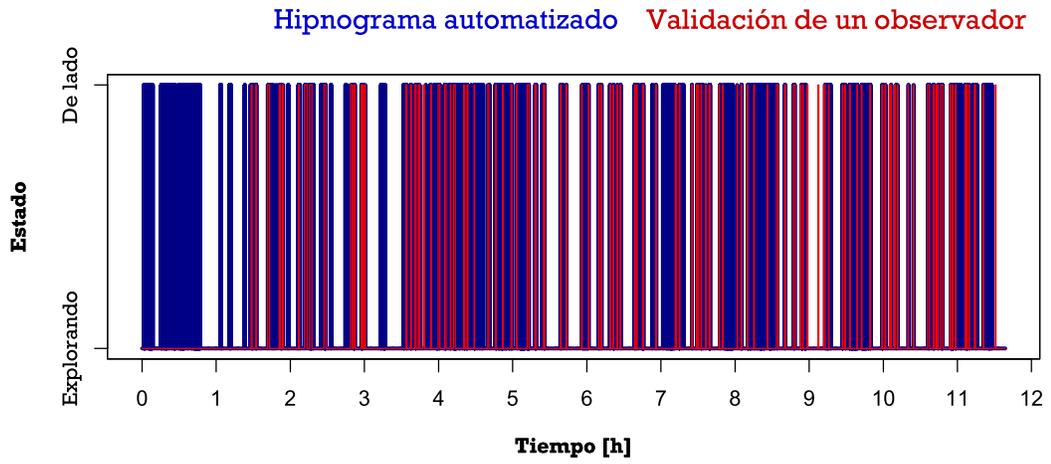


Figura A.2: Hipnograma de un acocil con luz roja

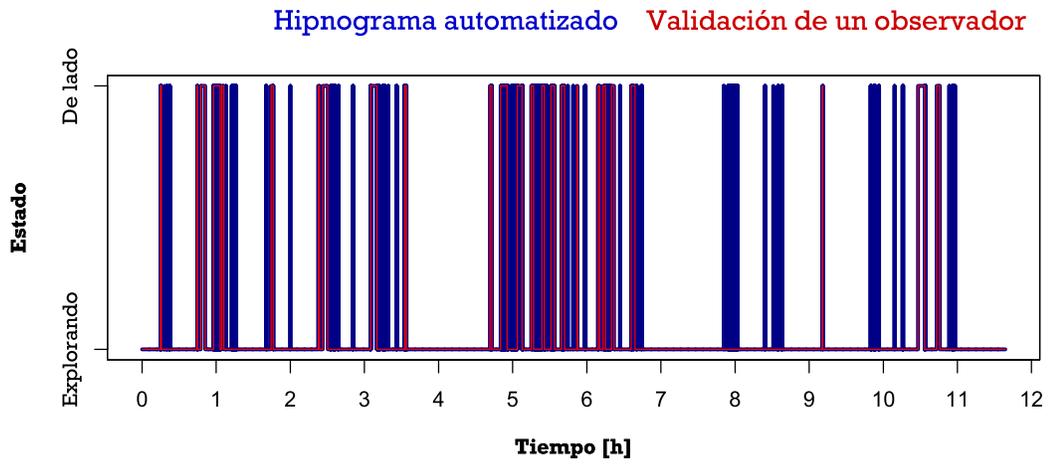


Figura A.3: Hipnograma de un registro con 3 acociles.

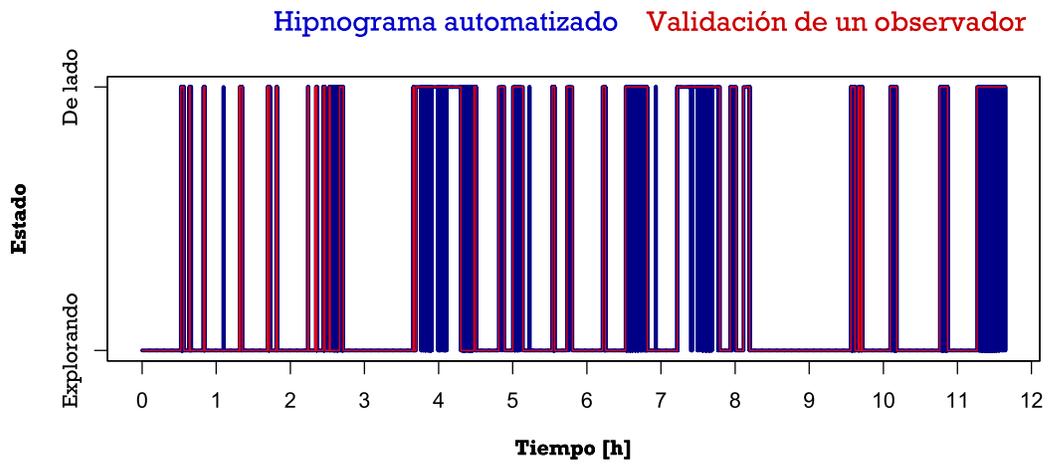


Figura A.4: Hipnograma de un registro con 2 acociles.

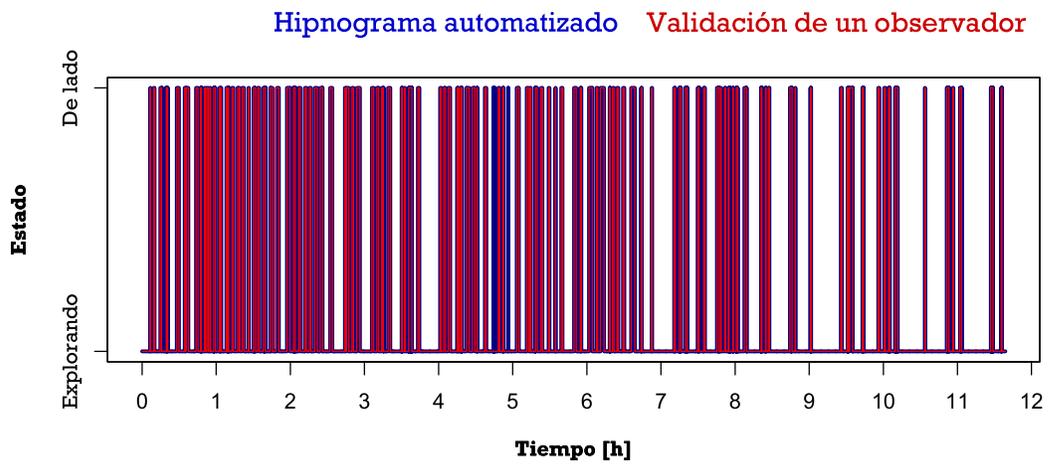


Figura A.5: Hipnograma de un registro con 2 acociles.



## Apéndice B

### Anatomía externa del acocil

En este anexo se puede consultar la anatomía externa del acocil, de esta manera, el lector puede identificar las partes del cuerpo del acocil que fueron marcadas con pintura para los experimentos.

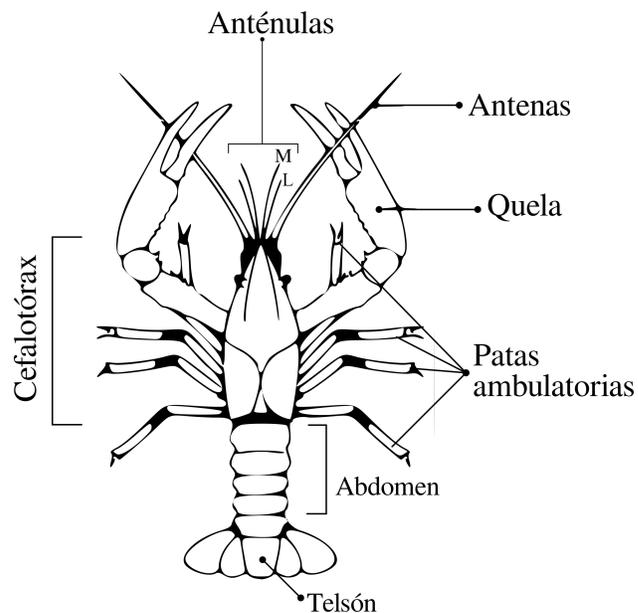


Figura B.1: Anatomía externa del acocil *Procambarus Clarkii*. Modificado de Horner et al., 2008.



# Apéndice C

## Archivos obtenidos del programa

Cuando se corre el programa, el usuario debe ingresar el nombre del directorio (<nombreDirectorio>) en el que se crearán los archivos y se guardará la información. Al término de un análisis conductual, los archivos que recibe el usuario tienen la siguiente estructura:

Nombre del archivo	Descripción
<nombreDirectorio>Coordenadas.txt	Contiene pares coordenados $(x,y)$ que representan la trayectoria
<nombreDirectorio>CoorPVerde.txt	Contiene pares coordenados $(x,y)$ que representan las coordendas donde se detectó el cambio de postura
<nombreDirectorio>Hipnograma.txt	Valores lógicos, 0 y 1, que representan la postura del acocil durante el registro (1 - De lado y 0 - Parado, explorando)
tablaAcocil<numAcocil>.txt	Hipnograma después del barrido de datos
tiemposAcocil<numAcocil>.txt	Conteo (en segundos) que se detectó cada postura durante el registro.s

Tabla C.1: Archivos de salida para el usuario.

Cualquier manipulación posterior puede ser aplicada con los archivos, ya dependerá del usuario lo que quiera evaluar de la información.

**tiemposAcocil<numAcocil>.txt**

El archivo que se obtiene de la Tabla C.1 tiene la siguiente información:

---

Segundos	Estado lógico	Inicio	Fin	Hora del registro
Integer	Integer	Integer	Integer	String

Tabla C.2: Formato del archivo tiemposAcocil<numAcocil>.txt

En la Tabla C.2 se representan los tipos de datos que contiene el archivo; es importante conocerlos para poder darles uso.

El archivo tiemposAcocil<numAcocil>.txt es el que contiene los segmentos de video que se deben analizar pues en la columna “Segundos” se tiene el tiempo que duró en cada estado lógico; de esta manera el usuario decidirá qué segmentos del video o qué cuadros quiere analizar.

# Apéndice D

## Manual de usuario

---

El usuario puede correr, de manera rápida, el programa presentado, considerando los siguientes puntos.

### D.1. Archivo `leer.m`

En este archivo, el usuario debe proporcionar la ruta del video que quiere analizar, el nombre de la carpeta en la que quiere guardar las imágenes y el número de cuadros por segundo que quiere tomar como muestreo.

La Figura D.1 contiene las líneas de código que el usuario debe manipular para la creación de la carpeta que contendrá las imágenes que el programa analiza.

`A=VideoReader('ruta de tu video')`: Se debe colocar, dentro de las comillas, la ruta de acceso para el video, incluyendo el nombre del video y su formato. Si el video está en la misma carpeta en la que se corre el programa sólo es necesario ingresar el nombre del video con su formato. Ejemplo:

```
A=VideoReader("20240611192703.MTS");
```

Los formatos de video que el comando soporta se pueden consultar en [este link](#).

`'nombre_carpeta_principal'`: En esta parte, el usuario debe asignar el nombre del directorio principal donde guardará todas las imágenes.

`'subcarpeta_principal'`: En esta parte, el usuario debe asignar el nombre del directorio principal donde guardará todas las imágenes.

`fps`: Se debe sustituir por el número cuadros que se tomarán entre muestras; es decir, el número de cuadros por segundo.

```
clc; clear; close all;

A=VideoReader("ruta_de_tu_video"); ← Ruta del registro
nombre='subcarpeta_principal';
i=1;
j=1000; ← Carpeta secundaria
ST='.jpg'; ← Guardará 1000/fps
cuadros=A.NumFrames; ← número de imágenes.
cuadros=cuadros/1;
cuadros=int16(cuadros);
mkdir "nombre_carpeta_principal" ← Carpeta principal
n=1; ← Contendrá todas las
x=1; imágenes.
corte=1;
mensaje='Lectura de imagen';
cd "nombre_carpeta_principal" ←
while n<cuadros
    disp(mensaje);
    disp(n);
    parte=num2str(n);
    crear=strcat(nombre,parte);
    mkdir(crear);
    vid=read(A,[i j]);
    for x=1:fps:size(vid,4) ← fps: intervalo entre
        Sx=num2str(corte); ← cada muestra
        frame=strcat(Sx,ST);
        direc=vid(:,:,,x);
        cd(crear)
        imwrite(direc,frame);
        cd ..
        corte=corte+1;
    end
    n=n+1;
    i=i+1000;
    j=j+1000;
end
cd ..
disp('Fin lectura de imágenes')
clear A;
```

Figura D.1: Fragmento de código del archivo `leer.m`

## D.2. Programa principal

Cuando se tienen todas las imágenes almacenadas, el usuario debe ingresar el nombre de la carpeta principal y la subcarpeta para que el programa reconozca los directorios y los pueda manipular. Al inicio del código principal, se deben localizar las líneas 9 y 10 que se presentan a continuación:

```
carpeta='subcarpeta_principal';  
buscarCarp='nombre_carpeta_principal';
```

Por ejemplo, para uno de los registros que se llevaron a cabo en el desarrollo del trabajo, se asignaron los siguientes nombres:

```
carpeta='prueba13';  
buscarCarp='imagenes13';
```

Cuando buscamos los directorios en el sistema tenemos la siguiente forma:

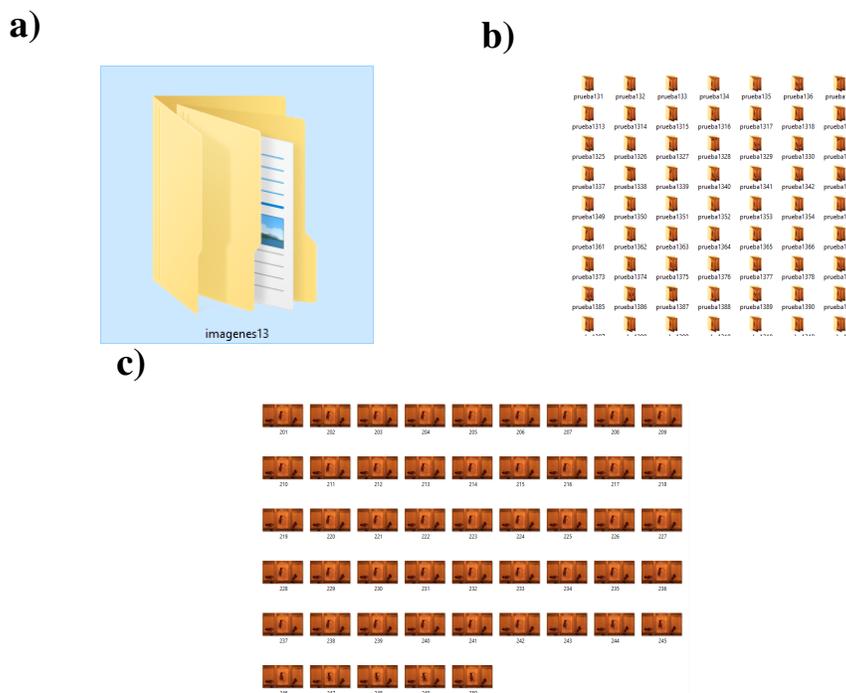


Figura D.2: Muestra de un directorio para un registro de luz roja. a) Carpeta principal. b) Subcarpeta. c) Contenido de una carpeta.

En la Figura D.2 se muestra el contenido de una carpeta principal para un

registro de luz roja. El contenido de la carpeta **imagenes13** se divide en varias carpetas con el formato "prueba13x" en donde **x** va cambiando de manera ascendente; cada carpeta contiene cierto número de imágenes que se irán guardando. El número de imágenes por carpeta será igual a  $\frac{1000 \text{ imagenes}}{\# \text{ fps}}$ ; por ejemplo, si son 30 fps, el número de imágenes por carpeta será de  $\frac{1000 \text{ imagenes}}{30 \text{ fps}} = 33$ . Se debe considerar la parte entera en el resultado. Para las pruebas realizadas se tomaron  $\frac{1000 \text{ imagenes}}{20 \text{ fps}} = 50$  para tener números cerrados en la validación.

### D.2.1. Detección de círculos para luz roja y blanca

El usuario debe tomar en cuenta las líneas 121 y 123 (en el código principal) en la que se asignan los valores de radio mínimo y máximo de la búsqueda de los círculos de color verde.

```

1   %[centroV,radioV]=separarVerde(imlocalbrighten(acociles{
    aux},0.4),5,16);
2   %%Comentar si es luz blanca
3   [centroV,radioV]=separarVerde(acociles{aux},5,16);
4   %%Comentar si es luz roja

```

Se debe comentar la línea 1 si se analizará un video con luz blanca; si se trata de un registro con luz roja la línea 3 se debe dejar como código y la línea 3 se debe comentar.

### D.2.2. Número de acociles de análisis

En la línea número 20, del código principal, se debe ingresar el número de acociles que el usuario desea estudiar; el mínimo es 1 y el máximo son 4; para obtener mejores resultados se recomienda que el máximo número de acociles sea 3. Este número es importante ya que con él se graficarán todas las figuras de salida que el usuario puede analizar.

```

num=1;   %Numero de acociles que se analizaran

```

### D.2.3. Número de carpetas de análisis

El usuario puede determinar el número de carpetas que desea analizar; en la línea del código 66-68 se encuentran las variables que controlan el conteo. La forma de saber cuántas carpetas tiene el registro total es a partir de la carpeta que

se creó en la Sección D.1. En las carpetas que contiene el directorio principal (Figura D.2a), se debe tomar el número final de carpeta; ese es el que se debe colocar en la variable `finConteo`, que será la encargada de finalizar el análisis del programa.

```
conteo=1;
correo=conteo;
finConteo=7;
```

#### D.2.4. Umbral de binarización

Como se mencionó en la Tabla 6.1, durante los experimentos realizados, se determinaron valores que pueden ser tomados en cuenta para analizar registros en futuras ocasiones; uno de ellos es el valor de la variable *umbral* que se utiliza para la binarización de las imágenes.

Dependiendo del tipo de luz que se analice, se debe cambiar el valor a 105 (luz roja) o 55 (luz blanca). Estas instrucciones sirven para que el usuario pueda utilizar el programa con las especificaciones que crea necesarias.

### D.3. Programa ejecutándose

Cuando el usuario corre el programa, este le pedirá que introduzca el nombre que le quiera dar a su registro, como se muestra en la Figura D.3.

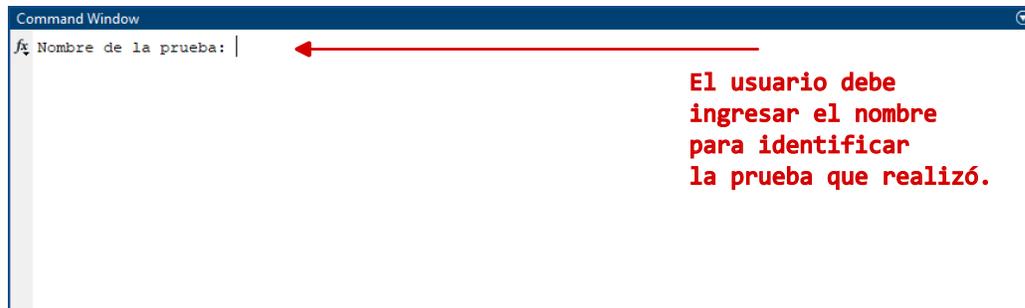


Figura D.3: Nombre de la prueba realizada

Para este ejemplo, se llamó al directorio "manualUsuario". En el siguiente paso, el programa le pedirá al usuario que marque las dimensiones de los acociles que quiere analizar; para este ejemplo se analizó 1 acocil.



Figura D.4: Selección de área de análisis.

En la Figura D.4 se seleccionó al acocil del acuario tres para el ejemplo. A partir de este punto, el programa sólo mostrará la carpeta que está analizando, de esta manera se evita que la memoria que usa MatLab para presentar Figuras se conserve para el análisis.

Una vez que el programa termina su ejecución, la consola sólo mostrará en la variable  $g$  el número de Figura que se creó y el tiempo que se detectó al acocil en su posición de lado.

```
1 Nombre de la prueba: manualUsuario
2 2
3
4 3
5
6 4
7
8 5
9
10 6
11
12 7
13
14
15 g =
```

```

16
17 Figure (5) with properties:
18
19     Number: 5
20     Name: ''
21     Color: [0.9400 0.9400 0.9400]
22     Position: [440 378 560 420]
23     Units: 'pixels'
24
25 Show all properties
26
27 Tiempo total dormido: 0h:0m:0s

```

El usuario puede corroborar que los datos se guardaron de manera correcta buscando en su sistema la carpeta que creó para su prueba; para este caso se buscó en el sistema a la carpeta "manualUsuario".

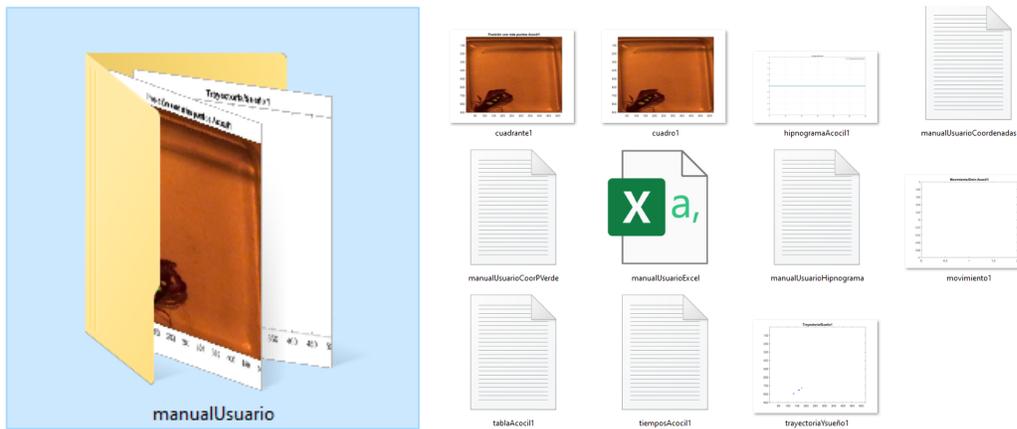


Figura D.5: Productos obtenidos de una ejecución del programa.

La Figura D.5 muestra los productos obtenidos de la ejecución del programa para una muestra de 300 imágenes. También se muestran los nombres de los archivos obtenidos como se mencionó en la Sección C.



# Apéndice E

## Código

---

En este anexo se presenta el código utilizado para los análisis de conducta que se realizaron en los experimentos.

### **main.m**

```
1 clc; clear; close all
2 warning off
3
4
5 prueba=input("Nombre de la prueba: ','s');
6 mkdir(prueba)
7 tipo='.jpg';
8 ver=1;
9 carpeta='prueba13';
10 buscarCarp="imagenes13";
11
12 cd(buscarCarp)
13 cd prueba131
14 A=imread("1.jpg");
15 cd ..
16 cd ..
17
18 imshow(A),title("Seleccione el area de muestreo");
19
```

```

20 num=1;      %Numero de acociles que se analizaran
21
22 [x,y]=getpts;
23 x=uint16(x);
24 y=uint16(y);
25 close all
26 i=2;
27
28 if num>2
29     cuadro=6;
30 else
31     cuadro=3;
32 end
33
34 acociles=cell(1,num);
35 negativos=cell(1,num);
36 acocilesRojo=cell(1,num);
37
38 im=1;
39
40 while im<=num
41     acociles{im}=A(y(1):y(2),x(i):x(i+1),:);
42     i=i+1;
43     im=im+1;
44 end
45
46 %umbral=105;
47 umbral=105; %%%Para luz blanca
48 dormido=1;
49 punto4=0;
50 xPos=[0 0 0 0];
51 yPos=[0 0 0 0];
52 vPos=[0 0 0 0];
53 centroVx=[0 0 0 0];
54 centroVy=[0 0 0 0];
55
56 cd(prueba);
57 muestra=fopen(strcat(prueba,'Coordenadas.txt'),'wt')

```

```

;
58 puntoVerde=fopen(strcat(prueba, 'Hipnograma.txt'), "wt
   ");
59 xyVerde=fopen(strcat(prueba, 'CoorPVerde.txt'), "wt");
60 validar=fopen(strcat(prueba, 'Excel.csv'), "wt");
61 cd ..
62
63 %conteo=112;
64 %conteo=345;
65 %conteo=435; 459
66 conteo=187;
67 correo=conteo;
68 finConteo=192;
69 %finConteo=350;
70 corte=9301;
71 while conteo<finConteo
72
73     while ver<=50
74         buscar=strcat(num2str(corte), tipo);
75         cd(buscarCarp)
76         cd(strcat(carpeta, num2str(conteo)))
77         imagen=imread(buscar);
78         cd ..
79         cd ..
80         [c,T]=negativo(imagen, umbral);
81
82         %%%% Imagen en pantalla completa
83         set(gcf, 'Position', get(0, 'Screensize'));
84
85         im=1;
86
87         i=2;
88         %%%%%%%%%Imágenes originales
89         while im<=num
90             acociles{im}=imagen(y(1):y(2), x(i):x(i
+1), :);
91             negativos{im}=T(y(1):y(2), x(i):x(i+1));
92             i=i+1;

```

```

93         im=im+1;
94     end
95     im=1;
96     figure(1)
97
98     i=1;
99     j=2;
100    k=3;
101    aux=1;
102    veces=1;
103
104    while aux<=num
105
106        [centro ,radio]=imfindcircles(negativos{
aux},[6 30]);
107
108        if isempty(centro)~=1
109            [ejeX,ejeY]=find(radio==max(max((
radio)))) );
110 %            subplot(2,cuadro ,j),plot(centro(
ejeX,1),centro(ejeX,2),'b. ');title('Trayectoria ')
;
111            xPos(aux)=centro(ejeX,1);
112            yPos(aux)=centro(ejeX,2);
113 %            xlim([1 size(negativos{1},2)]),
ylim([1 size(negativos{1},1)]),hold on;
114 %            axis ij;
115        end
116
117        %%%%%Detectar verde
118        %acocilesRojo{aux}=imlocalbrighten(
acociles{aux},0.4);
119        %%%Descomentar si es para luz roja
120
121        %[centroV ,radioV]=separarVerde(
imlocalbrighten(acociles{aux},0.4),5,16);
122        %%%Comentar si es luz blanca
123        [centroV ,radioV]=separarVerde(acociles{

```

```

aux},5,16);
124         %%Comentar si es luz roja
125
126         if isempty(centroV)~=1
127             if length(centroV)<3
128                 [ejeX,ejeY]=find(radioV==max(max
((radioV))) );
129 %                 subplot(2,cuadro,j),plot(
centroV(ejeX,1),centroV(ejeX,2),'g. '),hold on;
130 %                 title('Posicion de lado ');
131 %                 xlim([1 size(negativos{1},2)])
,ylim([1 size(negativos{1},1)]),hold on;
132                 %xlim([1 700]),ylim([1 800]),
hold on;
133 %                 axis ij;
134                 %subplot(2,cuadro,k),plot(punto4
,1,'b. '),title("Tiempo dormido");
135                 %xlim([1 100000]),ylim([0 2]),
hold on;
136                 vPos(aux)=1;
137                 %punto4=punto4+20;
138                 centroVx(aux)=centroV(1,1);
139                 centroVy(aux)=centroV(1,2);
140
141
142                 end
143             else
144                 vPos(aux)=0;
145             end
146
147
148         subplot(2,cuadro,im),imagesc(acociles{
aux});
149         title("segundo "+num2str(dormido))
150         viscircles(centro,radio)
151         viscircles(centroV,radioV,'Color','g');
152
153         j=j+3;

```



```

185     disp(conteo)
186     %keyboard
187
188 end
189
190 cd(prueba)
191 fclose(muestra);
192 fclose(puntoVerde);
193 fclose(xyVerde);
194 fclose(validar);
195 cd ..
196
197 pause(2)
198 close all
199
200 im=1;
201 i=2;
202 j=2;
203
204     cd(prueba)
205     [uno,dos,tres,cuatro,cinco,seis,siete,ocho]=
textread(strcat(prueba,'Coordenadas.txt'),"%f %f
%f %f %f %f %f %f",'delimiter','\n');
206     [unoV,dosV,tresV,cuatroV]=textread(strcat(prueba
,'Hipnograma.txt'),"%d %d %d %d");
207     [xV1,yV1,xV2,yV2,xV3,yV3,xV4,yV4]=textread(
strcat(prueba,'CoorPVerde.txt'),"%f %f %f %f %f %
f %f %f");
208     coorV=[[xV1],[yV1],[xV2],[yV2],[xV3],[yV3],[xV4
],[yV4]];
209     cd ..
210     coor=[[uno],[dos],[tres],[cuatro],[cinco],[seis
],[siete],[ocho]];
211 %     [unoV,dosV,tresV,cuatroV]=textread('
hipnograma210524.txt',"%d %d %d %d");
212     hVerde=[[unoV],[dosV],[tresV],[cuatroV]];
213     %%%%%%%%%Imagenes originales
214     while im<=num

```

```

215         %trayectoria(carpeta,buscarCarp,1,y(1),y(2),
x(i),x(i+1),num,coor(:,j-1),coor(:,j),im,
finConteo,prueba);
216         movimiento(coor(:,j-1),coor(:,j),im,prueba);
217         division(acociles{im},im,coor(:,j-1),coor(:,
j),prueba);
218 %         hipnograma(hVerde(:,im),hVerde(:,im+2),im
,prueba)
219         graficaV=hipnograma2(hVerde(:,im),im,prueba)
;
220         sleep(graficaV,coorV(:,j-1),coorV(:,j),coor
(:,j-1),coor(:,j),size(acociles{1},2),size(
acociles{1},1),im,prueba);
221         i=i+1;
222         im=im+1;
223         j=j+2;
224     end
225 %
226 % tiempo=hipnograma(prueba,im);

```

## E.1. Funciones

### **borrarUnos.m**

```
1 function limpio=borrarUnos(A)
2     i=3;
3     while i<(length(A)-2)
4         if A(i)==1
5             if A(i-2)==0 && A(i-1)==0
6                 if A(i+2)==0 && A(i+1)==0
7                     A(i)=0;
8                 end
9             end
10        end
11        i=i+1;
12    end
13    limpio=A;
14 end
```

### **casoA.m**

```
1 function limpio=casoA(A)
2     i=3;
3
4     while i<length(A)
5         if A(i)==1
6             if A(i-2)==0 && A(i-1)==0
7                 if A(i+2)==1 && A(i+1)==0
8                     A(i)=0;
9                 end
10            end
11        end
12        i=i+1;
13    end
14    limpio=A;
15
16
17 end
```

### **casoB.m**

```
1 function limpio=casoB(A)
2
3     i=3;
4
5     while i<length(A)
6         if A(i)==1
7             if A(i-2)==0 && A(i-1)==0
8                 if A(i+2)==0 && A(i+1)==1
9                     A(i)=0;
10                end
11            end
12        end
13        i=i+1;
14    end
15    limpio=A;
16
17 end
```

### **distancia**

```
1 function x=distancia(x2,x1,y2,y1)
2     x=sqrt( ((x2-x1)^(2)) + ((y2-y1)^(2)) );
3 end
```

## division.m

```
1 function division(imagen ,im ,uno ,dos ,prueba)
2
3     i=1;
4
5     ejeX1=[];
6     ejeY1=[];
7
8     ejeX2=[];
9     ejeY2=[];
10
11    ejeX3=[];
12    ejeY3=[];
13
14    ejeX4=[];
15    ejeY4=[];
16
17    while i<length(uno)
18        if uno(i)<(size(imagen ,2)/2) && dos(i)<(size
(imagen ,1)/2)
19            ejeX1=[ejeX1 ,uno(i)];
20            ejeY1=[ejeY1 ,dos(i)];
21        elseif uno(i)>(size(imagen ,2)/2) && dos(i)<(
size(imagen ,1)/2)
22            ejeX2=[ejeX2 ,uno(i)];
23            ejeY2=[ejeY2 ,dos(i)];
24        elseif uno(i)<(size(imagen ,2)/2) && dos(i)>(
size(imagen ,1)/2)
25            ejeX3=[ejeX3 ,uno(i)];
26            ejeY3=[ejeY3 ,dos(i)];
27        else
28            ejeX4=[ejeX4 ,uno(i)];
29            ejeY4=[ejeY4 ,dos(i)];
30        end
31        i=i+1;
32    end
33
```

```

34     g=figure(4);
35     title("Coordenadas por cuadrante Acocil"+num2str
(im))
36     imagesc(imagen);hold on;plot(ejeX1,ejeY1,"r.",
ejeX2,ejeY2,"b.",ejeX3,ejeY3,"g.",ejeX4,ejeY4,"c
.");
37     axis ij
38     tipo=".png";
39     nombre="cuadro"+num2str(im)+tipo;
40     cd(prueba)
41     saveas(g,nombre)
42     cd ..
43
44
45     f=figure;
46     imagesc(imagen);hold on;
47     title("Posicion con mas puntos Acocil"+num2str(
im));
48
49     cuadrante=[length(ejeX1),length(ejeX2),length(
ejeX3),length(ejeX4)];
50
51     cuad=find(cuadrante==max(max((cuadrante)))));
52
53     if cuad==1
54         plot(ejeX1,ejeY1,"r.")
55         xlim([1 size(imagen,2)]),ylim([1 size(imagen
,1)])
56         axis ij
57     elseif cuad==2
58         plot(ejeX2,ejeY2,"b.")
59         xlim([1 size(imagen,2)]),ylim([1 size(imagen
,1)])
60         axis ij
61     elseif cuad==3
62         plot(ejeX3,ejeY3,"g.")
63         xlim([1 size(imagen,2)]),ylim([1 size(imagen
,1)])

```

```
64         axis ij
65     else
66         plot(ejeX4,ejeY4,"c.")
67         xlim([1 size(imagen,2)]),ylim([1 size(imagen
,1)])
68         axis ij
69     end
70
71     tipo=".png";
72     nombre="cuadrante"+num2str(im)+tipo;
73     cd(prueba)
74     saveas(f,nombre)
75     cd ..
76 end
```

### **ecualiza.m**

```
1 function B=ecualiza(A)
2 % A es la imagen de entrada
3 A=uint8(A); % A se pasa a arreglos sin signo de 8
   bits
4
5 h=imhist(A); %Histograma de la imagen A
6 h=h';% Se manda la matriz transpuesta de h
7
8 A=double(A); % Se convierte A a datos tipo doble
9
10 [m,n]=size(A);% Filas y columnas de A
11 H=h/(m*n);
12 pF=zeros(1,256);%Vector de ceros (1x256)
13
14 suma=0;
15 %Se suman todos los valores de la media de
   cuantizacion
16 for j=1:256
17     suma=suma+H(j);
18     pF(j)=suma;
19 end
20
21 B=255*pF(A+1);% Probabilidad de ocurrencia
```

### escala.m

```
1 function B=escala(A)
2
3 A=double(A);
4
5 r1=min(A(:));
6 r2=max(A(:));
7 b=r2;
8 [s1,s2]=size(A);
9
10
11
12 m=255/(r2-r1);
13
14 %B=uint8((((A-r1)*s2)/(r2-r1))+s1);
15 %Aqui es la transformacion lineal - recta
16
17 %nivel=a:b;% creacion vector
18
19 %g=(m*(nivel-b))+255;
20
21 B=(m*(A-b))+255;
```

## hipnograma.m

```
1 function tiempo=hipnograma(uno,dos,im,prueba)
2
3 %%% Esta funcion recibe el arreglo que contiene el
   hipnograma, la variable
4 %%% 'dos' es un hipnograma validado por lo que se
   debe borrar si no se
5 %%% realizo una validacion previamente
6
7
8 %%% Rellenado de ceros
9
10 uno=rellenarCeros(uno);
11
12 %%% Eliminacion de 1's aislados
13 i=0;
14 while i<2
15     uno=borrarUnos(unos);
16     i=i+1;
17 end
18
19 %%% Caso A y Caso B
20 uno=casoA(unos);
21 uno=casoB(unos);
22 uno=borrarUnos(unos);
23
24
25 i=0;
26 j=0;
27 contador=1;
28
29 while contador<=length(unos)
30     if unos(contador)==1
31         i=i+1;
32     end
33     if dos(contador)==1
34         j=j+1;
```

```

35     end
36
37     contador=contador+1;
38 end
39
40 muestra1=(i/3600);
41 muestra2=(j/3600);
42
43 horas=floor(muestra1);
44 minutos=(muestra1-horas)*60;
45 segundos=(minutos-floor(minutos))*60;
46 nombre="Tiempo total dormido: "+num2str(horas)+"h:"+
        num2str(floor(minutos))+"m:"+num2str(segundos)+"s
        ";
47 disp("Tiempo total dormido: "+num2str(horas)+"h:"+
        num2str(floor(minutos))+"m:"+num2str(segundos)+"s
        ")
48
49 horas=floor(muestra2);
50 minutos=(muestra2-horas)*60;
51 segundos=(minutos-floor(minutos))*60;
52 minutos=floor(minutos);
53 disp("Tiempo total dormido: "+num2str(horas)+"h:"+
        num2str(floor(minutos))+"m:"+num2str(segundos)+"s
        ")
54 nombre2="Tiempo total dormido: "+num2str(horas)+"h
        :"+num2str(floor(minutos))+"m:"+num2str(segundos)
        +"s";
55 t=1:1:length(unos);
56
57 g=figure;stairs(t,unos,"Linewidth",8);hold on;
58 set(gcf, 'Position', get(0, 'Screensize'));
59 stairs(t,dos,"Linewidth",4)
60 legend(nombre,nombre2);
61 title("hipnogramaAcocil "+num2str(im))
62
63 tipo=".png";
64 nombre="hipnogramaAcocil "+num2str(im)+tipo;

```

```
65 cd(prueba)
66 saveas(g,nombre)
67 cd ..
68
69 tiempo=uno;
70
71 %%%!C:\Users\redesNeuronales\Documents\MATLAB\videos
    \pruebaRHipnograma.r
72 %%%Revisar para e
73 end
```

## hipnograma2.m

```
1 function tiempo=hipnograma2(unos,im,prueba)
2
3 %%% Esta funcion recibe el arreglo que contiene el
   hipnograma, la variable
4 %%% 'dos' es un hipnograma validado por lo que se
   debe borrar si no se
5 %%% realizo una validacion previamente
6
7 i=3;
8
9 %%% Rellenado de ceros
10
11 unos=rellenarCeros(unos);
12
13 %%% Eliminacion de 1's aislados
14 i=0;
15 while i<2
16     unos=borrarUnos(unos);
17     i=i+1;
18 end
19
20 %%% Caso A y Caso B
21 unos=casoA(unos);
22 unos=casoB(unos);
23 unos=borrarUnos(unos);
24
25
26 i=0;
27 j=0;
28 contador=1;
29
30 while contador<=length(unos)
31     if unos(contador)==1
32         i=i+1;
33     end
34
```

```

35
36     contador=contador+1;
37 end
38
39 muestra1=(i/3600);
40
41
42 horas=floor(muestra1);
43 minutos=(muestra1-horas)*60;
44 segundos=(minutos-floor(minutos))*60;
45 minutos=floor(minutos);
46 disp("Tiempo total dormido: "+num2str(horas)+"h:"+
       num2str(floor(minutos))+"m:"+num2str(segundos)+"s
       ")
47
48
49
50 t=1:1:length(unos);
51 leyenda="Tiempo total dormido: "+num2str(horas)+"h
         :"+num2str(floor(minutos))+"m:"+num2str(segundos)
         +"s";
52 g=figure;plot(t,unos,"Linewidth",2,"DisplayName",
               leyenda);hold on;
53 grid on;
54 set(gcf, 'Position', get(0, 'Screensize'));
55
56
57 title("hipnogramaAcocil"+num2str(im))
58 legend("Tiempo total dormido: "+num2str(horas)+"h:"+
         num2str(floor(minutos))+"m:"+num2str(segundos)+"s
         ")
59 tipo=".png";
60 nombre="hipnogramaAcocil"+num2str(im)+tipo;
61 cd(prueba)
62 saveas(g,nombre)
63 cd ..
64 archivo=strcat("tablaAcocil",num2str(im),".txt");
65 T=table(transpose(t),unos);

```

```
66 cd(prueba)
67 writetable(T,archivo,'Delimiter','\t','WriteRowNames
    ',false,'WriteVariableNames',false);
68 cd ..
69
70
71 marcarTiempo(transpose(t),uno,prueba,im);
72 tiempo=uno;
73
74 end
```

### leer.m

```
1 clc; clear; close all;
2
3 A=VideoReader("20240611192703.MTS");
4 nombre='prueba13';
5 i=1;
6 j=1000;
7 ST='.jpg';
8 cuadros=A.NumFrames;
9 cuadros=cuadros/1;
10 cuadros=int16(cuadros);
11 mkdir imagenes13
12 n=1;
13 x=1;
14 corte=1;
15 mensaje='Lectura de imagen';
16 cd imagenes13
17 while n<cuadros
18     disp(mensaje);
19     disp(n);
20     parte=num2str(n);
21     crear=strcat(nombre,parte);
22     mkdir(crear);
23     vid=read(A,[i j]);
24     for x=1:20:size(vid,4)
25         Sx=num2str(corte);
26         frame=strcat(Sx,ST);
27         direc=vid(:,:,x);
28         cd(crear)
29         imwrite(direc,frame);
30         cd ..
31         corte=corte+1;
32     end
33     n=n+1;
34     i=i+1000;
35     j=j+1000;
36 end
```

```
37 cd ..
38 % cd(nombre)
39 % A=imread('900.jpg');
40 % cd ..
41 disp('Fin lectura de imagenes')
42 clear A;
```

### marcarTiempo.m

```
1 function marcarTiempo(tiempo,hipno,prueba,im)
2
3     i=1;
4     tipo=".txt";
5     archivo=strcat("/tiemposAcocil",num2str(im),tipo
6 );
7     muestra=fopen(strcat(prueba,archivo),"wt");
8     contador=1;
9     inicio=tiempo(1);
10    fin=0;
11    estado=hipno(1);
12    while i<length(hipno)
13        if(estado==hipno(i))
14            contador=contador+1;
15        else
16            fin=tiempo(i);
17            muestra1=(fin/5027);
18            horas=floor(muestra1);
19            minutos=(muestra1-horas)*60;
20            segundos=(minutos-floor(minutos))*60;
21            minutos=floor(minutos);
22            fprintf(muestra,"%d\t%d\t%d\t%d\t%s\n",
23                contador,estado,inicio,fin,strcat(num2str(horas)
24                +":"+num2str(floor(minutos))+":"+num2str(floor(
25                segundos),"%g"))));
26            inicio=fin;
27            estado=hipno(i);
28            contador=1;
29        end
30        i=i+1;
31    end
32    fclose(muestra);
33 end
```

## movimiento.m

```
1 function movimiento(unos,dos,im,prueba)
2     i=length(unos)/300;
3     i=uint16(i);
4     d=zeros(1,i);
5     contar=i;
6     inicio=1;
7     fin=300;
8     i=1;
9     j=1;
10    while i<contar
11        x=unos(inicio:fin);
12        y=dos(inicio:fin);
13        while j<300
14            valor=distancia(x(j+1),x(j),y(j+1),y(j))
15        ;
16            d(i)=valor+d(i);
17            j=j+1;
18        end
19        j=1;
20        i=i+1;
21        inicio=inicio+300;
22        fin=fin+300;
23        valor=0;
24    end
25    g=figure(5),stairs(d,"-b",LineWidth=1);
26    title("Movimiento/5min Acocil"+num2str(im))
27
28    tipo=".png";
29    nombre="movimiento"+num2str(im)+tipo;
30    cd(prueba)
31    saveas(g,nombre)
32    cd ..
33 end
```

### negativo.m

```
1 function [c,B]=negativo(A,u)
2     umbral=u;
3     referencia=imread("chest_x_ray.jpg");
4     %referencia=imread("91.jpg");
5     %referencia=imread("226.jpg");
6     pz=imhist(referencia);
7     c=histeq(uint8(escala(A)),pz);%Ecuacion
8     c=uint8(c);
9     ac=rgb2gray(c);
10    GB=ac<umbral;
11    binaria2=bwmorph(GB,'majority',50);
12    negativo=1-double(binaria2);
13    B=uint8(negativo);
14 end
```

### **rellenarCeros.m**

```
1 function limpio=rellenarCeros(uno)
2     i=3;
3     while i<(length(uno)-2)
4         if uno(i)==0
5             if uno(i-2)==1 && uno(i-1)==1
6                 if uno(i+2)==1 && uno(i+1)==1
7                     uno(i)=1;
8                 end
9             end
10        end
11        i=i+1;
12    end
13    limpio=uno;
14 end
```

### **separarVerde.m**

```
1 function [centro,radio]=separarVerde(A,rMin,rMax)
2     img=rgb2gray(A);
3     G=A(:,:,2);
4     imV=imsubtract(G,img);
5     binaria=imbinarize(imV,'global');
6     morfo=bwareaopen(binaria,5);
7     morfo=bwmorph(morfo,'majority',50);
8     [centro,radio]=imfindcircles(morfo,[rMin rMax], '
    Sensitivity',0.82);
9 end
```

### sleep.m

```
1 function sleep(unoV,xV,yV,uno,dos,m,n,im,prueba)
2     i=1;
3     muestra=length(unoV);
4     g=figure;
5     while i<muestra
6         plot(uno(i),dos(i),'b.','LineWidth',4),hold
on;
7         xlim([1 m]),ylim([1 n]);
8         axis ij
9         if unoV(i)==1
10            plot(xV(i),yV(i),'g.','MarkerSize',15),
hold on;
11            xlim([1 m]),ylim([1 n]);
12            axis ij
13            end
14            i=i+1;
15        end
16        title("Trayectoria/Suenioo"+num2str(im))
17
18        tipo=".png";
19        nombre="trayectoriaYsuenioo"+num2str(im)+tipo;
20        cd(prueba)
21        saveas(g,nombre)
22        cd ..
23 end
```

### **transf\_lineal.m**

```
1 function B=transf_lineal(A)
2
3 A=double(A);
4
5 a=min(A(:));
6 b=max(A(:));
7 %255=L-1
8 %(L-1)/(b-a) es la pendiente
9 B=((A-a)*255)/(b-a);
```

## trayectoria.m

```
1 function trayectoria(carpeta, buscarCarp, conteo, dimY1
    , dimY2, dimX1, dimX2, num, uno, dos, im, finConteo,
    prueba)
2
3     tipo = '.jpg';
4     ejeX = [];
5     ejeY = [];
6
7     ver = 1;
8     i = 1;
9     corte = 1;
10    while conteo < finConteo
11
12        while ver <= 50
13            if i < (size(uno, 1) - 10)
14                buscar = strcat(num2str(corte), tipo);
15                cd(buscarCarp)
16                cd(strcat(carpeta, num2str(conteo)))
17                imagen = imread(buscar);
18                cd ..
19                cd ..
20                figure(1);
21
22                mostrar = imagen(dimY1:dimY2, dimX1:
23                dimX2, :);
24
25                imagesc(mostrar), hold on
26                title("Acocil acuario"+num2str(im));
27                ejeX = uno(i:i+10);
28                disp(length(ejeX))
29                ejeY = dos(i:i+10);
30                plot(ejeX, ejeY, "c-", 'LineWidth', 4);
31            hold on;
32                xlim([1 size(mostrar, 2)]), ylim([1
33                size(mostrar, 1)])
34            axis ij
```

```
32
33         end
34         i=i+10;
35         ver=ver+10;
36         corte=corte+10;
37
38     end
39     ver=1;
40     conteo=conteo+1;
41 end
42 end
```



# Referencias

- Brown, E. A. (2018). 24 Endearing Photos of Animals Sleeping. *Natgeo*.
- Cherabit, N., Chelali, F. Z., & Djeradi, A. (2012). Circular hough transform for iris localization. *Science and Technology*, 2, pp: 114-121.
- Corral, C. (2020). ANY-maze Protocol (Elevated Plus Maze) v6. 2.
- Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1), 11-15.
- Escalante, B. (2006). Procesamiento Digital de Imágenes - Capítulo 4 - Realce de la Imagen [Recuperado el 14 de agosto del 2024 de [http://lapi.fi-p.unam.mx/wp-content/uploads/PDI\\_Cap4\\_Realce-de-la-Imagen.pdf](http://lapi.fi-p.unam.mx/wp-content/uploads/PDI_Cap4_Realce-de-la-Imagen.pdf)].
- Gavaldón, R. (1959). Macario - Galería -Filmoteca UNAM [Recuperado el 14 de agosto del 2024.]. [https://www.filmoteca.unam.mx/ciclos\\_en\\_linea/macario-memoria-y-homenaje-universitario-a-nuestros-fallecidos/macario-galeria/](https://www.filmoteca.unam.mx/ciclos_en_linea/macario-memoria-y-homenaje-universitario-a-nuestros-fallecidos/macario-galeria/)
- Gonzales, R. C., & Woods, R. E. (2018). *Digital Image Processing*. Pearson. Cuarta Edición.
- Helfrich-Förster, C. (2018). Sleep in insects. *Annual review of entomology*, 63(1), 69-86.
- Hendricks, J. C., Finn, S. M., Panckeri, K. A., Chavkin, J., Williams, J. A., Sehgal, A., & Pack, A. I. (2000). Rest in *Drosophila* is a sleep-like state. *Neuron*, 25(1), 129-138.
- Horner, A., Schmidt, M., Edwards, D., & Derby, C. (2008). Role of the olfactory pathway in agonistic behavior of crayfish, *Procambarus clarkii*. *Invertebrate neuroscience : IN*, 8, 11-8. <https://doi.org/10.1007/s10158-007-0063-1>
- Hough, P. V. (1962). Method and means for recognizing complex patterns [US Patent 3,069,654].

- Jiménez-Morales, N., Mendoza-Ángeles, K., Porras-Villalobos, M., Ibarra-Coronado, E., Roldán-Roldán, G., & Hernández-Falcón, J. (2018). Who is the boss? Individual recognition memory and social hierarchy formation in crayfish. *Neurobiology of learning and memory*, *147*, 79-89.
- Liang, Q., Liu, D., Zhang, D., Wang, X., Zhu, B., & Wang, F. (2024). Machine learning-based aggressiveness assessment model construction for crabs: A case study of swimming crab *Portunus trituberculatus*. *Aquaculture*, *593*, 741304.
- MathWorks. (2024). Image Processing Toolbox [Recuperado el 16 de agosto del 2024 de [https://la.mathworks.com/help/images/index.html?s\\_tid=hc\\_product\\_card](https://la.mathworks.com/help/images/index.html?s_tid=hc_product_card)].
- Mendoza-Ángeles, K., Hernández-Falcón, J., Cabrera, A., & Ramón, F. (2005). Crayfish brain states characterization with wavelet transform. *2005 2nd International Conference on Electrical and Electronics Engineering*, 160-163.
- Miyazaki, S., Liu, C.-Y., & Hayashi, Y. (2017). Sleep in vertebrate and invertebrate animals, and insights into the function and evolution of sleep. *Neuroscience research*, *118*, 3-12.
- Nath, T., Mathis, A., Chen, A. C., Patel, A., Bethge, M., & Mathis, M. W. (2019). Using DeepLabCut for 3D markerless pose estimation across species and behaviors. *Nature protocols*, *14*(7), 2152-2176.
- Nixon, M. S., & Aguado, A. S. (2020). 5—High-level feature extraction: Fixed shape matching. *Feature Extraction and Image Processing for Computer Vision*, 223-290.
- Osorio-Palacios, M., Montiel-Trejo, L., Oliver-Dominguez, I., Hernández-Falcón, J., & Mendoza-Ángeles, K. (2021). Sleep phases in crayfish: relationship between brain electrical activity and autonomic variables. *Frontiers in Neuroscience*, *15*, 694924.
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), pp: 62-66. <https://doi.org/10.1109/TSMC.1979.4310076>
- Patullo, B., Jolley-Rogers, G., & Macmillan, D. (2007). Video tracking in the extreme: Video analysis for nocturnal underwater animal movement. *Behavior research methods*, *39*(4), 783-788.
- Pereira, T. D., Tabris, N., Matsliah, A., Turner, D. M., Li, J., Ravindranath, S., Papadoyannis, E. S., Normand, E., Deutsch, D. S., Wang, Z. Y., et al., (2022). SLEAP: A deep learning system for multi-animal pose tracking. *Nature methods*, *19*(4), 486-495.
- Piéron, H. (1913). *Le problème physiologique du sommeil*. E. Grevin.

- Plum, F., Bulla, R., Beck, H. K., Imirzian, N., & Labonte, D. (2023). replicAnt: a pipeline for generating annotated images of animals in complex environments using Unreal Engine. *Nature Communications*, *14*(1), 7195.
- Pophale, A., Shimizu, K., Mano, T., Iglesias, T. L., Martin, K., Hiroi, M., Asada, K., Andaluz, P. G., Van Dinh, T. T., Meshulam, L., et al., (2023). Wake-like skin patterning and neural activity during octopus sleep. *Nature*, *619*(7968), 129-134.
- Ramón, F., Hernández-Falcón, J., Nguyen, B., & Bullock, T. H. (2004). Slow wave sleep in crayfish. *Proceedings of the National Academy of Sciences*, *101*(32), 11857-11861.
- Rhody, H. (2005). Lecture 10: Hough circle transform. *Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology*.
- Suryanto, M. E., Luong, C. T., Vasquez, R. D., Roldan, M. J. M., Hung, C.-H., Ger, T.-R., & Hsiao, C.-D. (2023). Using crayfish behavior assay as a simple and sensitive model to evaluate potential adverse effects of water pollution: Emphasis on antidepressants. *Ecotoxicology and Environmental Safety*, *265*, 115507.
- Systems, L. (2021). Tracked crayfish in shuttle tank using LoliTrack 4 [Recuperado el 17 de octubre del 2024.]. [https://www.youtube.com/watch?v=mIYOlb\\_XjX4](https://www.youtube.com/watch?v=mIYOlb_XjX4)