



### 3.1 Introducción

En este capítulo se describe el procesamiento que se lleva a cabo sobre las imágenes de una misma secuencia tomadas en una jaula marina para la detección de ojos.

El cometido final de los algoritmos de procesamiento de imagen es la detección de los ojos de varios peces por imagen en ambas imágenes del sistema estéreo. Se trata de algoritmos de segmentación orientados a regiones debido a la imposibilidad de obtener buenos resultados al intentar segmentar el pez en su contorno completo. La detección de colas y la reconstrucción 3D de los puntos característicos ya se encuentran implantados, y por lo tanto no serán objeto de estudio en este proyecto.

Se procesarán secuencias de imágenes procedentes de distintas piscifactorías y con condiciones de contorno distintas, con el fin de dotar a la aplicación de la máxima robustez posible.

Al ser este un proyecto de investigación, durante la fase de desarrollo se han probado multitud de técnicas distintas para resolver los problemas que surgían. Se explicará con detalle cada una de las técnicas empleadas para resolver el problema de la detección de ojos, obviando las técnicas

desestimadas por no ofrecer buenos resultados. El contenido del capítulo se dividirá en tres apartados fundamentales:

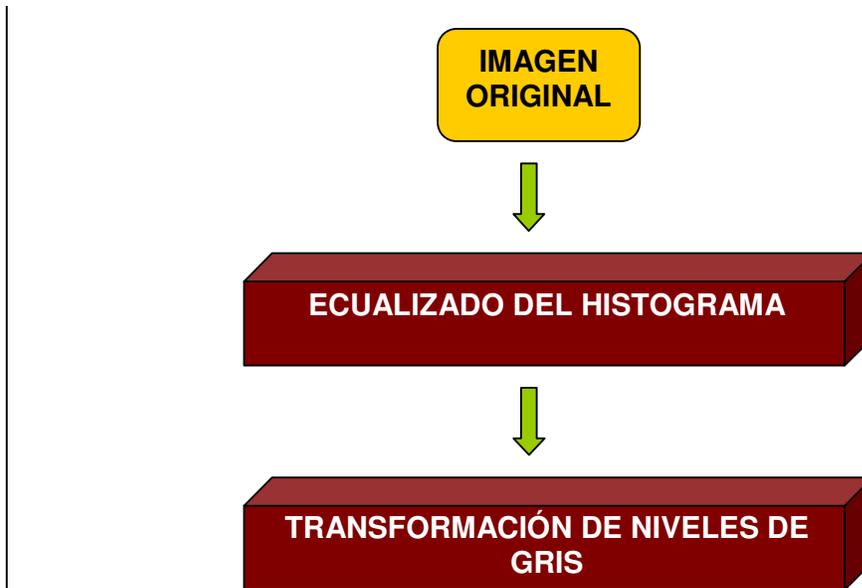


*Fig. 3.1 Contenido del capítulo*

## 3.2 Preprocesamiento

### 3.2.1 Introducción

Dentro de un esquema de visión artificial, la finalidad del preprocesamiento de la imagen es la obtención de otra imagen de mayor calidad que haga que las etapas posteriores sean más simples y fáciles. No se va a realizar extracción de información de la imagen, sino se actuará directamente sobre los niveles de grises correspondientes a la imagen de cada una de las dos bandas, para compensar determinados defectos presentes causados por las condiciones de captura. Los principales defectos son los provocados por una iluminación no adecuada o desigual, esto puede dar lugar a falta de contraste, no uniformidad, etc. También se intentará eliminar el ruido presente ocasionado por el sistema de digitalización, así como cualquier otro efecto espúreo. Aunque a lo largo de la duración del proyecto se han aplicado innumerables técnicas de preprocesamiento en la aplicación final se optó por las que aparecen la figura 3.2.



*Fig. 3.2 Diagrama de los métodos de preprocesado de la aplicación*

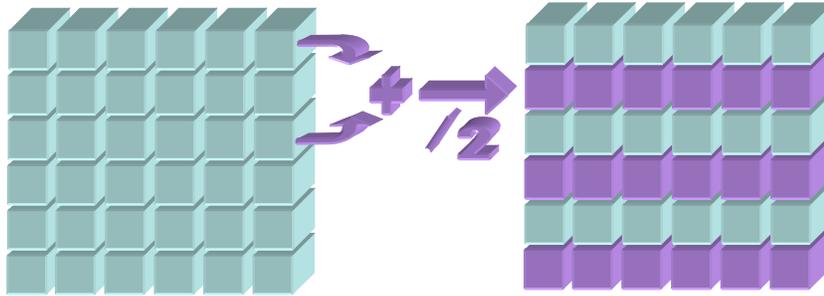
### 3.2.2 Eliminación de entrelazado

Este paso sólo se lleva a cabo si las imágenes que se desea procesar con la aplicación son entrelazadas. Aunque con las cámaras que se utilizan en la actualidad no se obtienen imágenes entrelazadas, se introdujo este paso porque las primeras imágenes de labinas que se procesaron sí lo eran.

Como efecto de la forma en la que el sensor de la cámara realiza la adquisición se produce una pérdida de definición en los bordes de los objetos. Esto se ocasiona debido a que la imagen está compuesta por dos campos, el par y el impar, correspondientes a las líneas pares e impares de la imagen. Dichos campos son captados por el sensor de la cámara en instantes diferentes. Lógicamente, para objetos en movimiento este efecto se aprecia con mayor intensidad. Existen dos posibilidades, por un lado reducir la resolución de la imagen en filas a la mitad o construir una nueva imagen en el que se sustituye el campo eliminado a partir del otro.

Se opta por la segunda posibilidad para no modificar la relación de aspecto de la imagen. Para montar el campo eliminado, se hace una interpolación entre líneas adyacentes. El procedimiento es muy simple, se

recorre píxel a píxel las filas del campo a eliminar y se sustituye cada píxel por la media aritmética hecha entre los píxeles vecinos superior e inferior.



*Fig. 3.3 Eliminación del entrelazado por interpolación de líneas.*

Este proceso debe ser aplicado a ambas imágenes del estéreo. La elección del campo par o impar es indiferente pero en ambas imágenes debe ser elegido el mismo. De otro modo, existiría un desfase entre instantes de captura de ambas imágenes, lo cual perjudicaría para la realización de correspondencias y posterior reconstrucción 3D.



*Fig. 3.4 Imagen original tomada por una de las cámaras del estéreo sin eliminación del entrelazado.*



*Fig. 3.5 Imagen capturada por una de las cámaras del estéreo después de haberse aplicado la eliminación del entrelazado*

### **3.2.3 Ecuación del histograma**

Para la ecuación del histograma de las imágenes se puede optar por alguna de las dos alternativas que se ofrecen:

- Ecuación básica
- Ecuación mejorada

Se empleará aquella que sea más adecuada para cada tipo de imagen, como se verá en el Capítulo 5.

A continuación se describe cada una de ellas con detalle.

### 3.2.3.1 Ecuación básica

El histograma es un diagrama de barras cuyas abscisas representan los niveles de gris de una imagen, y las ordenadas, las frecuencias relativas de los distintos niveles de gris.

$$P_k = \frac{n_k}{n} \quad (\text{Ec. 3.1})$$

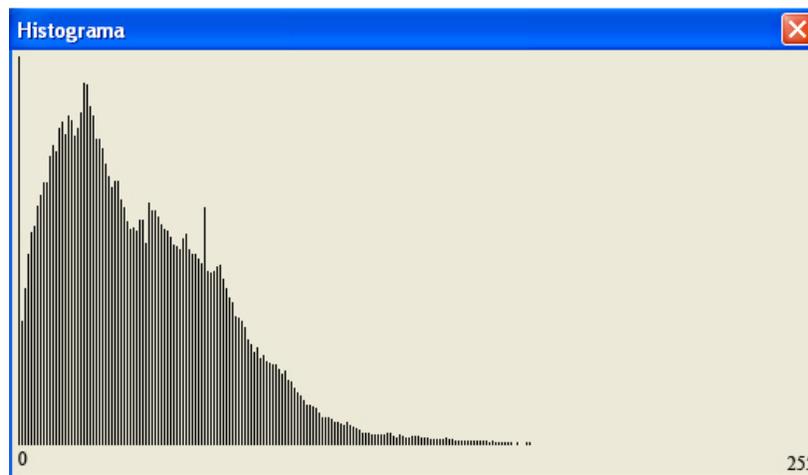


Fig. 3.6 Histograma de la imagen sin ecualizar, se concentra en una zona



Fig. 3.7 Imagen de partida sin ecualizar

En la figura 3.6 se observa un histograma estrecho, de bajo rango dinámico, en concordancia con el poco contraste visual de la imagen correspondiente, figura 3.7. El objetivo de la ecualización del histograma es ampliar el rango dinámico de los niveles de gris, para acentuar el contraste visual de la imagen.

El histograma es una función de dominio discreto, pero su ecualización está inspirada en una operación sobre funciones de dominio continuo, cuyo significado se explica brevemente. Supongamos que  $s$  es una variable aleatoria con valores en el intervalo continuo  $[0, 1]$ , con función de densidad  $f(s)$  y función de distribución  $F(s)$ . Si queremos definir una nueva variable aleatoria  $r$  que sea función de  $s$  y que tenga distribución uniforme en  $[0, 1]$ , se debe cumplir que:

$$r = F(s) = \int_0^s f(t)dt \quad (\text{Ec. 3.2})$$

Se ha obtenido la solución para el caso continuo. Para el caso discreto, se discretiza la integral, dividiendo  $[0, 1]$  en 256 partes iguales, para obtener una solución correspondiente a una escala de 256 niveles de gris. Queda entonces:

$$P_i = f\left(\frac{i}{256}\right) \cdot \frac{1}{256} \quad \text{con } 0 \leq i \leq 255 \quad (\text{Ec. 3.3})$$

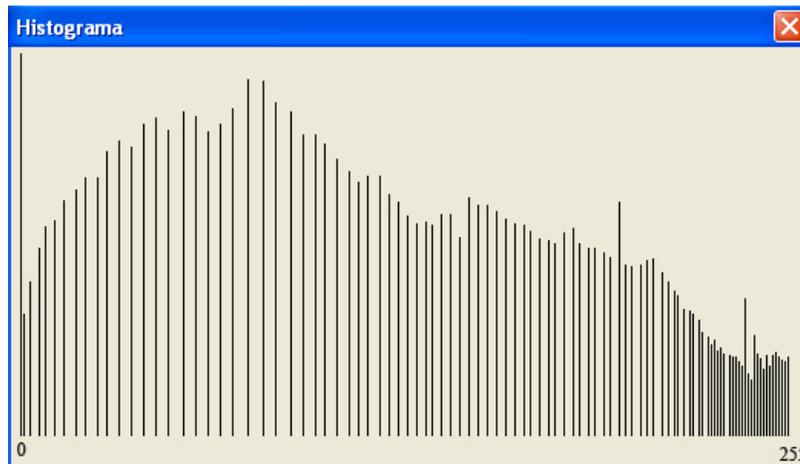
como en la práctica trabajamos con valores entre 0 y 255 tomaremos

$$F(k) = 255 \sum_{i=0}^k P_i \quad \text{con } 0 \leq k \leq 255 \quad (\text{Ec. 3.4})$$

La transformación  $F$  define una operación puntual que a cada píxel de nivel de gris  $k$ , lo transforma en un píxel de nivel de gris  $F(k)$ , logrando una ampliación del rango dinámico de la imagen dada. Mediante la utilización de la siguiente aplicación podemos distribuir los niveles de gris en todo el rango dinámico.

$$G(k) = 255 \frac{F(k) - F(0)}{255 - F(0)} \quad (\text{Ec. 3.5})$$

La imagen que se muestra a continuación resulta de aplicar el proceso de ecualización a la imagen de la figura 3.7. Puede observarse la ampliación del rango dinámico en el histograma, así como el aumento del contraste visual en la imagen.



*Fig. 3.8 Histograma igualado, se expanden los niveles de grises aprovechando todo el rango.*

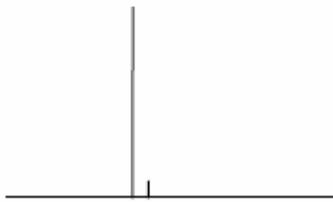


*Fig. 3.9 Imagen tras la ecualización*

La igualación de histograma se adapta perfectamente al procedimiento automático que se busca en la aplicación ya que la función de transformación viene únicamente determinada por el histograma de la imagen de entrada. Otra característica es el aprovechamiento automático de toda la escala de grises. Esta técnica se aplica para lograr un mayor contraste y poder detectar detalles de los peces como es el caso de los ojos.

### 3.2.3.2 Ecuación mejorada

En el proceso descrito anteriormente, que es la técnica estándar mencionada en la bibliografía, podemos perder información de pequeños detalles, si se da la situación de la Figura 3.10:



*Fig. 3.10 Detalle histograma de imagen*

En ella las barras del diagrama representan una porción del histograma de una imagen, en la que un objeto, cuyo nivel de gris corresponde a la barra más pequeña, se halla inmerso en una zona, en la que el nivel de gris, más oscuro, es el de la barra de mayor longitud. La separación de dos niveles de gris consecutivos, luego de la ecualización, depende de la cantidad de pixels asociada al nivel más brillante, según se desprende de las fórmulas dadas antes. En el caso presente, ello provocaría que el objeto fuera prácticamente indistinguible de su fondo en la imagen ecualizada. Este problema puede evitarse si se ecualiza el negativo de la imagen. El negativo se obtiene reemplazando el nivel gris  $k$  por  $255-k$  en cada píxel de la imagen.

En síntesis, tomando el negativo de la imagen, ecualizando y volviendo a aplicar el negativo, el objeto contrastaría claramente con su fondo. Sin embargo, es fácil imaginar una situación en la que resulte más eficiente la

ecualización directa de la imagen. Además, ambas situaciones pueden presentarse en la misma imagen, con lo cual los dos métodos fallarían. Una solución, tal como se especifica en el siguiente diagrama, es ecualizar de las dos formas y promediar las imágenes obtenidas.

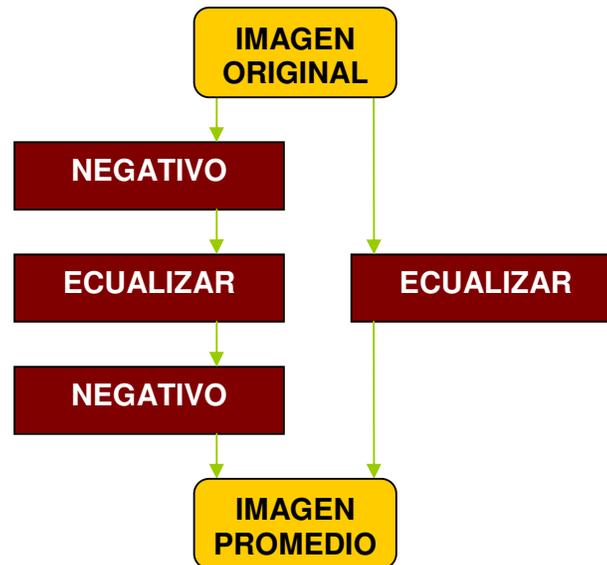


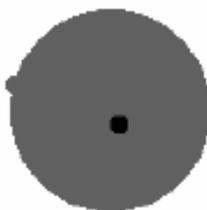
Fig. 3.11 Ecualización mejorada

El promedio de las imágenes se calcula promediando, para cada píxel, los niveles de gris de ambas imágenes.

Para ilustrar este proceso se ha generado una imagen sintética, de bajo rango dinámico, que está compuesta por dos zonas de gran cantidad de píxels y 3 pequeños círculos que involucran pocos píxels.



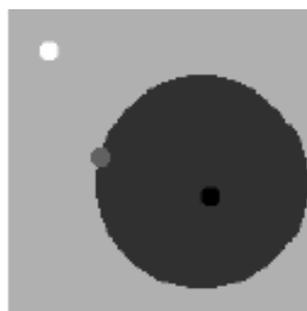
Fig. 3.12 Imagen generada para probar el ecualizado optimizado



*Fig. 3.13 Imagen generada ecualizada*



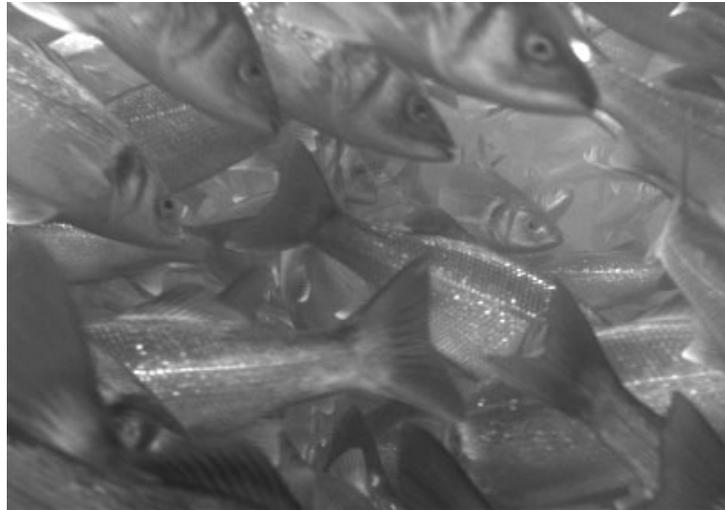
*Fig. 3.14 Negativo de la imagen negativa ecualizada*



*Fig. 3.15 Promedio*

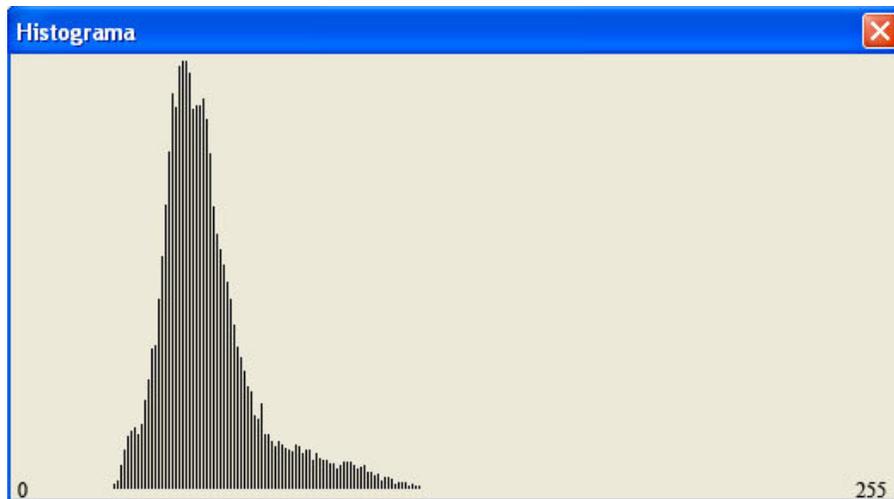
Obsérvese como se distinguen los tres pequeños círculos en la figura 3.15. Esto no ocurre ni con la ecualización directa (figura 3.13), ni con la ecualización del negativo (figura 3.14).

Una vez expuesto los principios teóricos se pasa a ver situaciones como las descritas en la Figura 3.10 en las imágenes de lubinas.



*Fig. 3.16 Detalle imagen lubinas*

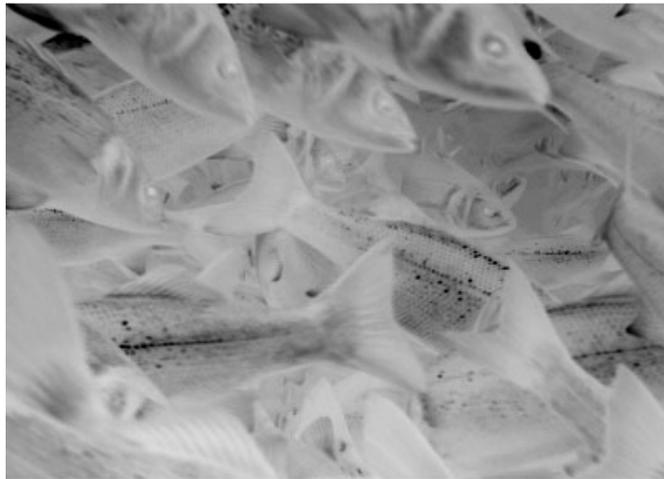
En la Figura 3.16 se encuentra una situación de bajo rango dinámico, como se puede observar en su histograma representado en la Figura\_3.17. Este bajo rango dinámico hará que al ecualizar el histograma se pierda información de pequeños detalles como el límite del círculo blanco que rodea a la pupila.



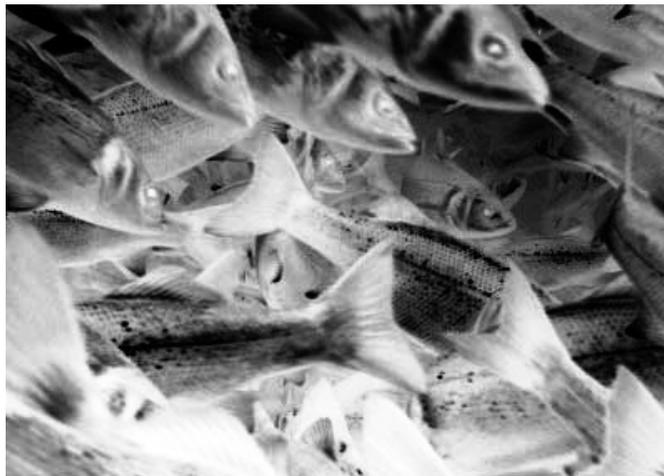
*Fig. 3.17 Histograma imagen de bajo rango dinámico*



*Fig. 3.18 Imagen generada ecualizada*



*Fig. 3.19 Negativo de la imagen*



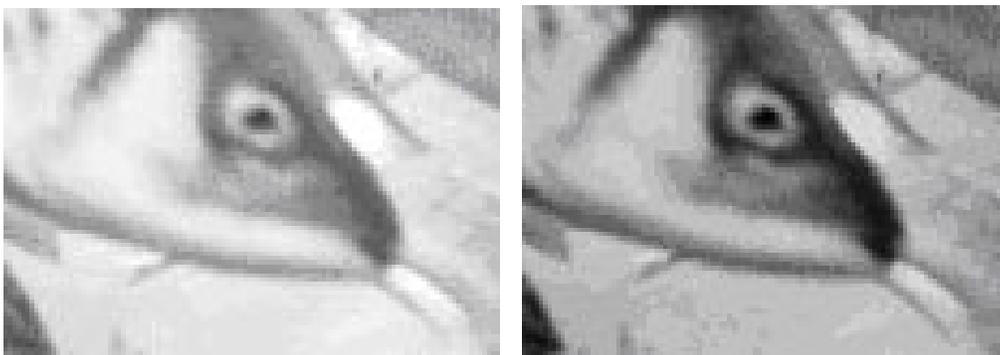
*Fig. 3.20 Imagen negativa ecualizada*



*Fig. 3.21 Negativo de la imagen negativa ecualizada*



*Fig. 3.22 Promedio*



*Fig. 3.23 Detalle del ojo en las ecualizaciones. A la izquierda ecualizado básico y derecha ecualizado mejorado*

Si se compara la imagen ecualizada con la ecualizada mejorada, se observa que en la última se acentúa más rasgos que apenas eran distinguibles en la primera. Más en concreto, lo que interesa es acentuar el límite de la zona blanca del ojo para facilitar la detección, cosa que se consigue de esta forma como se observa en la figura 3.23, donde a la izquierda está el detalle del ojo en la imagen con ecualizado básico y a la derecha con el ecualizado mejorado.

### 3.2.4 Transformación de niveles de gris

Tras utilizar los métodos expuestos se ha conseguido aumentar el contraste de toda la imagen por igual. Como el fin de la aplicación es la detección de ojos de peces en imágenes, lo que verdaderamente va a interesar no es aumentar el contraste de toda la imagen, sino aumentarlo en las zonas donde se encuentran los objetos a segmentar, es decir, en el rango de niveles de grises donde existe más posibilidad de encontrar un ojo.

Si además de aumentar el contraste en la zona de interés, lo reducimos en el resto, todo lo que esté fuera del rango elegido pasará a formar parte del fondo de la imagen.

Para aplicar esta técnica, se va a transformar el rango original de la imagen  $[0,255]$  al rango  $[0,1]$ .

La transformación que van a sufrir cada nivel de gris la podemos ver de forma gráfica en la figura 3.24, donde se ha representado todos los parámetros que tendremos que ajustar en función de la tonalidad de la zona a segmentar.

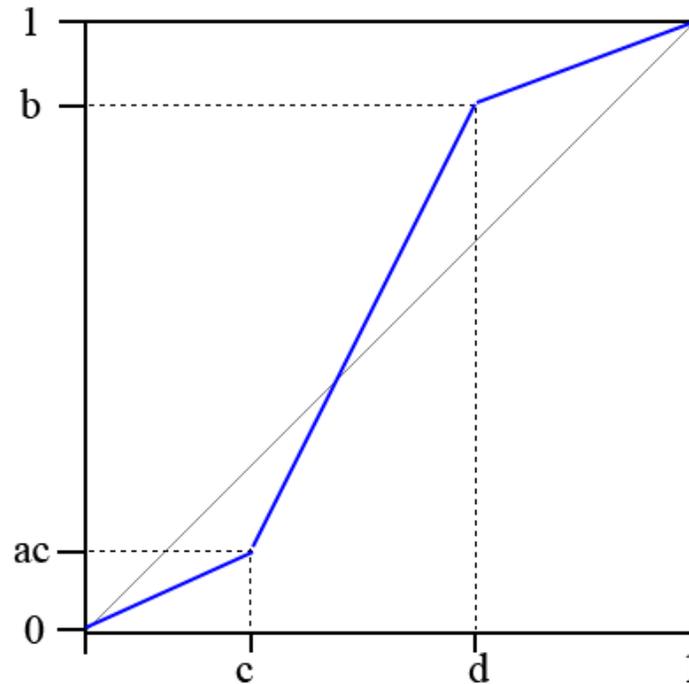


Fig. 3.24 Transformación de niveles de gris

En la gráfica se distinguen tres zonas con pendiente distinta:

- Tramo  $[0,c]$
- Tramo  $[c,d]$
- Tramo  $[d,1]$

En el primer tramo la pendiente es  $a$ , siendo  $a < 1$ , por lo que el rango  $[0,c]$  de entrada se reduce al rango  $[0,ac]$  de salida, es decir, se pierde contraste en dicha zona (zonas más oscuras).

En el tramo  $[c,d]$  la pendiente es mayor que uno, por lo tanto los valores del rango  $[c,d]$  se mapean al rango  $[ac,b]$ , siendo el segundo mayor que el primero. Este tramo va a definir la zona de interés.

En el tramo  $[d,1]$  la pendiente es de nuevo menor que uno, por lo tanto los valores del rango  $[d,1]$  se mapean al rango  $[b,1]$ , siendo el segundo menor que el primero. Se está reduciendo el contraste, pero esta vez en las zonas más claras.

Veamos ahora las expresiones matemáticas que habría que emplear en cada rango para llevar a cabo la transformación:

$$y = T(x) = \begin{cases} a \cdot x & \text{si } x \in [0, c] \\ a \cdot c + (x - c) \frac{b - ac}{d - c} & \text{si } x \in [c, d] \\ b + (x - d) \frac{1 - b}{1 - d} & \text{si } x \in [d, 1] \end{cases} \quad (\text{Ec. 3.6})$$

Con esta función se va a aumentar el contraste de la imagen en un rango de niveles de grises a consta de reducirlo en el resto consiguiendo mejorar la imagen, en cuanto a contraste se refiere, en las zonas de interés.

Tras el mapeado los objetos de interés tendrán un contraste más alto, y los menos interesantes pasarán a formar parte del fondo de imagen, con lo cual se facilita la detección de los ojos y se reduce la carga computacional al reducir el número de objetos de la imagen que se van a procesar.

En la Figura 3.27 se puede ver el resultado de aplicar la transformación de niveles de gris a una imagen de lubinas (Figura 3.25).



*Fig. 3.25 Imagen ecualizada de entrada a la función*



*Fig. 3.26 Detalle imagen ecualizada de entrada a la función*



*Fig. 3.27 Imagen tras realizar la transformación de niveles de gris*



*Fig. 3.28 Detalle imagen tras realizar la transformación de niveles de gris*

Se observa que se ha conseguido el resultado deseado, se ha aumentado el contraste en la zona de interés, es decir, en el ojo, a consta de reducirlo en el resto. Este aumento de contraste favorecerá a la correcta umbralización en las zonas de interés.

### **3.3 Umbralización por modelo de iluminación**

#### **3.3.1 Introducción**

En visión artificial es de gran importancia tener las imágenes con una iluminación adecuada y homogénea. En el caso de estudio de este proyecto no existe una iluminación uniforme, por lo que habrá que compensarla de alguna forma.

En las imágenes tomadas en estanques exteriores de tipo de jaula, para los que esta desarrollado el módulo de procesamiento implantado en la aplicación, se observan efectos de iluminación desigual debido a la iluminación natural del sol. En estas imágenes la iluminación viene desde arriba, luego la parte superior de la imagen aparece muy saturada y la parte inferior con escasa iluminación. Debido a la textura reflectante de la piel de los peces es aún más perjudicial para la segmentación. Aparecen los lomos de los peces muy brillantes y oscuras las partes inferiores.

Durante el desarrollo del proyecto se ha probado distintas técnicas para corregir la iluminación, decantándose al final por la generación de un modelo de iluminación de la jaula marina durante el tiempo de captura de una secuencia de imágenes, y umbralizar a partir de dicho modelo.



*Fig. 3.29 Ejemplo de iluminación desigual acusada*

### 3.3.2 Modelo de iluminación

Como se ha comentado en la introducción, se tendrá que generar un modelo de la iluminación de la jaula durante el tiempo de captura de la secuencia para poder compensar el efecto dañino de una iluminación desigual e inadecuada en las imágenes que se van a procesar.

El método empleado para generar el modelo parte de tres principios, que han de cumplirse para que el modelo sea fiel a la realidad:

- Los peces que se capturan en las imágenes de la secuencia se encuentran constantemente en movimiento
- El tiempo de captura de cada secuencia de imágenes tiene que ser suficientemente pequeño para considerar que la iluminación permanece constante durante ese tiempo
- Las cámaras tienen que permanecer en la misma profundidad y con el mismo ángulo de elevación durante el tiempo de captura de la secuencia

La solución propuesta es realizar la media aritmética píxel a píxel de todas las imágenes que componen la secuencia.

Al encontrarse los peces en constante movimiento, si la secuencia es suficientemente grande, tras realizar la media no se observará ningún pez, solamente se observará como penetra la iluminación del sol en el agua durante el tiempo de captura. Esta afirmación se basa en que en una posición determinada en la mayoría de las capturas se observa el fondo de imagen, no peces, ya que estos se encuentran en movimiento.

Hay que apuntar que cuanto menor sea la concentración de peces, el modelo de iluminación será más fiel a la realidad. Esto se debe al aumento con la concentración de peces de los reflejos no deseados del sol en las escamas.

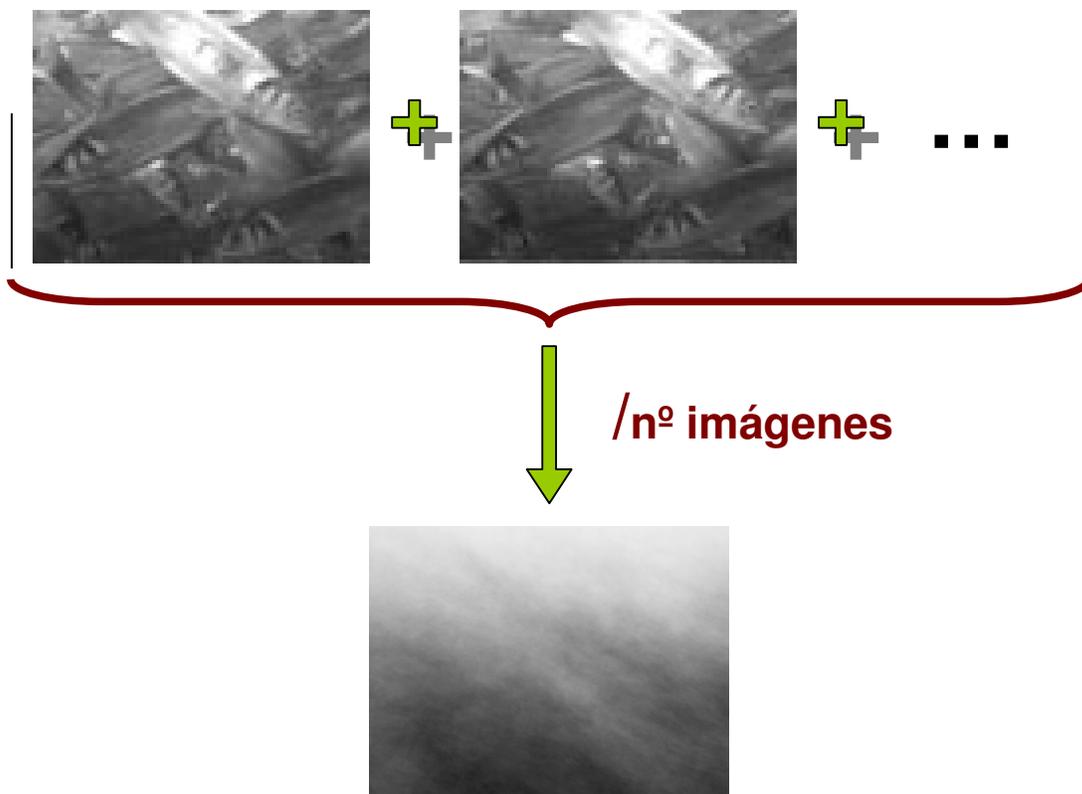


Fig. 3.30 Generación de modelo de iluminación

### 3.3.3 Umbralización

Es un método para convertir una imagen de niveles de gris en una imagen binarizada de forma que los píxeles pertenecientes a un mismo objeto tengan el mismo valor. Una imagen binarizada es aquella en la que los píxeles pueden tener dos únicos valores. La aplicabilidad como segmentador es cuando los objetos están sobre un fondo contrastado. La umbralización puede verse como una comparación con una función de la forma:

$$T = T[x, y, p(x, y), f(x, y)], \quad (\text{Ec. 3.7})$$

donde  $f(x, y)$  es el nivel de gris del punto  $(x, y)$ , y  $p(x, y)$  es alguna propiedad local de punto.

La imagen umbralizada  $g(x, y)$  se obtiene como:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases} \quad (\text{Ec. 3.8})$$

Un píxel de  $g(x, y)$  cuyo valor sea "1" (255 ó 0) corresponderá al objeto, mientras que los de valor "0" (0 ó 255) corresponderán al fondo. Cuando  $T$  depende solamente de  $f(x, y)$ , el umbral se denomina global. Si por el contrario,  $T$  depende de  $f(x, y)$  y  $p(x, y)$ , el umbral se llama local. Si además  $T$  depende de las coordenadas  $x$  e  $y$  se llama umbral dinámico. El método implantado en la aplicación para la binarización de la imagen es el de **umbralización local dinámica**.

Los umbrales locales se usan cuando no existe una clara definición entre los objetos y el fondo, y cuando la iluminación no es uniforme. Este es el caso de la detección de ojos en las jaulas.

Para binarizar cada una de las imágenes de la secuencia nos vamos a basar en el modelo de iluminación para obtener el umbral, es decir,  $p(x, y)$  será el modelo de iluminación. Tendremos por tanto un umbral local y dinámico para cada píxel, que se calculará aplicando un porcentaje al píxel situado en esa misma posición en el modelo de iluminación:

$$T(x, y) = \text{porcentaje} \cdot p(x, y) \quad (\text{Ec. 3.9})$$



Fig. 3.31 Imagen binaria

## 3.4 Detección de ojos

### 3.4.1 Introducción

Como se vio en el Capítulo 2, el punto elegido de las lubinas para su detección es el ojo. En este apartado se desarrolla el método empleado para detectarlo.

En primer lugar se justifica el por qué del método empleado, exponiendo las conclusiones sacadas tras la observación de la morfología de estos peces. En apartados posteriores se explica cada uno de los pasos para lograr la detección.

### 3.4.2 Justificación del método

En la segmentación se han buscado puntos característicos basados en la morfología de los peces y que tras largo proceso de investigación se han considerado los más fáciles de localizar dentro de la imagen de un estanque exterior en vez de buscar una detección del contorno completo del pez. Tras un estudio minucioso observando distintas imágenes de lubinas se extrajeron una

serie de características de los ojos, que los hacía distinguibles del resto de blobs de la imagen.

Todas estas características se van a explicar a partir de la Figura 3.32.



*Fig. 3.32 Detalle del ojo de una lubina*

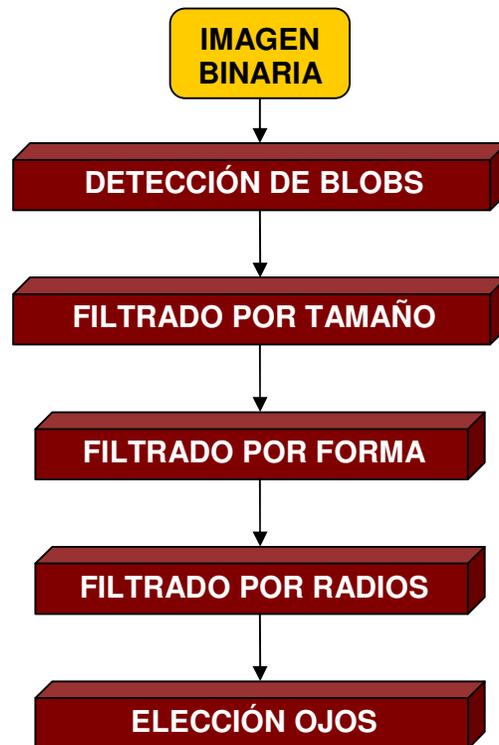
A simple vista llama la atención el fuerte contraste que existe entre la pupila del ojo, objeto muy oscuro, y el fondo sobre el que se encuentra, que es de una tonalidad muy clara. Este factor fue determinante a la hora de elegir el ojo como punto a detectar, ya que este fuerte contraste facilitará su detección incluso en imágenes con mala iluminación.

El objeto ojo lo podemos descomponer en dos objetos, un círculo de tonalidad muy clara, sobre el que se superpone otro círculo de área menor, de tonalidad muy oscura y situado en su centro. Este segundo objeto, la pupila, está perfectamente delimitado, y goza de cierta redondez.

Además de lo dicho, otra característica que será crucial para la detección de los ojos es que la separación entre la pupila y el borde del círculo de tonalidad clara es aproximadamente igual al diámetro de la pupila.

La solución propuesta se basa en la búsqueda en imágenes binarias de blobs que cumplan estas características.

En la Figura 3.33 se puede ver el diagrama funcional del método, que se basa en distintas características que deben cumplir los blobs para ser candidatos a ojos.



*Fig. 3.33 Diagrama funcional detección de ojos*

### 3.4.3 Detección de blobs

El primer paso en la detección de ojos es la búsqueda e identificación de blobs.

Llegados a este punto resulta conveniente definir el concepto de blob. Un blob es una región conectada de píxels, es decir, es un objeto. El análisis de blobs permitirá saber las características de los objetos de interés, y eliminar objetos no deseados.

Cada blob detectado en la imagen quedará localizado almacenando la posición de su centro de gravedad. Se dispondrá en todo momento de una tabla con la posición de cada blob detectado en la imagen, que será utilizada en los filtrados posteriores. Este centro de gravedad se calcula con la librería para el tratamiento de blobs desarrollada en el GRVC.

### 3.4.4 Filtrado por tamaño

El primer paso es la discriminación de blobs en función del área que tengan. Esta discriminación reducirá la carga computacional para procesos posteriores, ya que el número de blobs a analizar se reducirá drásticamente.

Podemos emplear esta discriminación porque la pupila es un objeto aislado cuya área dependerá de la distancia a la que se encuentre la lubina de la cámara. Además el área de la pupila no sufre grandes oscilaciones con el peso del pez.

Tendremos que establecer un área máxima y mínima de blob en función del área que tengan las pupilas de los peces que estén a la distancia de interés. Mediante métodos experimentales, que se expondrán en el Capítulo 5, se determinan estas áreas.

Los blobs que entren en el rango de área deseada pasarán a formar parte de los candidatos a ojos, y serán considerados como tal si cumplen el resto de características. Para identificar cada uno de los candidatos almacenamos la posición del centro de gravedad, ya que este será utilizado por el resto de procesos.

A continuación se muestra los distintos resultados que se pueden dar tras aplicar la discriminación de blobs por tamaño en imágenes de lubinas.



*Fig. 3.34 Detalle de candidatos no detectados*



*Fig. 3.35 Detalle de candidatos no detectados*

En la Figura 3.34 se puede observar dos candidatos no detectados. El primero de ellos (imagen de la izquierda), no se han considerado porque al estar en planos posteriores el tamaño de la pupila es demasiado pequeño como para entrar en el rango deseado. En la imagen de la derecha lo que ocurre es que la pupila no se encuentra totalmente aislada, y por lo tanto no es un blob.

Con el pez situado en la parte inferior de la Figura 3.35 ocurre lo mismo que en la imagen de la izquierda de la figura anterior. El ojo de pez situado en la parte superior no se ha detectado debido a que se encuentra demasiado cerca de la cámara y por lo tanto la pupila tiene un área demasiado grande como para entrar en el rango permitido.



*Fig. 3.36 Detalle de candidatos detectado*

En la Figura 3.36 se observa varios candidatos por tamaño detectados, entre los que hay tres pupilas de ojos de lubinas.

El rango permitido se definirá en función de la distancia de interés. Esta última será tal que el pez objetivo se vea en toda su longitud en la imagen. Con la mayor parte de imágenes, un rango que da buen resultado es [15,80].

### 3.4.5 Filtrado por forma

El segundo paso, una vez que se tiene un grupo de candidatos seleccionados por su área, es comprobar la siguiente característica que deben cumplir los blobs para seguir siendo considerados candidatos a ojos. Esta característica es su redondez, es decir, se seleccionarán aquellos dentro de los candidatos que tengan forma redondeada.

Para determinar la forma de un blob se utiliza la elongación. La elongación se define como:

$$\text{elongacion} = \frac{\text{diametro\_max}}{\text{diametro\_min}} \quad (\text{Ec. 3.10})$$

Para medir los diámetros se toma como centro de la posible circunferencia el centro de gravedad del blob que es candidato a ojo por su área. La unidad de medida va a ser el pixel. Como ya se ha dicho el centro de

gravidad se calcula con la librería para el tratamiento de blob desarrollada en el GRVC.

Para calcular los diámetros se recorre el blob hasta su borde en un determinado ángulo y en ese ángulo más  $180^\circ$  contando el número de píxels.

Un blob que sea perfectamente circular tendrá una elongación igual a uno. En el caso de estudio se especificará un rango de elongación admisible, definiendo una elongación máxima y mínima. Mediante métodos experimentales, que se expondrán en el Capítulo 5, se determinan estas elongaciones.

Los blobs cuya elongación esté en el rango deseado seguirán siendo considerados candidatos, el resto serán desestimados.

Debido a sombras y reflejos, tras binarizar la imagen algunas pupilas aparecerán deformadas, por lo que la aplicación no considerará esos blobs como candidatos a ojos y los desestimará. En la Figura 3.37 se observa un ejemplo de ello, donde la elongación es alta.



*Fig. 3.37 Blob desestimado por forma.*

En la Figura 3.38 se puede ver un ejemplo de detección de varios blobs candidatos por su forma, entre los que se encuentra una pupila.



*Fig. 3.38 Blob candidato a ojo por su forma*

### 3.4.6 Filtrado por Radios

En este apartado se explica la última propiedad que han de cumplir los blobs candidatos para ser considerados ojos definitivamente.

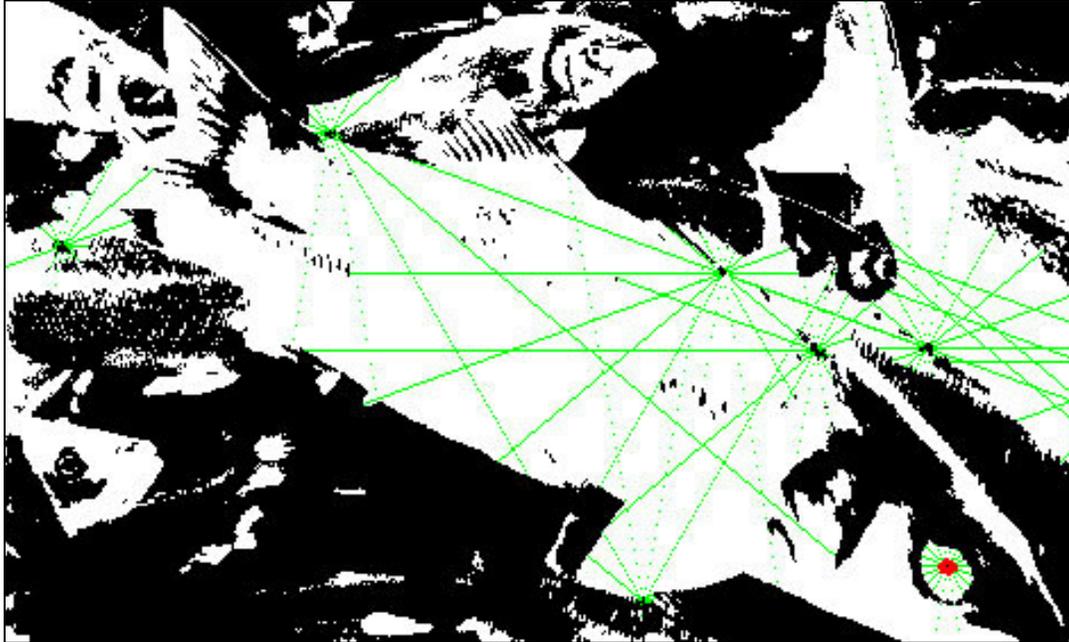
Como se explico en el Apartado 3.4.2, se va a considerar que un ojo de un pez está compuesto por un círculo de tonalidad muy clara sobre el que se superpone otro círculo negro, de área menor y situado en su centro. El círculo blanco se encuentra sobre un fondo de tonalidad más oscura.

Si se representa la imagen en tono de grises en tres dimensiones, tomando como coordenadas  $[x, y, z] = [\text{ancho}, \text{alto}, \text{nivel de gris}]$ , los ojos vienen representados por figuras que se asemejarían a volcanes. La pupila sería el cráter, y la parte blanca sería la ladera.

El círculo pequeño ya se ha localizado, ahora se comprobará que los candidatos son pupilas. Para ello se trazan radios desde el centro de gravedad de la pupila hasta el borde del círculo de tonalidad clara, y se comprueba la longitud de estos radios.

La longitud que han de tener esos radios desde el borde de la pupila hasta el borde del círculo blanco es aproximadamente el diámetro de la pupila.

La separación de los radios en grados es configurable, así como el porcentaje de radios correctos que se han de trazar para considerar que el blob candidato es la pupila del ojo de un pez. Se ajustarán estos parámetros para lograr la máxima detección con el menor número de falsas alarmas.



*Fig. 3.39 Trazados de radios ojos*

Los blob candidatos que cumplen esta propiedad son considerados ojos.

Habrán muchos ojos que no se detecten debido que es complicado detectar la zona que limita el círculo blanco que rodea la pupila. Será preferible detectar un número menor de ojos y reducir las falsas alarmas.

A continuación se presentan varios ejemplos del resultado obtenido con una buena configuración para reducir el número de falsas alarmas.



Fig. 3.40 Ejemplo de ojo no detectado.

En la Figura 3.40 no se ha detectado el ojo debido a que el porcentaje de radios correctos no ha superado el límite de la configuración, que en este ejemplo era del 75% para radios separados  $10^\circ$ .



Fig. 3.41 Ejemplo de ojo detectado. Situación ideal.

La Figura 3.41 representa la situación ideal, en la que el porcentaje de radios correctos es del 100%.



*Fig. 3.42 Ejemplo de ojo detectado.*

En la Figura 3.42, pese a no estar la completa la corona que rodea el ojo, si se ha detectado porque el porcentaje anteriormente citado si supera el límite en este caso.

### 3.5 Resumen

Como se ha visto en las líneas de este capítulo el método propuesto se basa en la búsqueda de blobs que cumplan una serie de características.

En primer lugar se prepara las imágenes mediante el preprocesado para obtener una buena binarización.

En segundo lugar se obtiene un modelo de iluminación de la jaula marina para realizar la umbralización.

En último lugar se busca en las imágenes blobs que cumplan las características deseadas. Esta búsqueda se hace en tres pasos en cascada:

- Filtrado de blobs por tamaño deseado
- Filtrado de blobs por forma deseada
- Filtrado de blobs por porcentaje de radios correctos

Los blobs que pasen los tres filtros serán considerados ojos.