ISSN 1870-4069

Sistema prototipo de monitoreo subacuático automático de peces por visión estereoscópica y aprendizaje profundo

Héctor Carlos Aranda-Martínez[1], Nidiyare Hevia-Montiel[2]

[1] Universidad Nacional Autónoma de México,
Posgrado en Ciencia e Ingeniería de la Computación,
México

[2] Universidad Nacional Autónoma de México,
Unidad Académica del Instituto de Investigaciones en
Matemáticas Aplicadas y en Sistemas en el Estado de Yucatán,
México

carlosaranda@comunidad.unam.mx,
nidiyare.hevia@iimas.unam.mx

Resumen. En este trabajo se presenta la propuesta de un prototipo de monitoreo subacuático mediante visión estereoscópica, técnicas de visión computacional, y aprendizaje computacional para obtener información espacial de peces (posición y longitud) de forma automatizada: su detección y localización en ambas vistas es generada por una Red Neural Convolucional tipo YOLO (You Only Look Once); mediante geometría epipolar se reconstruye la posición espacial de puntos de interés para rastreo de cada pez; finalmente se estima la posición y longitud de cada pez localizado. En un entorno controlado se ha obtenido un rendimiento satisfactorio, con errores de entre el 0.75 y el 2.5 % de la longitud real.

Palabras clave: Monitoreo subacuático, visión estereoscópica, aprendizaje profundo, geometría epipolar.

Prototype System for Automatic Underwater Monitoring of Fish by Stereoscopic Vision and Deep Learning

Abstract. This paper presents a fish underwater monitoring prototype using stereoscopic vision, computer vision, and deep learning techniques to obtain spatial information of the fish (position and length) in an automated way: fish detection and localization from the two steroscopic views is generated by a YOLO (You Only Look Once) Convolutional Neural Network; by epipolar geometry, spatial position of points of interest for fish tracking is reconstructed; position and length of each detected fish is estimated. For controlled environment tests, results show a satisfactory performance with an error range between 0.75 and 2.5% of real length.

Keywords: Underwater monitoring, stereo vision, deep learning, epipolar geometry.

1. Introducción

La obtención de datos sobre la fauna submarina ha sido siempre de gran interés, ya sea económico o científico. Especialmente, hablando de intereses económicos, en la industria acuícola es muy importante conocer datos estadísticos de los peces (peso, dimensiones y conteo) [10]. Sin embargo, estas tareas son todavía realizadas en gran medida manualmente, por lo que su automatización tendría un gran impacto en este sector productivo. En el monitoreo subacuático por video (o imágenes) es posible utilizar una o más cámaras. Para el caso de utilizar una cámara se tiene la dificultad de realizar mediciones del entorno sin elementos de referencia, que permitan recuperar la información espacial del objeto de interés.

El uso de trampas por donde se hacen pasar los peces [15], o de elementos de dimensiones conocidas en las imágenes [5] son algunas medidas utilizadas para sobrepasar tal dificultad. Para el caso de un sistema de dos cámaras, o estereoscópico, la ventaja sobre el sistema monocular es la cantidad de información que se puede obtener del entorno, por lo que hacer mediciones sobre el mismo genera mejores resultados aunque aumenta la complejidad del procesamiento [14, 21]. Uno de los principales inconvenientes para su completa automatización es la detección de los puntos de interés para hacer las mediciones pertinentes.

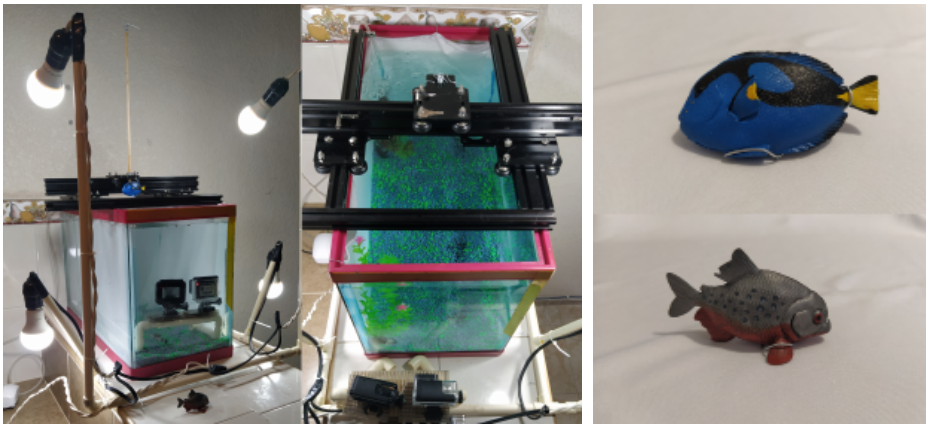
Shortis et al. [19] realizaron un compendio de técnicas y métodos de visión computacional en el monitoreo subacuático. Para la tarea de medición de características de peces, algunos procedimientos se realizan de forma manual [8, 21] o semiautomática [16]. Rodriguez et al. [14] proponen un método de medición de peces por visión estereoscópica que alcanza un error relativo promedio del $10\,\%$. Muñoz et al. [9] proponen un método automático para medir las longitudes de atunes en ambientes reales, utilizando visión estereoscópica y un modelo deformable de la forma del pez, logrando una evaluación del Kolmogorov-Smirnov p-value de $0{,}0183$.

Posteriormente Shi et al. [17] desarrollaron un método basado en detección de movimiento y generación del mapa de disparidad para medir longitudes de peces logrando un error cuadrático promedio relativo del $1.22\,\%$. En este trabajo se desarrolla la propuesta de un prototipo estereoscópico para monitoreo subacuático (de ahora en adelante identificado como PEMS), que a partir de datos de video pueda detectar, localizar y estimar las longitudes de los peces que se encuentran en escena de manera automática. En la primera parte se explican los métodos y materiales utilizados; después el diseño y realización de los experimentos; y finalmente se mostrarán los resultados obtenidos.

2. Materiales y métodos

2.1. Sistema físico de pruebas

Se cuentan con dos cámaras GoPro Hero®, capaces de sumergirse bajo el agua. Se configuraron para trabajar con un campo de visión lineal, a 1080p y 30 cuadros por segundo. Se trabajará en un ambiente controlado (dimensiones, posiciones, luminosidad y turbidez del agua, ver figura 1a) que se compone de una pecera de 26 cm de ancho, 35.5 cm de alto y 51 cm de largo, 4 fuentes de luz y un sistema de rieles.



(a) (b)

Fig. 1. Montaje del ambiente controlado. a) Sistema de rieles y de iluminación para controlar posición de phantoms y uniformidad lumínica. b) Phantoms utilizados para las pruebas: pez cirujano (Paracanthurus hepatus, arriba) y piraña (Serrasalmus nattereri, abajo).

Con los últimos dos elementos se controla la uniformidad lumínica y la posición de los phantoms usados en las pruebas, respectivamente. En la figura 2 se esquematiza la posición de las cámaras C_1 y C_2, así como de las 30 locaciones distintas donde se colocarán los phantoms, marcadas por las **X**. La calibración del sistema de cámaras estereoscópico toma como referencia a la cámara izquierda (C_1), por lo que se utilizarán sus coordenadas para encontrar la posición relativa de cada punto al PEMS.

2.2. Obtención de datos

La base de datos será conformada por archivos de video de 1920×1080 píxeles, en espacio de color RGB. Estos archivos deben contener un evento visual de sincronización, es decir, un evento puntual que pueda ser captado por ambas cámaras y que sirva para sincronizar los archivos. Para esta sincronización se utilizó un evento visual creado por el encendido y apagado de una luz, mediante el cual se ajustó la reproducción de los video. La decisión de utilizar video en lugar de imágenes se fundamenta en el deseo de trabajar con organismos vivos que se mueven en su entorno (para los que no siempre se tendrá la mejor fotografía), y la posibilidad de ajustar los errores en las mediciones según la cantidad de información recibida.

2.3. Calibración

Tal como lo propone [18], se utilizó un patrón de ajedrez de dimensiones conocidas para hacer la calibración subacuática. Se utilizó el algoritmo propuesto por Zhang [23]. Este algoritmo toma un conjunto de puntos $m = [u, v]^T$ conocidos en la imagen de calibración correspondientes a puntos conocidos en el patrón de calibración (ver figura 3) $M = [X, Y, Z]^T$.

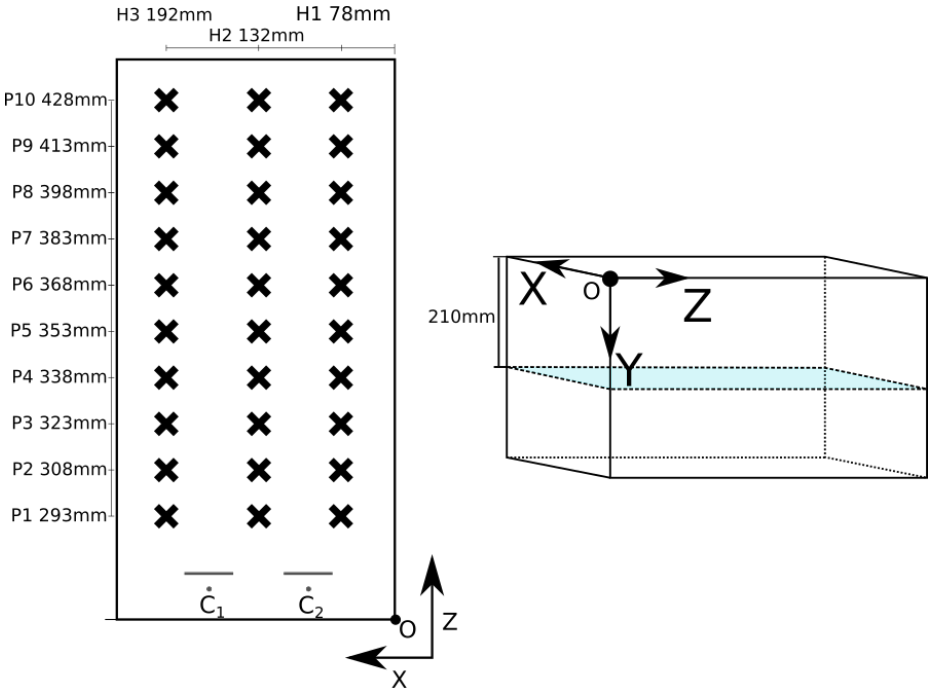


Fig. 2. Posiciones **X** identificadas por las coordenadas (H, P) y utilizadas durante las pruebas. El origen (O) del sistema de referencia espacial se encuentra en la esquina superior derecha de la pecera; el phantom se colocó en el eje Y a 210 mm; C_1 y C_2 son las cámaras izquierda y derecha.

Con estos puntos se estima la matriz de calibración interna A, la matriz de rotación R y el vector de traslación t de la cámara, de manera que se cumpla:

$$s\tilde{m} = A \left[R\, t\right] \tilde{M}, \tag{1}$$

donde s es un factor de escalamiento y $\tilde{m}$ y $\tilde{M}$ son los puntos m y M en coordenadas homogéneas. El objetivo del algoritmo es minimizar el error de proyección de los puntos. Los coeficientes de distorsión radial también son considerados dentro de la calibración.

Una vez hecha la calibración de cada cámara, se procede a calibrar el sistema de dos vistas, lo que implica encontrar:la matriz de rotación (R), que rota el sistema coordenado de la segunda cámara C_2 para que corresponda con la primera C_1; al vector de traslación (t), que traslada el sistema coordenado de C_2 al de C_1, la matriz esencial (E), que describe la relación entre puntos en los sistemas coordenados de ambas cámaras; y la matriz fundamental (F), que describe la relación entre los puntos en los planos de la imagen de ambas cámaras.

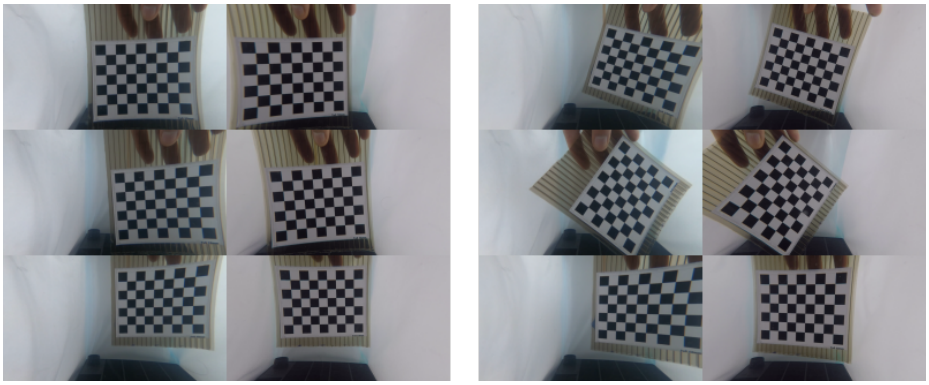


Fig. 3. Ejemplos de capturas subacuáticas del tablero de calibración.

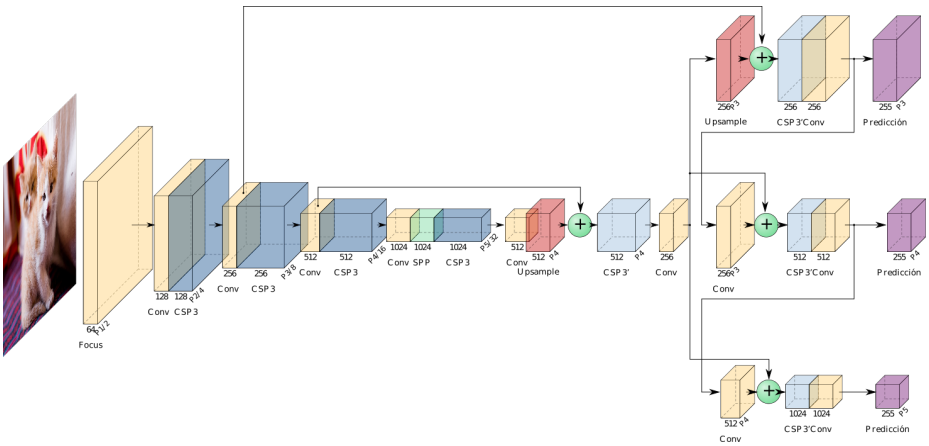


Fig. 4. Arquitectura Yolov5l.

2.4. Localización automática de peces

Para este proceso se optó por utilizar una red neuronal artificial tipo YOLO (You Only Look Once) [12], precisamente un modelo tipo YOLOv5 [7]. Estas arquitecturas de redes han demostrado tener desempeños tan buenos como para ser implementadas en sistemas de detección en tiempo real.

En comparativa con otra familia de arquitecturas llamada EfficientDet [20], la familia YOLOv5 tiene una rapidez de hasta 5 veces mayor sin disminuir su desempeño. Esto deja en evidencia que los tiempos de inferencia para una red tipo YOLOv5 permiten su uso en sistemas de detección en tiempo real.

Este grupo de arquitecturas cuenta con 4 predeterminadas: **s**, **m**, **l** y **x**. Cada una de ellas se diferencia por su profundidad, siendo la **s** la de menor cantidad de capas y la **x** la más profunda; en este trabajo se utilizó la configuración **l** (ver figura 4), que se compone de ciertos bloques que se reutilizan y se explican a continuación.

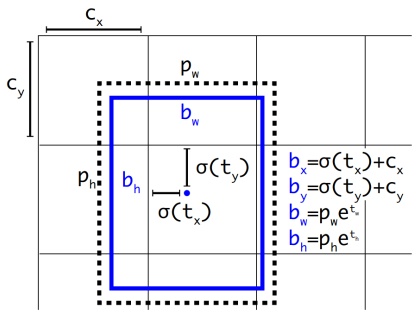


Fig. 5. Cálculo del centro y dimensiones de la caja de detección del objeto. c_x y c_y corresponden a las coordenadas de la celda responsable de la detección. p_w y p_h son las dimensiones predefinidas del anchor box utilizado. Obtenida de [12].

– **Focus.** Esta primer capa hace una reducción tipo SpaceToDepth, tal como se menciona en [13]. Este primer proceso es más rápido y tiene menor pérdida de información que otras capas donde solamente se hace una convolución con reducción de resolución.

– **Conv.** Se compone de una capa de convolución seguida de una normalización por lote (para el entrenamiento, según [6]), y una capa de activación SiLU [11] expresada por:

$$\text{silu}(x) = x * \sigma(x), \tag{2}$$

donde σ es la función sigmoide logística.

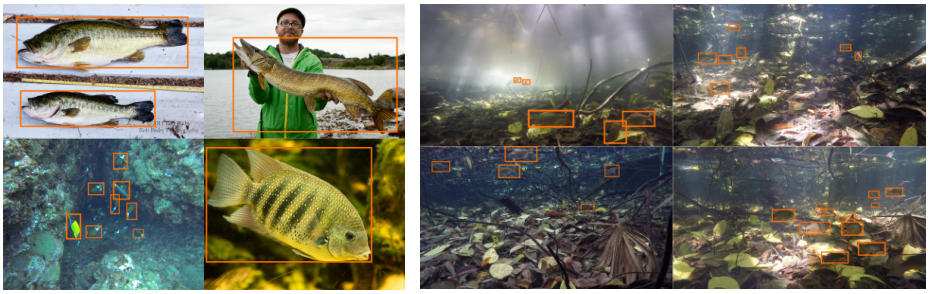
– **CSP 3 (Cross Stage Partial).** Este bloque está basado en la arquitectura propuesta por [22]. Se compone de una capa de cuello de botella (Bottleneck) en conjunto con tres operaciones de convolución. Estos bloques han demostrado disminuir el costo computacional sin sacrificar el desempeño de la red (incluso mejorando la precisión de tareas como la clasificación).

– **SSP (Spatial Pyramid Pooling).** Esta capa permite que la red acepte imágenes de cualquier dimensión, lo que permite manejar cualquier tipo de escalas y razones de aspectos (aspect ratios). Su implementación está basada en [4].

Finalmente, se tienen 3 capas de salida, cada una a diferentes resoluciones ($I3$, $I4$ y $I5$). Esto le permite a la red hacer detecciones de objetos muy pequeños ($I3$), medianos ($I4$) y grandes ($I5$). Cada celda (vector) de estos tensores se compone por 3 secciones formadas por el siguiente vector $(t_x, t_y, t_w, t_h, P_0, p^T)^T$, donde t_x y t_y son las predicciones del centro de la región predicha, t_w y t_h representan la altura y el ancho de la caja, P_0 es la probabilidad que la caja contenga un objeto y p es el vector de clasificación (que predice la clase a la que pertenece el objeto identificado).

Son 3 secciones, ya que se trabajan con 3 tamaños de cajas predefinidas (anchor boxes), y cada celda se encarga de ajustarlas todas.

Para la localización se considera la sección que contenga la mayor puntuación en p^T. Posteriormente se calculan las coordenadas relativas de la detección en la imagen original, como se muestra en la figura 5.



(a) (b)

Fig. 6. Ejemplos de imágenes de la base de datos. a) imágenes de Open IMages Dataset, b) imágenes de datos etiquetados por [1].

Después los resultados son filtrados por el método de supresión no máxima (Non-Maximum Suppresion), para eliminar las regiones propuestas que puedan representar el mismo objeto y estén superpuestas. Este proceso termina generando un arreglo de $n \times (5 + c)$, donde n es el número de detecciones válidas y c el número de clases (para el prototipo propuesto se consideró 1).

Para el entrenamiento se utilizaron 2767 imágenes (1883 de entrenamiento y 884 de prueba) provenientes de: Open IMages Dataset[3], figura 6a; y del conjunto de datos obtenido por [2] y procesado por [1], que está conformado por imágenes de peces en petenes del estado de Yucatán (figura 6b).

El entrenamiento se hizo en un equipo con tarjeta NVIDIA Tesla T4, 8 imágenes por lote, 1000 épocas, el tamaño de la imagen de entrenamiento y de inferencia de 448×448 px. Se utilizó transferencia de aprendizaje con los pesos del modelo YOLOv5m (pre-entrenado con la base de datos COCO val2017).

2.5. Puntos de interés

La red genera la región de interés (ROI) que contiene al pez de tal forma que tanto la boca como la aleta caudal tocan lados opuestos de esta. Esto nos permite tener una buena aproximación a las dimensiones (longitud) del pez. Se toman entonces como puntos de interés los correspondientes al punto medio de la altura de la ROI en cada lado de la misma (ver figura 7).

2.6. Reconstrucción tridimensional

Al hablar de reconstrucción 3D nos referimos a recuperar la información espacial de los puntos de interés encontrados en la escena, no debe interpretarse como la reconstrucción tridimensional del objeto completo.

La reconstrucción se basó en geometría epipolar, utilizando la información de la calibración de las cámaras. Es probable que para los puntos de interés seleccionados en ambas vistas no se cumpla que:

$$m_2^T F m_1 = 0, \tag{3}$$

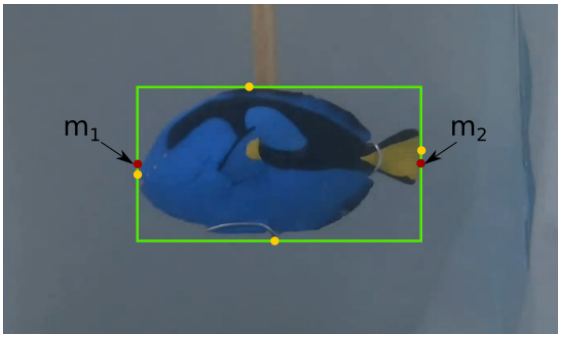


Fig. 7. En verde se muestra la región de interés que contiene al phantom de pez cirujano. Los puntos amarillos representan puntos de contacto del contorno del pez y el recuadro verde (correspondiente con la región de interés que lo contiene). Los puntos rojos m_1 y m_2 son los puntos de interés utilizados.

donde m_1 es el punto en la primera imagen y m_2 en la segunda. Esto puede ser debido a errores en la selección de los puntos o deficiencias en la calibración (cálculo de la matriz fundamental F). Para ello se hace una corrección de la siguiente manera:

1. Considerar que existen los puntos $\hat{m}_1$ y $\hat{m}_2$ tal que $\hat{m}_2^T F \hat{m}_1 = 0$, y que se encuentran cercanos a m_1 y m_2.
2. La distancia euclideana entre dos puntos está expresada por $d(p_i, p_j)$.
3. Definimos una función error que sirva para minimizar la distancia entre los puntos:

$$E(m_1, m_2) = d(m_1, \hat{m}_1)^2 + d(m_2, \hat{m}_2)^2. \tag{4}$$

Siempre que se cumpla $\hat{m}_2^T F \hat{m}_1 = 0$, donde $\hat{m}_1$ y $\hat{m}_2$ son los puntos corregidos.

Con los puntos corregidos se procede a hacer la triangulación de los mismos según el algoritmo descrito por [3]. En él, se genera una ecuación lineal de la forma $BX = 0$ a partir del conocimiento de las matrices de proyección de ambas cámaras (P_1 y P_2) y los puntos corregidos (como se describió anteriormente). El desarrollo de este sistema de ecuaciones considera la eliminación del factor de escala mediante el producto cruz $m_i \times (P_i M_i) = 0$. Esto genera las ecuaciones:

$$m_{i,x}(p_i^{3T} M) - (p_i^{1T} M) = 0, \tag{5}$$
$$m_{i,y}(p_i^{3T} M) - (p_i^{2T} M) = 0, \tag{6}$$
$$m_{i,x}(p_i^{2T} M) - m_{i,y}(p_i^{1T} M) = 0, \tag{7}$$

donde p_i^k es la k-ésima fila de la matriz de proyección P_i, y $m_i = (m_{i,x}, m_{i,y})^T$. De esto observamos que solamente dos de las 3 ecuaciones son linealmente independientes.

[3] https://storage.googleapis.com/openimages/web/index.html

Además, estas ecuaciones son lineales en términos de X, por lo que la ecuación $BX = 0$ tiene como valor de B:

$$B = \begin{bmatrix} m_{1,x}p_1^{3T} - p_1^{1T} \\ m_{1,y}p_1^{3T} - p_1^{2T} \\ m_{2,x}p_2^{3T} - p_2^{1T} \\ m_{2,y}p_2^{3T} - p_2^{1T} \end{bmatrix}. \tag{8}$$

La solución de dicha ecuación dará como resultado el punto espacial Q_w en coordenadas homogéneas.

2.7. Cálculo de la longitud del pez

La triangulación de los puntos anteriores nos permite estimar su posición espacial. Nuestro principal interés es hacer la medición de la longitud del pez localizado, por lo que se consideró que los puntos anteriormente obtenidos ($q_{w,\text{boca}}$ y $q_{w,\text{aleta}}$ en coordenadas no homogéneas) corresponden a los dos extremos horizontales del pez. Así, la longitud de cada pez puede ser estimada según:

$$L_{\text{estimada}} = ||q_{w,\text{boca}} - q_{w,\text{aleta}}||. \tag{9}$$

2.8. Validación de resultados

Los resultados del PEMS se validarán utilizando los datos conocidos de los tamaños de los peces utilizados, así como de las dimensiones del ambiente en el que se hacen las pruebas. Se toman en cuenta dos mediciones: 1) la longitud del pez, y 2) la posición relativa de éste respecto al sistema de cámaras.

Estas estimaciones se harán con base en los puntos de interés ($m_{i,1}$ y $m_{i,2}$) localizados en las imágenes y devueltos por el sistema de detección con los cuales se puede conocer su posición espacial ($M_{w,1}$ y $M_{w,2}$).

Con la información obtenida anteriormente es importante evaluar dos propiedades, considerando que ($M_{w,1}$ y $M_{w,2}$) se encuentran en extremos contrarios del pez. La primera corresponde a la longitud del pez (L_S), calculada de la siguiente manera:

$$L_S = ||M_{w,1} - M_{w,2}||. \tag{10}$$

La segunda corresponde con la distancia del punto espacial donde se estima que se encuentra el pez a la posición donde realmente se encuentra. Para esto se calcula el centro de masa del objeto $M_{w,c}$, considerando que ($M_{w,1}$ y $M_{w,2}$) se encuentran en extremos contrarios del pez, con la siguiente ecuación:

$$M_{w,c} = \frac{1}{2}(M_{w,1} + M_{w,2}). \tag{11}$$

Para evaluar estas métricas se proponen dos funciones de error:

$$e_L = \left| \left(1 - \frac{L_S}{L} \right) \right| \times 100, \tag{12}$$

$$e_D = ||M_{w,c} - M_w||, \tag{13}$$

Tabla 1. Se muestra la longitud estimada promedio y la desviación estándar promedio de cada posición (H, P) (en mm). Las longitudes reales son: 70 mm para el phantom de pez cirujano, y 66 mm para el de pez piraña.

	Cirujano			**Piraña**		
	H3	**H2**	**H1**	**H3**	**H2**	**H1**
P1	71.66± 0.47	71.38± 0.43	**72.17**± 0.93	67.90± 0.63	67.05± 0.31	66.99± 0.25
P2	**72.28**± 0.70	71.55± 0.45	70.43± 1.30	**67.91**± 0.40	**66.99**± 0.21	67.59± 0.29
P3	71.48± 0.70	**72.05**± 1.25	69.55± 0.55	66.96± 0.34	**67.54**± 0.24	**67.70**± 0.34
P4	70.48± 1.70	69.75± 0.71	69.46± 0.70	66.42± 0.43	67.42± 0.22	**66.06**± 0.25
P5	70.49± 0.61	70.12± 0.29	**69.36**± 0.59	**66.11**± 0.51	67.46± 0.32	66.60± 0.26
P6	71.11± 0.45	70.15± 0.50	68.80± 0.96	65.83± 0.37	67.28± 0.23	66.75± 0.18
P7	70.46± 0.87	70.40± 0.53	69.52± 1.33	66.62± 0.71	67.01± 0.32	67.09± 0.26
P8	69.52± 0.25	69.69± 1.39	69.74± 1.10	66.44± 0.60	67.20± 0.40	66.48± 0.48
P9	69.32± 0.50	70.21± 0.57	69.90± 1.21	67.65± 0.76	67.15± 0.44	66.66± 0.55
P10	**68.76**± 0.82	**69.66**± 1.06	70.10± 0.90	66.44± 0.69	67.03± 0.34	66.22± 0.20

donde L es la longitud real y M_w es la posición espacial relativa a las cámaras del phantom utilizado. e_L puede considerarse como un error porcentual que nos permite intuir rápidamente qué tan alejada está la estimación del valor real.

3. Resultados

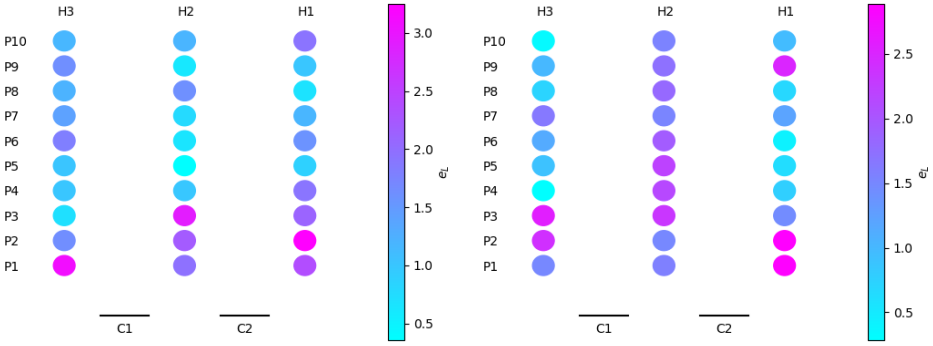
El procedimiento de obtención de datos fue el siguiente: calibrar el sistema estereoscópico de cámaras, colocar el phantom en una de las posiciones, obtener un registro de video de 5 segundos en dicha posición, cambiar a una nueva posición y repetir desde el paso 2 hasta agotarse las posiciones. Los datos de video fueron procesados por el PEMS que se encara de detectar, cuadro a cuadro, el phantom que se encuentra en la escena.

Por cada uno devuelve los puntos de interés $(m_{1,1}, m_{1,2})$ y $(m_{2,1}, m_{2,2})$ de cada vista basándose en la región encontrada anteriormente. Después hace la reconstrucción tridimensional de cada par de puntos de interés $M_{w,1}$ y $M_{w,2}$. Calcula la distancia que existe entre el par de puntos, esta es la longitud estimada del objeto L_S.

Y finalmente calcula los errores para validación e_L y e_D. Es importante mencionar que para este proceso, con las características del equipo utilizado, cada imagen toma 0.8 segundos en promedio en procesarse.

Se realizaron 5 repeticiones de las adquisiciones para cada phantom, de todas estas repeticiones se creó una sola tabla que contiene los valores promedio por cada posición (H, P) (ver la tabla 1).

Esta información está sintetizada en la figura 8, donde podemos darnos una idea de la zona espacial donde la exactitud de nuestro método es mayor para cada uno de los casos. Gracias al resultado anterior podemos darnos cuenta que la variación en las mediciones parece depender más de las posiciones en P que en H.



(a) Errores del phantom de pez cirujano.

(b) Errores del phantom de pez piraña.

Fig. 8. Representación gráfica de los errores por posición. Se muestra la posición relativa a las cámaras C_1 y C_2. Estos puntos corresponden con los esquematizados en la figura 2.

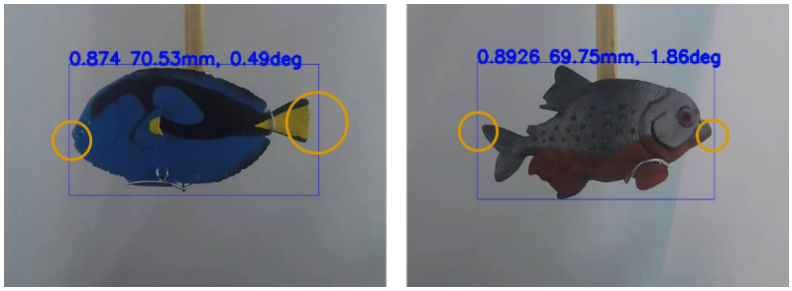


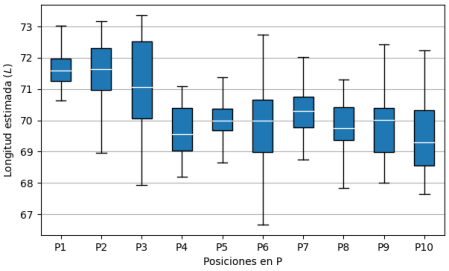
Fig. 9. Se muestran los errores en la localización de los peces en las imágenes. Para ambos casos existen cuadros donde la región de interés no se ajusta bien a la figura del pez. Estas diferencias, aunque parezcan pequeñas, pueden representar errores del orden de 2 o 3mm en las mediciones. En la parte superior de la detección se pueden apreciar tres valores, de izquierda a derecha tenemos: el valor de confianza de la detección, la longitud instantánea, el ángulo de inclinación del pez respecto al plano de la imagen.

Este resultado fue esperado por dos razones: 1) la posición espacial de las imágenes que se utilizaron para realizar la calibración corresponden con las posiciones de menor error (se explica más adelante); y 2) en ocasiones la red neuronal convolucional no devuelve una región que contenga perfectamente al pez, lo que genera errores en la posición de los puntos de interés (ver figura 9).

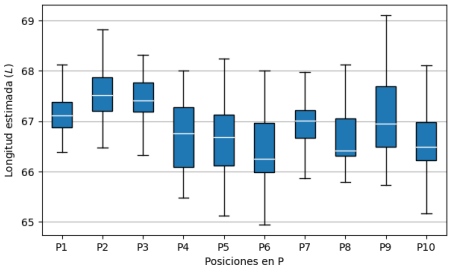
Para poder visualizar mejor el comportamiento de las estimaciones de las longitudes a lo largo de las posiciones en P, podemos revisar los resultados de la figura 10.

Como se espera el comportamiento en ambos casos es similar, comenzando con un error relativamente alto en posiciones cercanas a las cámaras y disminuyendo a medida que se aleja.

El por qué de este comportamiento puede ser explicado por la misma calibración, ya que el patrón de ajedrez utilizado fue posicionado a distancias mayores a los 35 cm de las cámaras (debido a las dimensiones de éste), es por esto que se esperaba obtener menor error en las mediciones cercanas a la región espacial de la calibración.

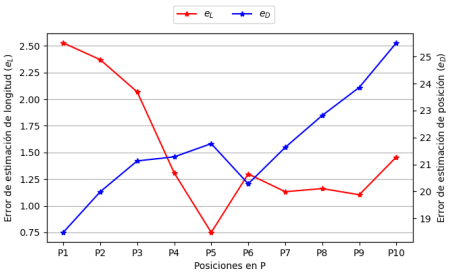


(a) Longitudes estimadas del phantom de pez cirujano.

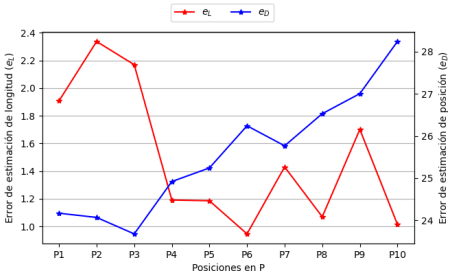


(b) Longitudes estimadas del phantom de pez piraña.

Fig. 10. Presentación de las mediciones de longitudes en gráficos de caja. Para este análisis se utilizaron las mediciones en las posiciones $H1$, $H2$ y $H3$ para cada posición en P. Recordemos que la longitud real del pez cirujano es de 70 mm y del pez piraña de 66 mm.



(a) Errores para las mediciones del phantom de pez cirujano.



(b) Errores para las mediciones del phantom de pez piraña.

Fig. 11. Errores de las mediciones de longitud e_L y posicionamiento e_D.

Finalmente, podemos ver una síntesis de la evaluación de los errores en la figura 11, donde cada punto corresponde al promedio del error evaluado por cada medición de todas las posiciones H para cada posición P. Claramente vemos una relación directa entre los comportamientos de esta figura con los de la figura 10.

De ésta podemos notar dos comportamientos: 1) El error de estimación de longitud e_L disminuye a medida que la posición es más lejana a las cámaras; y 2) El error de estimación de la posición e_D aumenta casi de forma lineal. Lo primero es de esperarse dado que estos valores fueron obtenidos de las longitudes estimadas. Lo segundo parece ser más un problema de la resolución espacial del objeto en la imagen digital.

A medida que éste se aleja de las cámaras, la exactitud de la reconstrucción tridimensional disminuye, lo que provoca errores en la recuperación de las coordenadas espaciales de los puntos de interés.

Ya que la región generada por la red neuronal es consistente, este error también lo es para ambos puntos, lo que hace que se calcule la posición del objeto considerando dicho error (que es relativo al sistema), pero que no afecte directamente a la estimación de la longitud (que es relativa al objeto).

A pesar de todo esto podemos ver que los errores en las estimaciones de longitud e_L obtenidos son relativamente pequeños; el error más grande ronda el 2.5 % de la longitud

real y el más pequeño el 0.75 %. Estos resultados son satisfactorios considerando la inexactitud de la red neuronal al generar la región que contiene al pez.

4. Conclusiones

El prototipo PEMS ha demostrado generar buenos resultados en la detección y estimación de longitud de phantoms de peces, obteniendo errores de entre el 0.75 y el 2.5 % de la longitud real, lo cual se encuentra cercano a lo obtenido por [17] y mejorando lo obtenido por [14]. Es necesario mejorar la velocidad de procesamiento para poder implementarlo en un sistema en tiempo real. Así también, continuar las pruebas para trabajar con peces en vivo en ambientes semi-controlados, lo cual implicará tomar otras consideraciones dada la forma y movimiento de los mismos.

Referencias

1. Castillo-Varguez, E. J.: Redes neuronales profundas para la detección de peces en ambientes subacuáticos (2019)
2. Espinosa-Mendoza, D. A.: Variaciones temporales de la comunidad de peces en un humedal costero de Yucatán, mediante imágenes subacuáticas y técnicas tradicionales. Master's thesis, UNAM, Instituto de Ciencias del Mar y Limnología (2019)
3. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, 2nd edition (2003)
4. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. Lecture Notes in Computer Science, pp. 346–361 (2014) doi: 10.48550/ARXIV.1406.4729
5. Hsieh, C. L., Chang, H. Y., Chen, F. H., Liou, J. H., Chang, S. K., Lin, T. T.: A simple and effective digital imaging approach for tuna fish length measurement compatible with fishing operations. Computers and Electronics in Agriculture, vol. 75, no. 1, pp. 44–51 (2011) doi: 10.1016/j.compag.2010.09.009
6. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift (2015) doi: 10.48550/ARXIV.1502.03167
7. Jocher, G., Stoken, A., Borovec, J., Changyu, L., Laughing, Tkianai, Hogan, A., Ayush Chaurasia, Diaconu, L., Ingham, F., Colmagro, A., Ye, H., Poznanski, J., Fang, J., Kim, J., Doan, K., Yu, L.: ultralytics/yolov5: v4.0 - nn.silu() activations, weights; biases logging, pytorch hub integration (2021) doi: 10.5281/ZENODO.4418161, URL https://zenodo.org/record/4418161
8. Komeyama, K., Tanaka, T., Yamaguchi, T., Asaumi, S., Torisawa, S., Takagi, T.: Body measurement of reared red sea bream using stereo vision. Journal of Robotics and Mechatronics, vol. 30, no. 2, pp. 231–237 (2018) doi: 10.20965/jrm.2018.p0231
9. Muñoz-Benavent, P., Andreu-García, G., Valiente-González, J. M., Atienza-Vanacloig, V., Puig-Pons, V., Espinosa, V.: Enhanced fish bending model for automatic tuna sizing using computer vision. Computers and Electronics in Agriculture, vol. 150, no. 1, pp. 52–61 (2018) doi: 10.1016/j.compag.2018.04.005
10. Perez, D., Ferrero, F. J., Alvarez, I., Valledor, M., Campo, J. C.: Automatic measurement of fish size using stereo vision. In: Proceedings of the IEEE International Instrumentation and Measurement Technology Conference, pp. 1–6 (2018)
11. Ramachandran, P., Zoph, B., Le, Q. V.: Searching for activation functions (2017) doi: 10.48550/ARXIV.1710.05941

12. Redmon, J., Farhadi, A.: YOLOv3: An incremental improvement (2018) doi: 10.48550/ARXIV.1804.02767
13. Ridnik, T., Lawen, H., Noy, A., Baruch, E. B., Sharir, G., Friedman, I.: Tresnet: High performance GPU-dedicated architecture (2020) doi: 10.48550/ARXIV.2003.13630
14. Rodriguez, A., Rico-Diaz, A. J., Rabuñal, J. R., Puertas, J., Pena, L.: Fish monitoring and sizing using computer vision. In: Fish monitoring and sizing using computer vision (eds. Ferrández-Vicente, J. M., Álvarez-Sánchez, J. R., de la Paz-López, F., Toledo-Moreo, F. J., Adeli, H.), vol. 9108, pp. 419–428 (2015)
15. Sanchez-Torres, G., Ceballos-Arroyo, A., Robles-Serrano, S.: Automatic measurement of fish weight and size by processing underwater hatchery images. Engineering Letters, vol. 24, no. 4, pp. 461–472 (2018)
16. Shafait, F., Harvey, E. S., Shortis, M. R., Mian, A., Ravanbakhsh, M., Seager, J. W., Culverhouse, P. F., Cline, D. E., Edgington, D. R.: Towards automating underwater measurement of fish length: A comparison of semi-automatic and manual stereo–video measurements. ICES Journal of Marine Science, vol. 74, no. 6, pp. 1690–1701 (2017) doi: 10.1093/icesjms/fsx007
17. Shi, C., Wang, Q., He, X., Zhang, X., Li, D.: An automatic method of fish length estimation using underwater stereo system based on LabVIEW. Computers and Electronics in Agriculture, vol. 173, pp. 105419 (2020) doi: 10.1016/j.compag.2020.105419
18. Shortis, M.: Camera Calibration Techniques for Accurate Measurement Underwater, Springer International Publishing, Cham, pp. 11–27 (2019) doi: 10.1007/978-3-030-03635-5_2
19. Shortis, M. R., Ravanbakskh, M., Shaifat, F., Harvey, E. S., Mian, A., Seager, J. W., Culverhouse, P. F., Cline, D. E., Edgington, D. R.: A review of techniques for the identification and measurement of fish in underwater stereo-video image sequences (2013) doi: 10.1117/12.2020941
20. Tan, M., Pang, R., Le, Q. V.: EfficientDet: Scalable and efficient object detection (2020) doi: 10.48550/ARXIV.1911.09070
21. Tanaka, T., Ikeda, R., Yuta, Y., Tsurukawa, K., Nakamura, S., Yamaguchi, T., Komeyama, K.: Annual monitoring of growth of red sea bream by multi-stereo-image measurement. Fisheries Science, vol. 85, no. 6, pp. 1037–1043 (2019) doi: 10.1007/s12562-019-01347-7
22. Wang, C. Y., Liao, H. Y. M., Yeh, I. H., Wu, Y. H., Chen, P. Y., Hsieh, J. W.: CSPNet: A new backbone that can enhance learning capability of CNN (2019) doi: 10.48550/ARXIV.1911.11929
23. Zhang, Z.: A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330–1334 (2000) doi: 10.1109/34.888718