

**OBTENCIÓN DE CAMPO DE VELOCIDADES EN RÍOS MEDIANTE EL  
REGISTRO DE IMÁGENES DIGITALES -VIDEOS-  
(VeloRIVi: Velocities in Rivers using Image Visualization)**

**YEBEL NUÑEZ C.<sup>1</sup>  
THOMAS MAEDLER<sup>2</sup>  
HERNÁN ALCAYAGA<sup>2</sup>  
ALONSO PIZARRO<sup>2</sup>**

**RESUMEN:**

Las técnicas de velocimetría por imágenes apoyan las redes de medición estándar y amplían la información hidrológica e hidráulica de los ríos en las cuencas que no están densamente instrumentadas o tienen una accesibilidad limitada. Por esta razón, el monitoreo del caudal de los ríos es esencial para muchas aplicaciones hidráulicas e hidrológicas relacionadas con la gestión de los recursos hídricos y la hidrodinámica. Durante los últimos años varios métodos basados en imágenes para medir la velocidad de la superficie en los ríos han sido probados en flujos permanentes. Sin embargo, estos métodos requieren de un post procesamiento exhaustivo y de grandes recursos computacionales (*e.g.*, una importante capacidad de almacenamiento de datos e imágenes). El objetivo de este estudio es presentar un nuevo algoritmo escrito en Python para el análisis, exploración y obtención de la velocidad superficial en ríos para diferentes condiciones hidráulicas. El proceso busca optimizar los recursos computacionales simplificando el cálculo de las velocidades. Se busca minimizar la operación humana en cada proceso y también disminuir el uso de grandes recursos computacionales, reduciendo trabajo post procesamiento.

Se realizó un primer experimento de bajo costo, mediante el uso de una cámara en un sistema raspberry pi y celulares. En este experimento se identificaron trazadores flotantes distribuidos y las velocidades estimadas fueron comparadas con las mediciones realizadas mediante un correntómetro. Los resultados mostraron que la metodología aplicada y el algoritmo desarrollado, llamado VeloRIVi, arrojan porcentajes de error entre 5,6 – 30,3%. Se espera disminuir el error inicial mediante la detección del centroide de cada trazador.

---

<sup>1</sup>Estudiante Escuela de Ingeniería en Obras Civiles, Universidad Diego Portales - email: yebel.nunez@udp.cl

<sup>2</sup>Profesor, Escuela de Ingeniería en Obras Civiles, Universidad Diego Portales. Ejercito 441, Santiago

## 1. INTRODUCCIÓN

Los recursos hídricos son parte fundamental para el desarrollo de un país y de la vida humana, por esta razón su cuantificación y uso debe ser constantemente monitoreado. En este contexto, profesionales relacionados con los recursos hídricos (*e.i.* ingenieros e hidrólogos) requieren de herramientas fiables y económicamente viables para supervisar y gestionar dichos recursos. Lo anterior es crítico en lugares donde el acceso es complejo o en situaciones donde se puede poner en riesgo la vida de los operadores (*e.i.* durante crecidas de los ríos). En el último tiempo se han realizado diferentes investigaciones que abarcan dicha problemática. Dentro de los avances más destacados están los algoritmos *Kanade–Lucas Tomas Image Velocimetry* (KLT-IV), *Large-Scale Particle Image Velocimetry* (LSPIV), *Large-Scale Particle Tracking Velocimetry* (LSPTV), *Optical Tracking Velocimetry* (OTV), *Space- Time Image Velocimetry* (STIV), entre otras (Fujita et al., 2007; Pearce et al., 2020). Con el uso de los *Unmanned Aerial System* (UAS) combinados con técnicas de velocimetría de imágenes proporcionan una alternativa significativa de bajo costo para el monitoreo fluvial o en canales, permitiendo la estimación de los caudales y de las velocidades de los flujos superficiales a partir de adquisiciones de vídeo (Manfreda et al., 2019). La precisión de estos métodos tiende a ser sensible a varios parámetros, como la presencia de materiales flotantes masivos, las condiciones ambientales adversas (velocidad del viento) y la densidad de siembra de los trazadores, la cual no es constante durante el periodo de adquisición de imágenes o videos, e incluso el conocimiento del operador post toma de datos en terreno.

En este contexto, el objetivo principal de este trabajo es desarrollar un algoritmo que permita obtener el campo de velocidades superficiales en ríos, reduciendo tiempos de post procesamiento. Para cumplir este objetivo se desarrolla un nuevo algoritmo de velocimetría por imágenes basado en LSPTV.

Para ello se realiza un exhaustivo análisis y procesamiento de videos e imágenes para determinar la velocidad superficial del flujo mediante un experimento in-situ. La metodología aplicada y algoritmo construido, permite identificar la velocidad superficial a través de la detección de cada trazador obteniendo su trayectoria en cada fotograma ortorectificado (corregidas por distorsión geométrica) consecutivo y separadas por un intervalo de tiempo conocido (Pearce et al., 2020).

## 2. METODOLOGÍA

Para desarrollar el algoritmo, basado en LSPTV, se usó el lenguaje de programación Python, empleando diferentes librerías para crear un código que permita estimar las velocidades superficiales para escurrimientos en canales o cauces naturales. La metodología se divide en tres grandes partes secuenciales, las cuales se mencionan a continuación:

- a) Proceso de ortorectificación
- b) Procesamiento y análisis de los fotogramas
- c) Cálculo de la velocidad superficial



En terreno se ubicaron seis puntos de control identificadas mediante dianas (Figura 2) formando una sección rectangular. El algoritmo requiere que la sección de interés sea rectangular para facilitar su ortorectificación y procesamiento, minimizando la propagación de errores en la estimación de velocidades debido a la deformación geométrica de las imágenes. Para los trazadores se escogieron diferentes materiales biodegradables, como hojas de papel, hojas de árboles e incluso pequeñas ramas de árboles. Estos fueron distribuidos manualmente en toda la sección trasversal del río. En la Figura 3 se muestra la sección transversal aforada mediante correntómetro, y las secciones donde se realizaron las mediciones observadas, que son sujeto de estudio para comparar con las velocidades estimadas mediante el algoritmo desarrollado, llamado VeloRIVi.

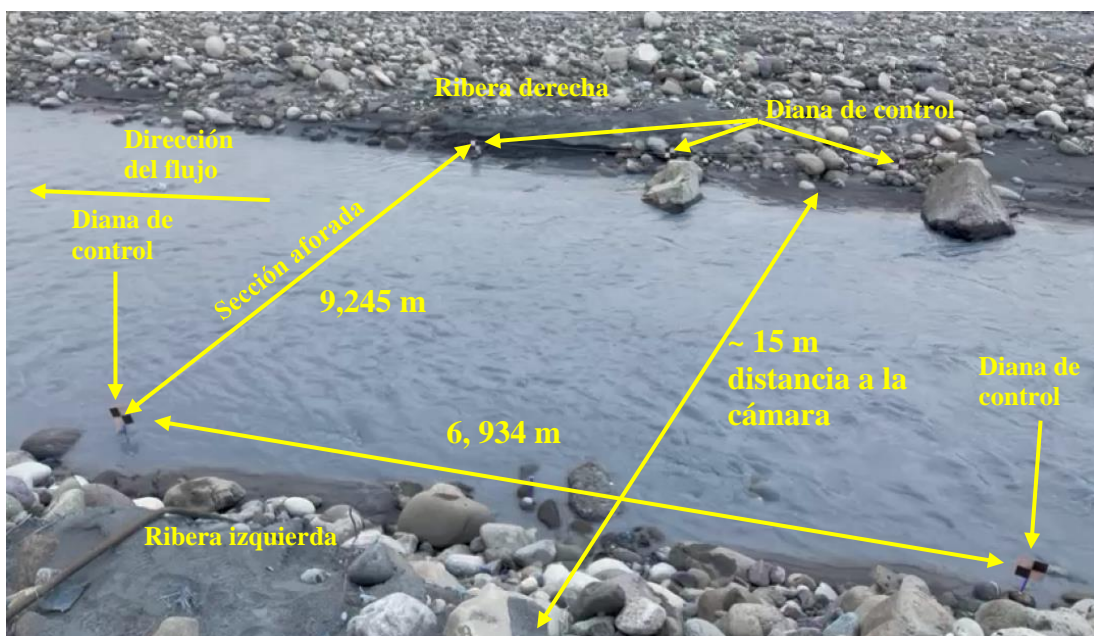


Figura 2. Tramo de interés del río Mapocho, sector Lo Cañas, comuna de Viticura.

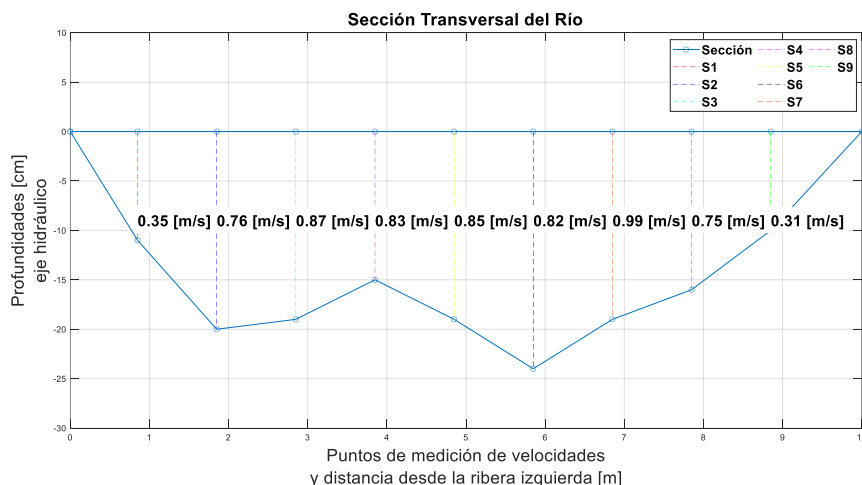


Figura 3. Sección transversal del río Mapocho. 09-08-2023 10:33

## 2.2. VELOCIMETRIA POR IMÁGENES

### a) PROCESO DE ORTORECTIFICACIÓN

Se realizó una reconstrucción del terreno, para poder efectuar la ortorectificación de las imágenes en la zona de interés, su posterior procesamiento y finalmente el cálculo de la velocidad superficial. Para ello se aplicaron los siguientes pasos:

- i) Geolocalización de los puntos de control en un plano cartesiano en Python usando las distintas mediciones de las distancias entre los seis puntos de control establecidos mediante dianas identificables en las imágenes.
- ii) A partir de cada punto se realizó un trazado de circunferencias (usando como centro el punto de control) que contengan como diámetro las medidas tomadas en terreno para poder determinar, con respecto a un solo punto de control, las coordenadas de los demás puntos de control restantes.
- iii) Una vez ubicados los seis puntos de control con sus coordenadas respectivas en un plano cartesiano, se trazan vectores entre ellos para formar un polígono perimetral. A partir de ello, se realizaron cálculos de la dirección vectorial (Ec.1) y el producto punto entre vectores (Ec.2) para verificar la ortogonalidad de las nuevas rectas en las que se genera un cuadrilátero perfecto en caso de que en terreno fuera imposible generar una figura geométrica rectangular.
- iv) Con la ecuación 1 se determina un parámetro  $t$  el cual genera estas nuevas coordenadas (los vectores son ortogonales) que forman un rectángulo. Dicho parámetro es una relación entre la distancia de dos puntos de control (coordenadas conocidas) y otra entre el nuevo punto generado por la sección rectangular y un punto de control (punto de intersección). Esta relación se denomina parámetro  $t$ , donde  $t = \frac{\sigma}{\tau}$  (Figura 4).

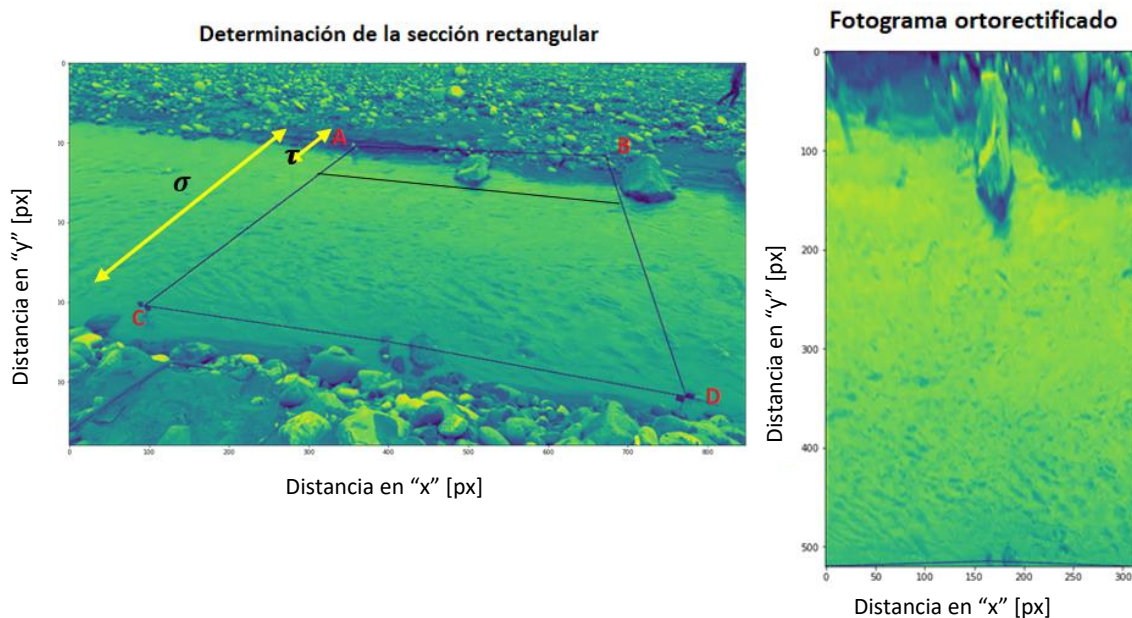


Figura 4. Formación zona rectangular para el proceso de ortorectificación del video.

La finalidad de la reconstrucción y ubicación de los puntos es realizar un proceso inverso para la ortorectificación. Es decir, poder trazar mediante los puntos de control un rectángulo visto desde arriba (en planta) para luego con los fotogramas originales (con un ángulo de inclinación) establecer una figura geométrica rectangular, en caso de que las condiciones en terreno no permitan que los puntos de control sean ubicados de tal forma que formen un rectángulo. Esta verificación es necesaria debido a que la función de Python usada, llamada `cv2.warpPerspective`, requiere que el área de interés sea rectangular para poder realizar la ortorectificación. Por esta razón, en terreno se debe verificar que los lados paralelos entre los puntos de control sean iguales.

Con el parámetro  $t$  o punto de intersección encontrado con la reconstrucción de los puntos de control en terreno, se pueden determinar las coordenadas que trazan la sección rectangular en cada fotograma del video, siempre y cuando el video sea estable. De esta manera se obtiene la ortorectificación del video.

$$P(t) = (P_i - P_0) * t + P_0 \quad (1)$$

$$\overrightarrow{R(t)} \perp \overrightarrow{P(t)} \Rightarrow \vec{R} \cdot \vec{P} = 0 \quad (2)$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = M(\alpha) * \begin{bmatrix} x \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \quad (3)$$

Para las ecuaciones anteriores,  $P(t)$  es un punto con coordenadas  $[x(t), y(t)]$  de una línea de píxeles en función del parámetro  $t$ .  $P_i$  y  $P_0$  corresponden a dos puntos diferentes,  $P_i$  es el punto final y  $P_0$  es el punto de inicio en el que se interpola,  $t$  es un parámetro que puede tomar valores entre 0 y 1, donde 0 representa el punto de inicio  $P_0$  y 1 representa el punto final  $P_i$  de las rectas correspondiente al punto de intersección entre 2 rectas ortogonales. La ecuación 2 determina la condición de ortogonalidad entre los vectores  $R$  y  $P$  definido entre los diferentes puntos de control perimetrales. En el caso que no se cumpla con la ortogonalidad, se utiliza el sistema de ecuaciones (Ec.3) y a través de una de rotación (matriz  $M$ ), se rota el punto en un plano 2D por un ángulo  $\alpha$  en sentido antihorario, se encuentra las nuevas coordenadas de los puntos  $X'$  y  $Y'$  correspondiente a las nuevas coordenadas de los puntos de control, ahora de la imagen ortorectificada.

En la figura 5 se muestra un diagrama esquemático que resume el proceso de ortorectificación en caso de que la posición de las dianas cumpla con los requerimientos de la función llamada `cv2.warpPerspective` o en caso de que no lo cumplan.

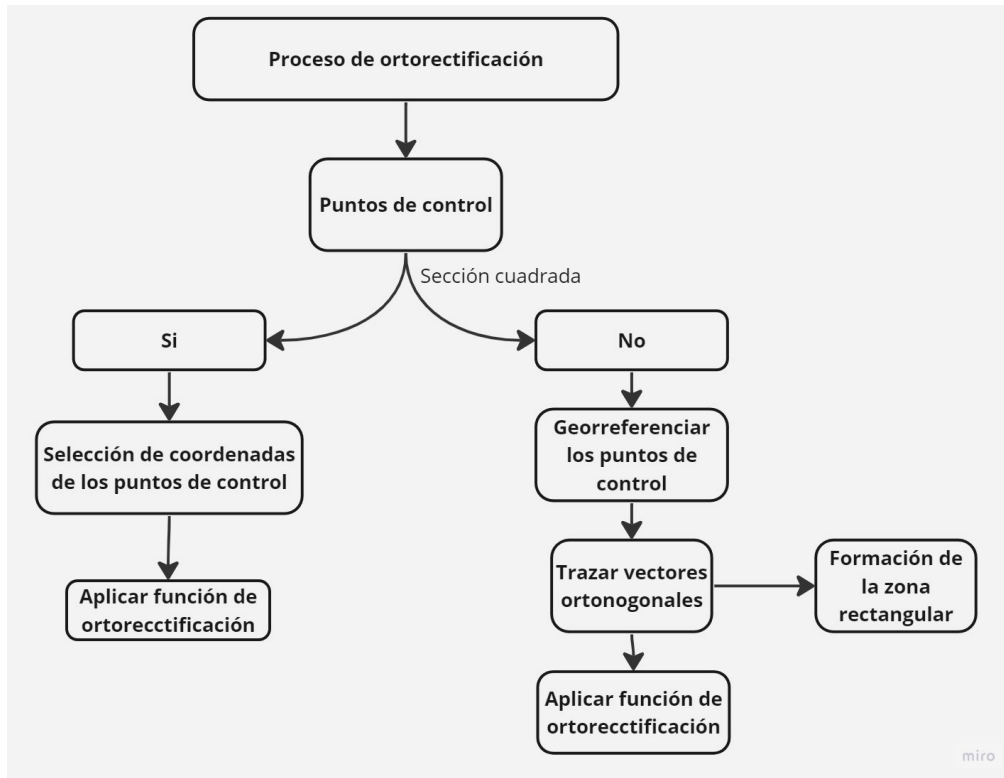


Figura 5. Diagrama del proceso de ortorectificación del video.

## b) PROCESAMIENTO DE IMÁGENES

Una vez realizado el proceso de ortorectificación se usaron diferentes técnicas para reducir ruido visual en la superficie del agua, es decir elementos que no son necesarios para el cálculo de la velocidad, destacando solo los trazadores. Para ello se requiere analizar las imágenes en escala de grises, lo que permite utilizar el histograma de intensidades.

Se seleccionó esta metodología debido a que usa un enfoque automatizado, evitando la introducción de sesgo. El proceso consiste en clasificar los píxeles de una imagen en dos niveles de intensidad, lo que facilita la extracción de información relevante de la imagen y la identificación de objetos (trazadores).

Mediante el histograma de intensidades se determina el umbral óptimo, comprendido entre 0 y 255, del cual posteriormente se obtiene una imagen booleana, con valores de 1 (negro) y 0 (blanco) correspondiente a los trazadores elementos de objeto de estudio para la velocidad superficial, calculando un umbral local para todos los fotogramas del video y de esa manera discriminar con mayor eficiencia los píxeles de interés de aquellos que no lo son.

### c) CÁLCULO DE LA VELOCIDAD SUPERFICIAL

Después de ortorectificada y procesada la imagen es decir obteniendo para cada fotograma información relevante (identificación de los trazadores), se aplica la ecuación 4 de la velocidad. Para la aplicación de esta ecuación de debe conocer la posición de cada trazador para cada imagen consecutiva. Con la posición de los trazadores en cada imagen, se determina la longitud que recorre cada uno. Además, es conocido el tiempo trascurrido entre la captura de cada fotograma.

$$V = \frac{L \cdot \beta}{t_{i+1} - t_i} \quad (4)$$

Donde V es la velocidad superficial, L corresponde a la longitud recorrida por cada partícula entre los instantes  $t_i$  y  $t_{i+1}$  (tiempo entre fotogramas) y  $\beta$  factor de conversión entre pixeles y metros.

El la Figura 6 se presenta un diagrama esquemático que resume la metodología propuesta en VeloRiVi.

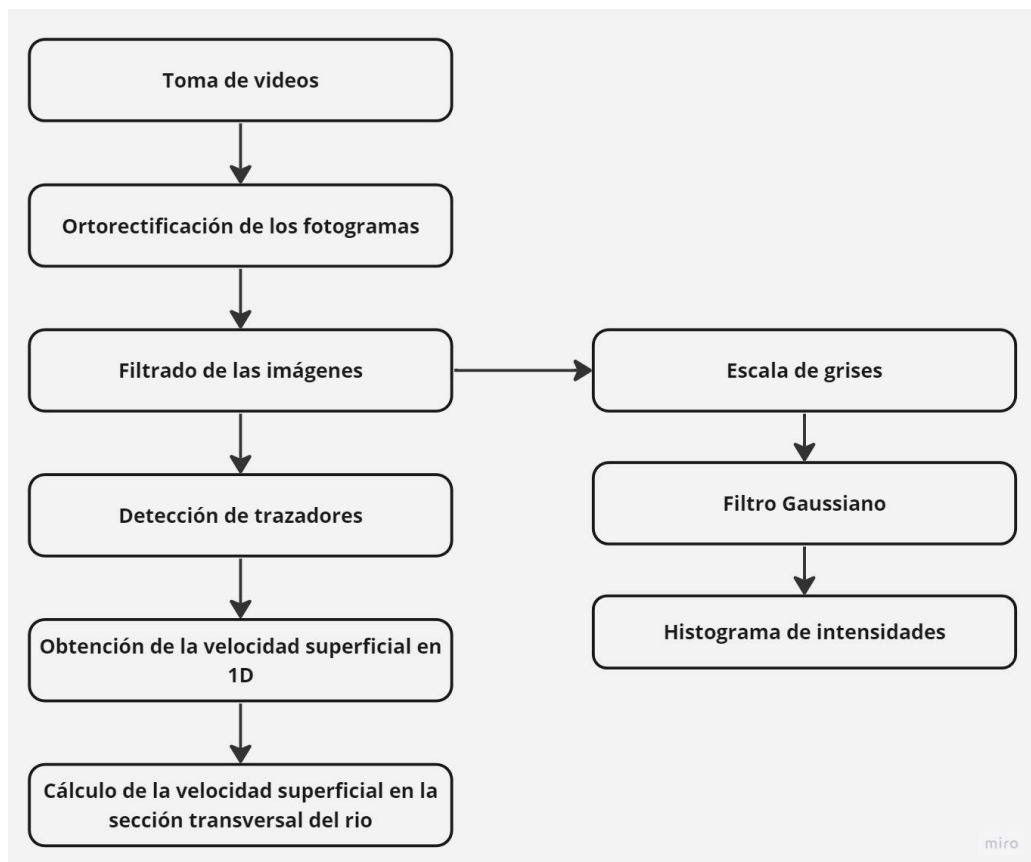


Figura 6. Diagrama de la metodología de VeloRiVi y el proceso de filtrado para la detección de trazadores.

### 3. RESULTADOS

Para la metodología antes mencionada, se obtuvieron los resultados presentados a continuación, aplicando el algoritmo VeloRIVi.

#### a) ORTORECTIFICACIÓN DE LA IMAGEN

Los resultados de la ortorectificación utilizando la geolocalización de cada punto de control se muestran en la Figura 7. En este proceso se corrige la distorsión de la perspectiva del video para que los trazadores a detectar tengan las dimensiones correctas. Debido a que el video fue tomado desde una perspectiva inclinada se requiere este proceso para evitar errores en el cálculo de la distancia que recorre cada trazador.

Para este caso de estudio no fue necesario realizar un proceso previo (generación de la zona rectangular) para la ortorectificación, puesto que los puntos de control estaban posicionados de tal forma que cumplían con la zona rectangular requerida por la función de python para la ortorectificación.

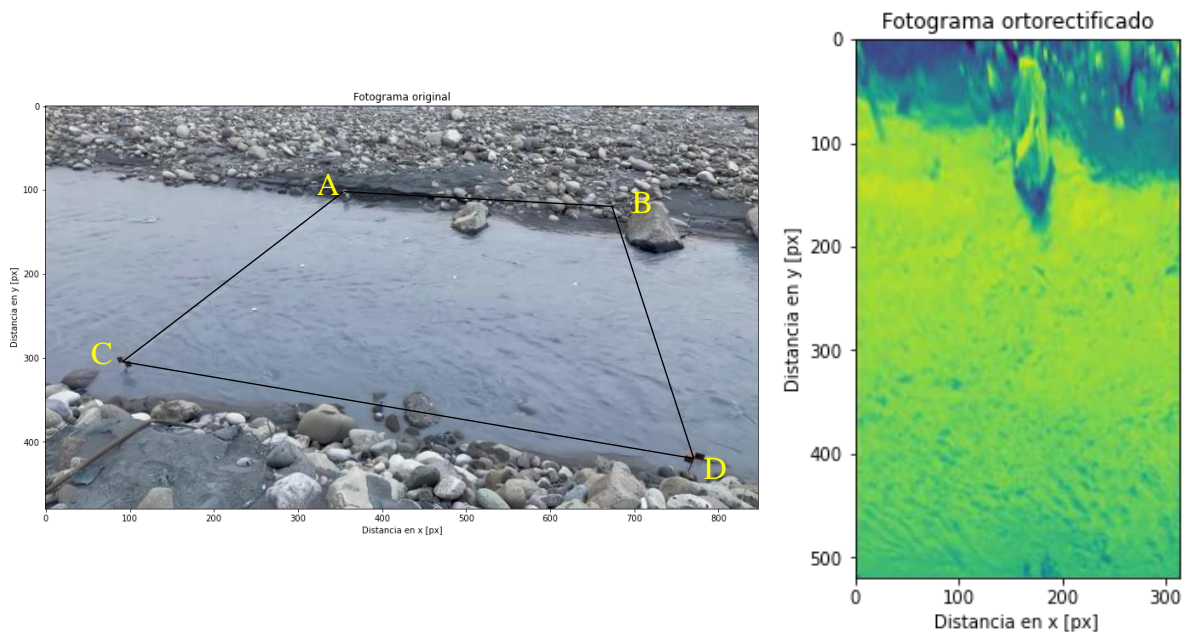


Figura 7. Imagen en bruto (sin ortorectificar) y determinación de sección rectangular a ortorectificar ABCD, a la izquierda e imagen ortorectificada, a la derecha.

## b) PROCESAMIENTO DE IMAGEN

A la imagen ortorectificada se le aplica la umbralización, la cual ajusta la intensidad en cada pixel de los fotogramas, destacando solo aquellos donde se encuentran los trazadores. Sin embargo, existe un bajo porcentaje de ruido visual aceptable, el cual se espera en el futuro una eliminación total de este. El proceso de filtrado es esencial ya que permite un mejor cálculo de la velocidad superficial. El resultado de la aplicación de umbralización según el histograma de intensidades se puede ver en la figura 9. Para los trazadores identificados posteriormente se les determina la distancia que recorren entre fotogramas y el tiempo que tardan en recorrer desde el fotograma  $i$  hasta el fotograma  $i+1$ .

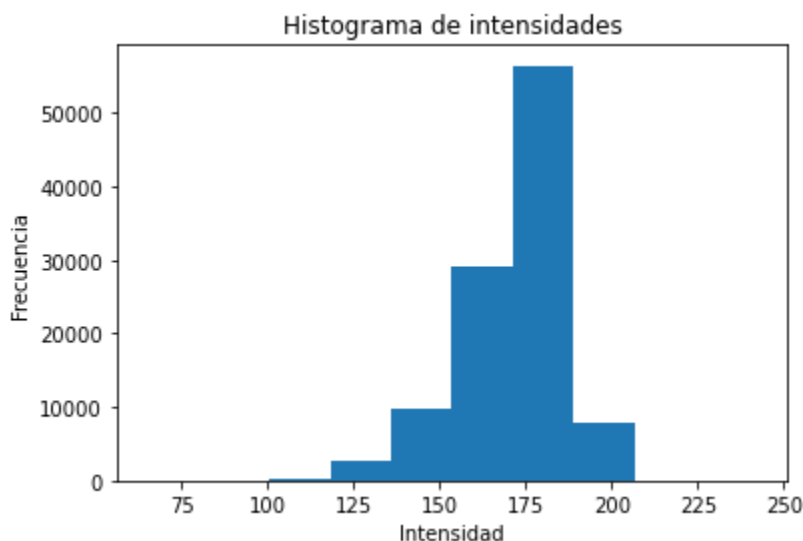


Figura 8. Histograma de intensidades de los fotogramas (blanco y negro)

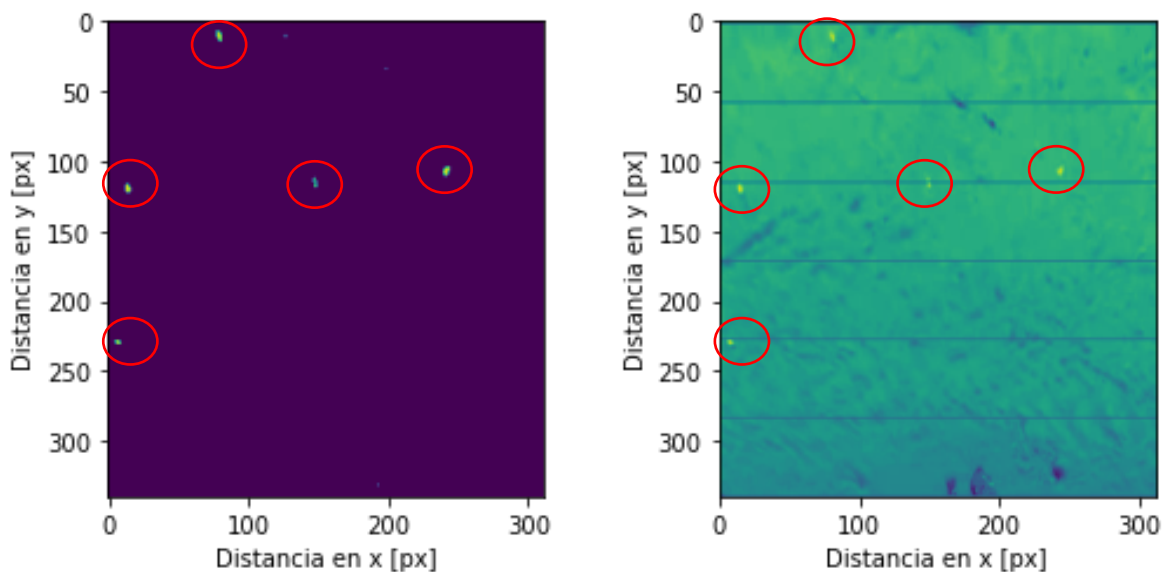


Figura 9. Fotograma antes del filtrado y las secciones de medición, a la derecha. Fotograma filtrado con detección de la intensidad de los trazadores, a la izquierda.

### c) CÁLCULO DE LA VELOCIDAD SUPERFICIAL

Las velocidades estimadas por el algoritmo VeloRIVi en algunas de las secciones de la región de interés se presentan en la Tabla 1. Estas velocidades estimadas fueron determinadas según las condiciones del video y la distribución de los trazadores.

Tabla 1. Velocidades superficiales estimadas usando VeloRIVi en diferentes puntos de la sección transversal.

Distancia desde ribera izquierda (m)	Velocidades superficiales VeloRIVi (m/s)
0,85	-
1,85	0,99
2,85	0,82
3,85	0,74
4,85	1,06
5,85	0,85
6,85	0,92
7,85	0,83
8,85	-

### d) VERIFICACIÓN DE LAS VELOCIDADES ESTIMADAS

Las velocidades medidas mediante el correntómetro en los nueve puntos, con sus cinco replicas se presentan en la Tabla 2, junto al promedio y desviación estándar.

Tabla 2. Velocidades superficiales medidas con correntómetro en diferentes puntos de la sección.

Distancia desde ribera izquierda (m)	Medición 1 (m/s)	Medición 2 (m/s)	Medición 3 (m/s)	Medición 4 (m/s)	Medición 5 (m/s)	Promedio (m/s)	Desviación estándar (m/s)
0,85	0,34	0,34	0,34	0,34	0,34	0,40	0,003
1,85	0,70	0,76	0,76	0,82	0,76	0,76	0,007
2,85	0,82	0,91	0,82	0,85	0,91	0,87	0,009
3,85	0,76	0,82	0,82	0,82	0,91	0,83	0,012
4,85	0,82	0,82	0,82	0,91	0,85	0,85	0,006
5,85	0,76	0,82	0,85	0,82	0,82	0,82	0,004
6,85	0,91	0,91	1,10	1,04	0,98	0,91	0,025
7,85	0,70	0,70	0,82	0,76	0,76	0,75	0,010
8,85	0,34	0,34	0,34	0,27	0,27	0,31	0,004

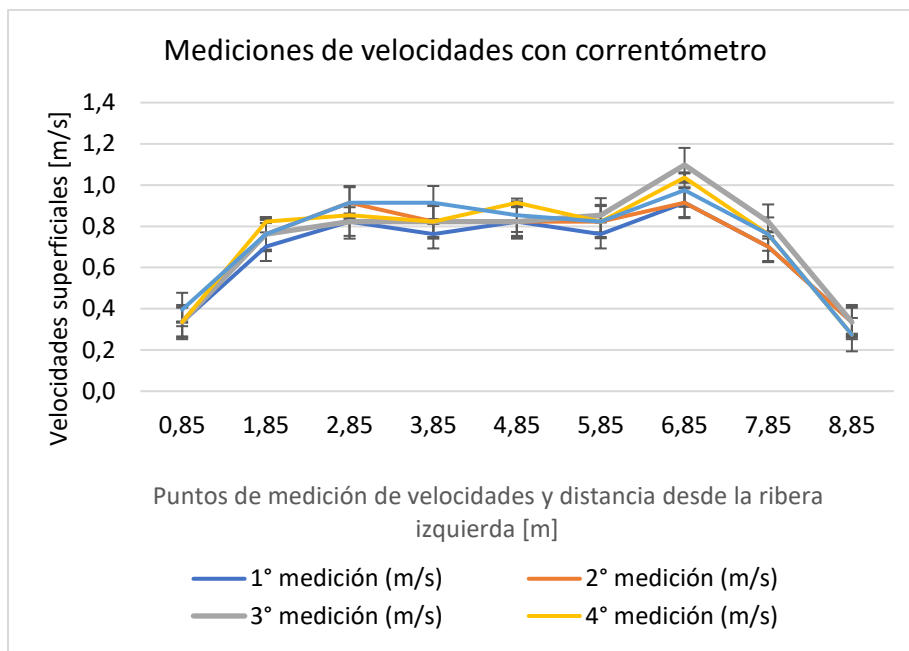


Figura 8. Variación de una serie de mediciones de velocidades superficiales en sus nueve puntos.

A continuación, en la Tabla 3 se muestra el porcentaje de error entre las mediciones realizadas en terreno y las procesadas por el algoritmo VeloRIVi. Además, se puede apreciar en la tabla 3, que si bien el error en algunos casos es alto (30,3%), exista una buena correlación (patrón) entre las mediciones realizadas con correntómetro y las estimadas mediante VeloRIVi. Lo anterior indica que puede existir un error sistemático en alguno de los procesos de cálculo y en algunas zonas cercanas a las orillas del río, donde no se pudo determinar la velocidad superficial, debido a que los trazadores no recorrieron dichas zonas.

Tabla 3. Comparación de velocidades superficiales medidas con correntómetro y VeloRIVi en diferentes los nueve puntos del área de interés

Profundidad (cm)	Distancia desde ribera izquierda (m)	Promedio de velocidades superficiales con correntómetro (m/s)	Velocidades superficiales VeloRIVi (m/s)	Error (%)
11	0,85	0,35	-	-
20	1,85	0,76	0,99	-30,3%
19	2,85	0,87	0,82	5,2%
15	3,85	0,83	0,74	10,3%
19	4,85	0,85	1,02	-20,9%
24	5,85	0,82	0,85	-4,6%
19	6,85	0,99	0,92	6,7%
16	7,85	0,75	0,83	-10,5%
10	8,85	0,31	-	-

#### 4. CONCLUSIONES

Los resultados obtenidos con la metodología desarrollada y la aplicación del algoritmo VeloRIVi evidenciaron la efectividad y un buen desempeño de ella, sin embargo, aún se debe mejorar su eficacia para obtener velocidades superficiales de tal forma que puedan ser comparables con la de un correntómetro tradicional. Se distinguieron de forma efectiva los objetos o trazadores a los cuales se desea hacer seguimiento, permitiendo visualizar la trayectoria y patrones de movimiento que necesitan destacar. Esto permitió poder llevar a cabo un posterior análisis para la determinación de la velocidad.

El proceso de filtrado muestra un buen comportamiento y aporta gran certidumbre en los resultados. Sin embargo, en el futuro se pueden aplicar otras funciones o metodologías para el procesamiento de los fotogramas que permitan depurar más detalladamente información no valiosa (reducción de ruido). También se pueden hacer uso de otras librerías como YOLO (Jiang et al., 2022), entre otras las que permiten detectar el movimiento de objetos.

Según los objetivos propuestos, se logró la automatización del código, ahorrando tiempo, capacidad de los computadores y minimizando errores humanos. Sin embargo, se espera que el algoritmo arroje resultados más certeros antes de que pueda ser implementado.

#### REFERENCIAS

- Fujita, I., Watanabe, H., & Tsubaki, R. (2007). Development of a non-intrusive and efficient flow monitoring technique: The Space-Time Image Velocimetry (STIV). *International Journal of River Basin Management*, 5, 105-114.  
<https://doi.org/10.1080/15715124.2007.9635310>
- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, 199, 1066-1073.  
<https://doi.org/10.1016/j.procs.2022.01.135>
- Manfreda, S., Sasso, S. F. D., Pizarro, A., & Tauro, F. (2019). New Insights Offered by UAS for River Monitoring. *Applications of Small Unmanned Aircraft Systems*, 211-234. <https://doi.org/10.1201/9780429244117-10>

Pearce, S., Ljubicic, R., Peña-Haro, S., Perks, M., Tauro, F., Pizarro, A., Dal Sasso, S. F., Strelnikova, D., Grimaldi, S., Maddock, I., Paulus, G., Plavšic, J., Prodanovic, D., & Manfreda, S. (2020). An evaluation of image velocimetry techniques under low flow conditions and high seeding densities using unmanned aerial systems. *Remote Sensing*, *12*(2). <https://doi.org/10.3390/rs12020232>