

SOFTWARE PARA LA DETECCIÓN DE VARIABLES HIDRODINÁMICAS EN LABORATORIO MEDIANTE TÉCNICAS DE VIDEO DE ALTA DEFINICIÓN

Agustín D. Pérez Paladini^a, Paolo Gyssels^b, Oscar H. Bustos^a y Mariana Pagot^b

^a *Facultad de Matemática Astronomía y Física, Universidad Nacional de Córdoba, Medina Allende
s/n, 5000 Córdoba, Argentina,
adperez@famaf.unc.edu.ar, bustos@mate.uncor.edu, <http://www.famaf.unc.edu.ar>*

^b *Laboratorio de Hidráulica LH, Facultad de Ciencias Exactas, Físicas y Naturales, Universidad
Nacional de Córdoba, Av. Filloy s/n, 5000 Córdoba, Argentina.
pgyssels@isrg.unc.edu.ar, <http://www.efn.uncor.edu/investigacion/hidraulica/>*

Palabras Clave: Procesamiento de imágenes, Altura y Periodo de olas, Tiempo Real, CPU, GPU.

Resumen. En este trabajo se presenta un sistema de detección de altura y período de olas en tiempo real por medio de video cámaras en el canal de oleaje bidimensional instalado en el Laboratorio de Hidráulica de la Universidad Nacional de Córdoba. El proceso usa cámaras de bajo costo y alta resolución, representando un método de medición no intrusivo. La precisión de la altura de ola y el periodo medido son comparados con los instrumentos tradicionales como el sensor resistivo. Los errores estimados de las medidas realizadas se encuentran en el orden de los 0,1 cm para la altura de la ola y 0,009 segundos para el periodo de la ola.

Una interfaz amigable ha sido desarrollada para configurar el sistema y procesar los datos obtenidos. El programa desarrollado es flexible ya que tiene el potencial de manipular los parámetros usados en los algoritmos y ha sido implementado en un entorno que permite crear futuras aplicaciones (para medir otras variables hidrodinámicas o sedimentarias por ejemplo) de una manera fácil y rápida, dando la oportunidad de utilizar la Unidad de Procesamiento Grafico (GPU por sus siglas en inglés).

1 INTRODUCCIÓN

En la modelación física de laboratorio se emplean en muchos casos instrumentos costosos para la medición de variables hidrodinámicas tales como velocidades, variaciones de niveles, presiones, trayectorias de partículas, turbulencia, entre otras.

Algunos instrumentos modernos de medición en laboratorio, como los que utilizan las técnicas de *Particle Image Velocimeter* (PIV), requieren de una experiencia y mucha precaución por parte del usuario principalmente por el uso del sensor láser. Otros instrumentos utilizados generalmente son los de tipo intrusivo, como el *Acoustic Doppler Velocimeter* (ADV), o los sensores de nivel tipo resistivo, que se sumergen directamente dentro del fluido, produciendo una interacción con el mismo.

En general, estos sensores son específicos para la medición de las variables en un cierto rango de escala espacio-temporal, en algunos casos se utilizan solamente en ensayos bidimensionales, y presentan limitaciones de uso tales como: determinan la variable medida en un volumen de fluido definido, no describen los procesos cerca de los contornos, dependen de la siembra de partículas en el fluido, se corroen si el agua es salada.

En este trabajo se desarrolla un sistema de medición y análisis de variables hidrodinámicas, principalmente de altura y periodo de la ola, en laboratorio mediante técnicas de vídeo con cámaras web de alta definición. Estas técnicas tienen diferentes ventajas con respecto a los métodos e instrumentos tradicionales: son técnicas no intrusivas, económicas, son fácilmente utilizables por usuarios con poca experiencia en trabajos de laboratorio, y son muy flexibles porque se pueden aplicar a diferentes tipos de mediciones y a diferentes escalas espacio-temporales de los procesos.

La ventaja principal de los instrumentos tradicionales es la elevada precisión de la medición que tienen que realizar. Sin embargo en aplicaciones ingenieriles de laboratorio, una webcam de alta definición estándar puede ser suficiente para el nivel de detalle requerido.

Las aplicaciones de estas técnicas en laboratorio son múltiples en los campos de la mecánica de fluidos y de la dinámica sedimentaria, entre otros. En este trabajo se implementó una técnica para la adquisición de datos de altura de ola en tiempo real y su posterior análisis. La metodología empleada es factible de aplicación para las mediciones de otros parámetros hidrodinámicos de interés como rotura de oleaje, dispersión de contaminante y sedimentológicos.

2 TÉCNICAS BASADAS EN PROCESAMIENTO DE IMÁGENES

El conjunto de técnicas investigadas en este trabajo se basa en la utilización de cámaras que siguen el modelo *pinhole* o *pinhole model* (Pérez et al., 2012).

Dentro del conjunto de técnicas basadas en procesamiento de imágenes se pueden diferenciar aquellas que son en tiempo real y las que no lo son, pero ambas, a grandes rasgos siguen un mismo proceso. Este proceso, que tiene como entrada una imagen y como salida la detección de alguna variable determinada (en este caso la altura de la superficie libre) podemos describirlo, de forma general, como un conjunto de pasos (secuenciales) o subprocesos: rectificación, detección de contornos, análisis y procesamiento.

Cada uno de estos pasos pueden ser llevados a cabo de distintas maneras y utilizando diferentes algoritmos o modelos matemáticos.

Algunas de las técnicas analizadas en este trabajo se pueden encontrar en Brady et al., 2004 y Sánchez, 2010.

2.1 Rectificación

Para lograr una correcta medición de la altura de ola (u otra variable hidrodinámica de interés) es necesario trabajar sobre una imagen libre (o al menos con la menor cantidad posible) de errores. Podemos diferenciar 2 tipos de errores o problemas: a) aquellos

introducidos por el dispositivo de captura (cámara de vídeo), también conocidos como distorsiones (Pérez et al., 2012 y Bradski et al., 2008); b) aquellos relacionados con la posición y orientación de la cámara.

Con respecto a estos últimos (b), el problema de mayor importancia es el de la diferencia que existe entre el sistema de coordenadas entre la cámara y el plano sobre el cual se quiere realizar las mediciones. En la Figura 1 se muestra el plano de la zona a analizar (“*object plane*”) que en este trabajo representaría la pared del canal donde se realiza la captura de imagen para detectar la altura de la ola), se representa dentro de la imagen capturada por el dispositivo (“*image plane*”) con cierta perspectiva. Esto es un problema debido a que, como el objeto de estudio es la altura de la superficie libre (aunque se aplique en general a cualquier otra variable a medir), la altura real de la misma que se observa en el canal se encuentra proyectada en la imagen y con una perspectiva diferente a la real, esto es, con un sistema de coordenadas diferente. Es por esto que se debe encontrar la relación entre estos 2 sistemas de coordenadas (por medio de lo que se denomina “transformación de la perspectiva”) para poder determinar la correspondencia de cada punto de la imagen a su respectivo punto en la pared del canal.

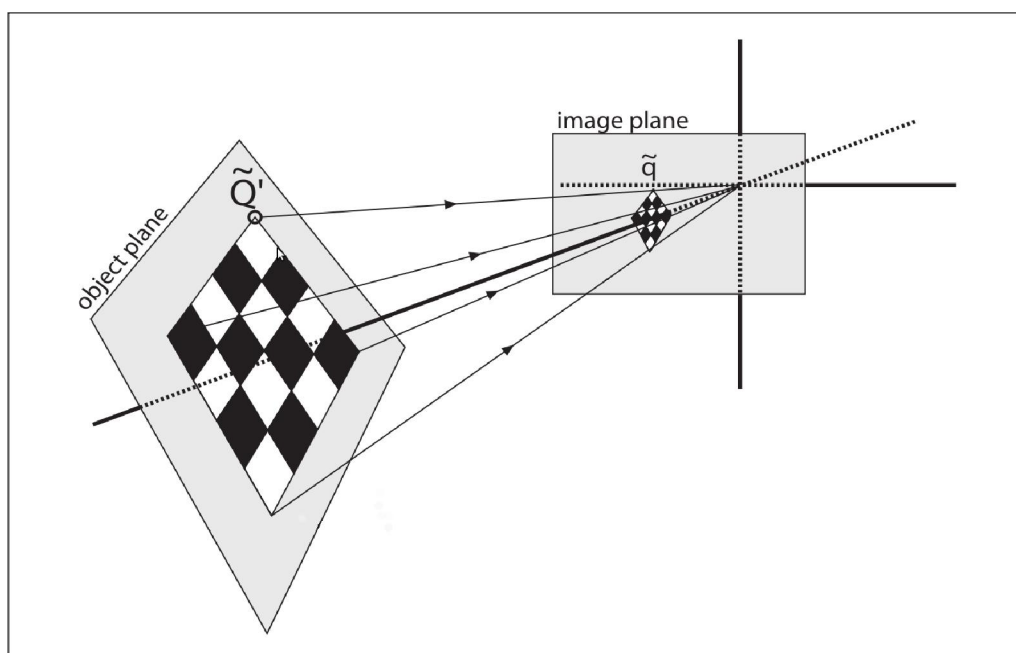


Figura 1: Diferencia de perspectiva entre la imagen en la cámara (*image plane*) y el plano real (*object plane*).
(Fuente: Bradski et al., 2008)

Para llevar a cabo la rectificación, se utilizan diferentes métodos, entre ellos el modelo de Transformada Lineal Directa (DTL), solución por mínimos cuadrados (Pérez et al., 2012), y métodos alternativos.

2.2 Detección de contornos

Esta etapa de trabajo tiene como fin detectar los bordes asociados a la superficie libre que se encuentran en la imagen de entrada, minimizando la detección de bordes pasivos (aquellos que pertenecen al entorno y no a la superficie libre en sí).

En una primera instancia, se procede a disminuir la complejidad de la imagen reduciendo la cantidad de canales (colores) de la misma, en general, convirtiéndola a una imagen monocromática.

En segundo lugar, se aplican una serie de filtros (o máscaras) con el fin de delimitar la zona que se quiere analizar, evitando procesar zonas que no son de interés y a su vez,

eliminando posibles errores causados por falsos valores (esto es, los bordes falsos mencionados anteriormente).

Por último se realiza la detección de contornos sobre la imagen filtrada por medio de algún algoritmo. Existen varios métodos utilizados para detectar los bordes, algunos de estos se pueden ver en [Nadernejad et al., 2008](#).

En este trabajo se utilizó el algoritmo denominado *Canny Edge Detector* ([Canny, 1986](#)), el cual es uno de los algoritmos más utilizados y que da mejores resultados a la hora de encontrar bordes en una imagen.

Cabe indicar en este apartado que para lograr obtener resultados adecuados es necesario capturar imágenes con la mayor calidad posible, además de una buena resolución de imagen, es importante tener en cuenta varios factores del entorno donde se realizarán las muestras de laboratorio, como por ejemplo: luz ambiental, contraste de la superficie libre con respecto al fluido, suciedades en los vidrios laterales del canal, y otros tipos de ruidos asociados a zonas que pueden generar contornos que se confundan con la superficie libre.

2.3 Análisis y procesamiento

Una vez detectado los bordes de la imagen y en base a los algoritmos utilizados en la etapa anterior, se aplica cierta heurística para diferenciar los bordes correspondientes a la altura de la superficie libre (bordes activos), de aquellos que no corresponden a la variable que se quiere analizar (lo que se denomina ruido o bordes pasivos). La forma en la que se detectan estos bordes activos no es única y está fuertemente ligada en como fue modificada la imagen en la etapa anterior y/o en la forma de representar los contornos detectados.

Para este trabajo se desarrolló un algoritmo simple, para alcanzar tal objetivo, denominado “*Wave Height Analyzer*” (Analizador de Altura de Ola, ver Punto 4), (más aplicaciones en [Brady et al., 2004](#)).

Luego de lograr la detección de los bordes activos se procede a obtener la altura en metros (o alguna otra unidad de longitud) por medio de la transformación de la relación píxeles/metros (o la unidad de longitud correspondiente).

Podemos decir, en base a los trabajos analizados, que para esta etapa (Análisis y procesamiento) no existe algún lineamiento específico de cómo llevarla a cabo, si no que es la que mas varia (de trabajo en trabajo) y está sumamente relacionada con la forma en que fueron realizadas las etapas anteriores.

3 TÉCNICAS UTILIZADAS

En el presente trabajo podemos dividir el proceso general en cuatro sub-procesos como se muestra en el diagrama de la [Figura 2](#).

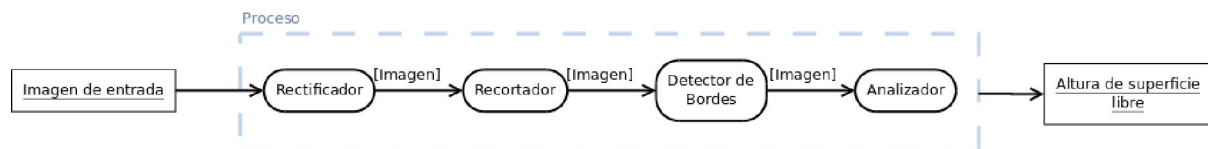


Figura 2: Diagrama general del proceso de la técnica utilizada.

Como se puede ver, el proceso (rectángulo azul) tiene como entrada una imagen que puede provenir de una cámara de vídeo o de un archivo. Dentro del proceso general se pueden diferenciar los cuatro subprocesos que trabajan de manera secuencial, donde cada uno de estos tiene como entrada la imagen procesada por el subproceso inmediatamente anterior.

3.1 Rectificador

Este subproceso se basa en 2 grupos de 4 puntos (bidimensionales) correspondientes a los puntos de origen y de destino (de la imagen de entrada y salida respectivamente). Los

primeros están asociados a la zona de trabajo que se quiere analizar. Mientras que el segundo grupo está relacionado con el tamaño de la imagen capturada por el dispositivo en sí (el tamaño del *frame*/imagen con el que se trabaja). El primer grupo de puntos necesita ser especificado por el usuario mientras que el segundo se asigna de manera automática (por el sistema).

Se genera una matriz de transformación de perspectiva por medio de los 8 puntos mencionados anteriormente, la cual mapea cada píxel de la imagen origen a su nueva posición en la imagen destino (solventando el desfase de los distintos sistemas de coordenadas).

De esta manera se soluciona el error de la perspectiva y al mismo tiempo se eliminan las zonas (a grandes rasgos) que no pertenecen al área de interés. Además de que es un método sencillo y fácil de configurar por parte del usuario.

Este subproceso se basa en el algoritmo (implementado) “*Perspective Transform*” (Transformación de Perspectiva, ver Punto 4).

3.2 Recortador

Este proceso realiza un recorte aún mayor sobre la imagen de entrada descartando aquellas zonas que no son de interés, con el fin de agilizar el procesamiento de posteriores subprocesos.

La existencia de este simple proceso se debe a que en este trabajo necesitamos detectar la altura de la superficie libre (y por ende realizar todo el proceso) en tiempo real minimizando el uso de memoria de la PC. Por otro lado, solo es necesario capturar la altura de la misma en un punto determinado (una columna de píxeles), aunque esta única columna no alcanza debido a que podría no detectarse el contorno de la superficie de forma correcta dando lugar a cálculos erróneos y un sistema muy propenso a fallas. Además, simplifica las configuraciones de los posteriores subprocesos.

3.3 Detector de bordes

Su objetivo es detectar los bordes, eliminando en lo posible la mayor cantidad de ruido, pero sin discernir entre bordes activos y bordes pasivos. Podemos distinguir la siguiente secuencia de acciones realizadas por este proceso:

1. Se convierte la imagen de entrada (en general color) en una imagen en blanco y negro, por medio de un primer filtro de imagen.
2. Se le aplica un segundo filtro denominado “*Gaussian Blur*” (Fisher et al., 2003) con el fin de “suavizar” la imagen suprimiendo la mayor cantidad posible de ruido (como puntos aislados debido a partículas sobre la pared del canal) para una mayor diferenciación de los bordes reales.
3. Por último se corre el algoritmo denominado “*Canny Edge Detector*” utilizado para detectar los bordes en la imagen el cual se basa en ciertos parámetros previamente configurados por el usuario.

Al finalizar esta secuencia de pasos la imagen de salida pasa a ser una matriz de valores booleanos (donde un píxel de la imagen contiene el valor verdadero si y sólo si forman parte de un contorno o un valor falso en el caso contrario).

Este sub-proceso junto con la captura de imágenes desde el dispositivo son los que ocupan casi la totalidad del tiempo de procesamiento.

3.4 Analizador

Dada la imagen de entrada I preprocesada por el algoritmo de detección de contornos, este proceso tiene como fin decidir cuáles de éstos píxeles marcados como verdaderos son bordes activos y cuales son bordes pasivos. Para lograrlo se basa en el algoritmo, implementado en este trabajo, denominado *Wave Height Analyzer* (Analizador de Altura de Ola, ver Punto 4).

Una vez detectado los bordes activos, se realiza una conversión de unidades, pasando de

altura de superficie libre en píxeles a altura de superficie libre en metros, por medio de uno de los parámetros de configuración que establece la relación de equivalencia entre píxeles y metros.

A diferencia de las otras etapas, ésta tiene como salida un valor numérico (y no una imagen), que es la salida del proceso completo, esto es, la altura de la superficie libre detectada y convertida en metros.

3.5 Proceso completo

En una primera instancia, tenemos la imagen de entrada sin ningún tipo de transformación (obtenida directamente de la cámara de vídeo o un archivo), ésta es procesada por el Rectificador dando lugar a la segunda imagen, la cual solo contiene la zona seleccionada por el usuario. Ésta imagen resultante es la entrada del proceso denominado *Recortador*, el cual descarta fragmentos de la misma generando una imagen con menor cantidad de columnas. El siguiente proceso involucrado es el Detector de bordes, encargado de localizar los contornos en la imagen de entrada (por medio del algoritmo *Canny Edge Detector*) dando lugar a una nueva imagen en la que sólo se aprecian bordes. Puede existir gran cantidad de ruido (bordes falsos) por encima y por debajo de la superficie libre, lo que varía mucho dependiendo de varios factores como la luz utilizada, suciedad en las paredes del canal, parámetros utilizados en el algoritmo de *Canny Edge Detector* (umbrales de “sensibilidad” a la detección de contornos), entre otros. Por último se encuentra el Analizador, el cual tiene como finalidad detectar la altura de ola en la imagen de contornos procesada anteriormente. Este proceso tiene como salida un valor numérico perteneciente a la altura de ola detectada.

4 ALGORITMOS UTILIZADOS

4.1 Algoritmo Perspective transform

Algoritmo utilizado en la etapa de rectificación de la imagen (primera parte del proceso). Dada la imagen de entrada I_{mn} y ocho puntos de configuración (4 de origen y 4 de destino) se genera la matriz de transformación T mapeando cada píxel i,j de la imagen origen (I) en su correspondiente posición i',j' de la imagen destino I' como se muestra en la ecuación (1):

$$i' = \frac{m_{00}i + m_{01}j + m_{02}}{m_{20}i + m_{21}j + m_{22}} \quad (1)$$

$$j' = \frac{m_{10}i + m_{11}j + m_{12}}{m_{20}i + m_{21}j + m_{22}}$$

donde

$$T = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \quad (2)$$

con $m_{22} = 1$.

Los valores de los m_{ij} se obtienen resolviendo un sistema lineal de ecuaciones utilizando los 8 puntos (origen y destino) seleccionados por el usuario y el sistema (Perez Paladini, 2012).

4.2 Algoritmo Gaussian Blur

Algoritmo también conocido como *Gaussian smoothing* (Fisher et al., 2003), cuya finalidad es disminuir la amplitud de las variaciones de la imagen (ruidos), especialmente utilizado en la fase previa a los algoritmos de detección de contornos.

Este algoritmo se basa en la operación matemática de convolución utilizando un *kernel* con distribución gaussiana (Perez Paladini, 2012).

4.3 Canny Edge Detector

Desarrollado por Canny (1986), tiene como finalidad detectar los bordes en una imagen, filtrando todo aquello que no pertenezca a un contorno. Este algoritmo se puede dividir en 5 pasos: *Smoothing*, *Finding gradients*, *Non-maximum suppression*, *Double Thresholding*, *Edge tracking by hysteresis* (Canny, 1986; Perez Paladini, 2012).

4.4 Algoritmo Wave Height Analyzer

Ultimo algoritmo del proceso, tiene como finalidad determinar cuál de todos los contornos detectados en la imagen corresponde a la altura de ola que se está analizando. Definimos el funcionamiento como sigue:

Sea I_{mn} la imagen de entrada, donde $I_{mn} \ni a_{ij} = 1$ si y solo si a_{ij} (píxel) pertenece a un contorno detectado.

Sea h_0 el valor de la altura de ola inicial en píxeles, esto es, el número de fila en la imagen I_{mn} en la que se encuentra la altura inicial ($0 \leq h_0 \leq m$).

Sea V una constante numérica predefinida por el usuario representando la máxima variación de altura de ola aceptable en píxeles (cantidad máxima de píxeles aceptables que puede cambiar la ola de un paso al otro).

Se asume además que el movimiento de la ola se realiza a lo largo de las columnas de píxeles de las imágenes (i.e. podemos observar el movimiento de la ola en una columna k de píxeles a través de las diferentes filas de la imagen).

Teniendo en cuenta que como objetivo principal, este algoritmo debe obtener la altura de ola en un punto determinado de la imagen (en una columna determinada), se asume que este punto se encuentra sobre la columna $n/2$.

Podemos describir entonces el funcionamiento del algoritmo como:

1. Para cada columna $I_{*k}: 0 \leq k \leq n$, sea \bar{w}_k la ultima altura calculada en la k -ésima columna (donde \bar{w}_k es el índice de una fila), buscamos la nueva altura w_k de la siguiente forma: Partiendo de $a_{w_k k} \in I_{mn}$, buscamos el i tal que cumpla

$$a_{ik} = 1 \wedge |i - \bar{w}_k| \leq |j - \bar{w}_k| \quad \forall j = 0..m: a_{jk} = 1 \quad (3)$$

asignando entonces $w_k = i$. En caso de que tal i no exista asignamos $w_k = \bar{w}_k$. En palabras, esto no es mas que encontrar el píxel marcado como verdadero que se encuentre más cercano (en posición) a la ultima altura calculada (\bar{w}_k).

2. Sea $k = n/2$ la columna de mayor importancia (la que se está analizando), se calcula la varianza $v = |w_k - \bar{w}_k|$. Si $v \leq V$ se termina el algoritmo y se devuelve v , caso contrario se continua con el siguiente paso. En palabras, calculamos la distancia entre la altura anterior y la altura actual y verificamos si esta altura es menor que la constante V , de ser así se la devuelve, caso contrario se sigue con 3.

3. Sea $k = n/2$ se calculan las $v_i = |w_{k-i} - \bar{w}_{k-i}| + |w_{k+i} - \bar{w}_{k+i}|$, $i = 1.. \frac{n-1}{2}$ y se elije el i tal que $v_i = \min v_j$; $j = 1.. \frac{n-1}{2}$ y devolvemos h como altura. En caso de $h > V$ se devuelve

\bar{w}_k . En palabras, esto no es más que buscar las alturas correspondientes a los bordes del punto que se está analizando (que son las columnas de píxeles a los alrededores de la que se analiza) que presente la menor variación, si estas variaciones son mayores que V se devuelve la última altura detectada en la columna k .

Cada vez que finaliza el algoritmo se actualizan las últimas alturas encontradas para evitar acumulación de error. Las \bar{w}_k de cada una de las columnas son inicializadas por defecto con el valor H_0 (altura inicial de ola).

Puede parecer que el paso 3 agrega complejidad innecesaria, pero debido a que en ciertas situaciones (por problemas de luz o de entorno), la detección de contornos no se realiza correctamente en ciertos puntos (y por ende la altura de ola no es detectada), es necesario una forma alternativa de encontrar la altura de la superficie libre a los alrededores del punto que se está analizando, evitando de esta forma posibles mediciones erróneas como tomar un contorno marcado no perteneciente a la altura de ola y asumiendo que lo es. De esta forma se logra que el algoritmo sea más robusto y por ende un sistema menos propenso a fallas.

5 MEDICIONES Y ANÁLISIS DE LOS RESULTADOS

5.1 Ensayos propuestos

Los ensayos se realizaron intentando mantener constante los factores externos (como luz del entorno, intensidad, zona de captura) a excepción de la posición de la cámara en donde se trabajó con 2 ubicaciones distintas (**P1** y **P2**) con el fin de verificar cómo reaccionaba el sistema para cada una de éstas posiciones. Se realizaron inicialmente ensayos con oleajes monocromáticos o regulares. Sin embargo, en este trabajo se presentan solamente los resultados sobre ensayos de oleajes irregulares, debido a que aparecen más tipologías de errores en el proceso de detección.

P1 se caracterizaba por estar el lente a la altura de la superficie libre en reposo y a 0,5 metros del canal, mientras que **P2** se encontraba por debajo del seno de la ola más grande de la muestra y a una distancia de 0,9 m.

En todos los casos los datos fueron comparados con los capturados por sensores resistivos introducidos en el canal como se lo muestra de manera esquemática en la [Figura 3](#).

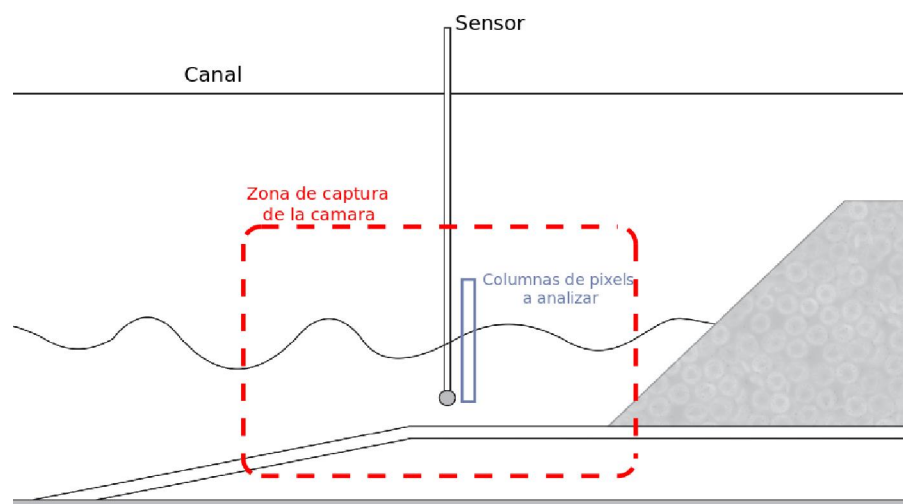


Figura 3: Esquema de la zona de captura (de la vídeo cámara) y analizada por el sistema implementado en el canal de laboratorio.

5.2 Datos del oleaje ensayado

En cada uno de los ensayos realizados el sistema implementado captura la altura de ola a lo largo del tiempo en un punto determinado del canal. Este conjunto de datos ha sido

comparado con los obtenidos por los sensores resistivos utilizados en el laboratorio con el fin de poder determinar la precisión del sistema implementado. En base a esto se han realizado dos tipos de comparaciones distintas: a) los datos “crudos” (altura en el tiempo) medidos por la cámara y por el sensor resistivo; b) análisis espectral del oleaje obtenido por el sistema y por el sensor y la comparación de los parámetros asociados al mismo (H_s - altura de ola significativa - y f_p - frecuencia de pico - a los oleajes irregulares). Se han usado espectros del tipo JONSWAP (*JOint North Sea WAve Project*).

5.3 Resultados

Durante el primer ensayo realizado se simuló un oleaje irregular con parámetros de altura de ola significativa (H_s) igual a 5 cm y frecuencia del pico (f_p) de 1 Hz. La duración del ensayo fue de 5 minutos y se ubicó la cámara en la posición **P1**. Los datos capturados de la altura de ola por ambos métodos (sistema implementado y sensor resistivo) se muestran en la Figura 4.

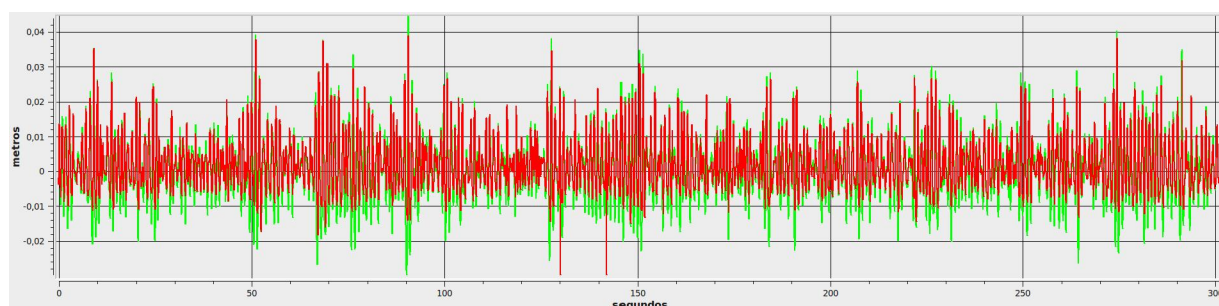


Figura 4: Datos obtenidos por ambos métodos (sensor resistivo en verde y sistema implementado en rojo) durante 5 minutos (cámara en la posición **P1**).

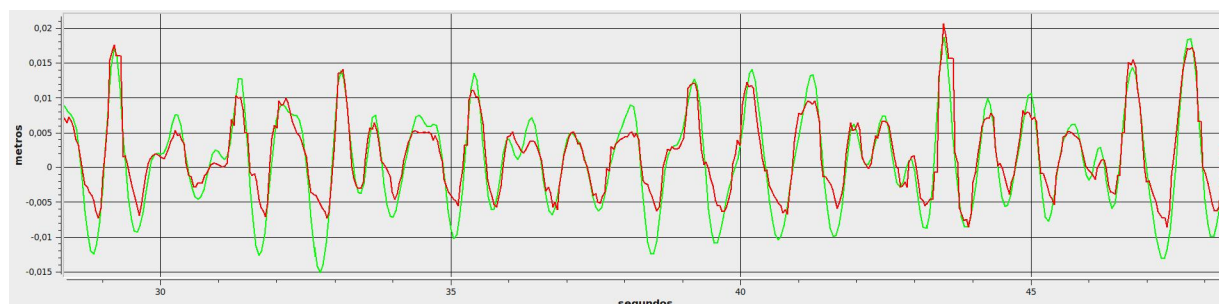


Figura 5: Detalle de los datos obtenidos por ambos métodos (sensor resistivo en verde y sistema implementado en rojo) durante 20 segundos (segundo 28 al 48 aproximadamente).

Como se puede observar (Figura 5), en la gran mayoría de los valles (parte inferior) de las olas, el sistema implementado difiere hasta 0,0153 metros en las diferencias de mediciones extremas con respecto al obtenido por el sensor resistivo. Por otro lado se puede ver la presencia de pequeños saltos a lo largo de la línea roja del gráfico, esto es otro problema relacionado con la velocidad de captura de la cámara.

Por otro lado (Figura 6) se comparan los espectros obtenidos por el sensor resistivo y por el sistema. Claramente se puede observar (a través del espectro de JONSWAP) que por un lado la simetría de ambos espectros se asemeja, pero que el asociado a los datos obtenidos por el sistema implementado es inferior al del obtenido por los del sensor resistivo (como se lo percibe en la Figura 5 en donde las ondas capturadas por el sensor resistivo poseen una mayor amplitud que las capturadas por el sistema).

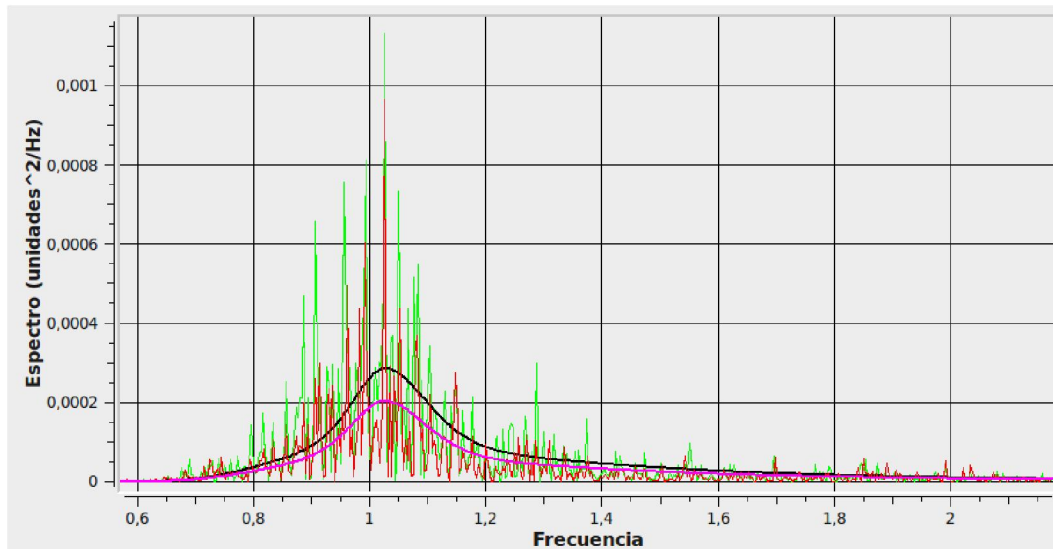


Figura 6: Espectros calculados para los datos obtenidos por ambos métodos (sistema implementado y sensor resistivo) para un oleaje irregular con parámetros $H_s = 5$ cm y $f_p = 1$ Hz durante 5 minutos (cámara en la posición **P1**). En verde (y negro) se observa el espectro resultado (y espectro de JONSWAP respectivamente) de los datos capturados por el sensor resistivo mientras que en rojo (y magenta) el espectro calculado (y espectro de JONSWAP) por el sistema implementado.

Se calcularon además los parámetros de interés (H_s , T_p) asociados al oleaje capturado por ambos métodos, como se lo muestra en la [Tabla 1](#), donde claramente se observa una gran diferencia entre las alturas de olas significante (H_s), mientras que el periodo de pico (T_p) se asemeja bastante (Tabla 1).

| Método | H_s (m) | T_p (s) |
|----------------------|-----------|-----------|
| Sensor resistivo | 0,040 | 0,974 |
| Sistema implementado | 0,034 | 0,975 |

Tabla 1: Comparación de los parámetros H_s y T_p asociados a los datos capturados por el sistema implementado y el sensor resistivo (cámara en la posición P1).

En el segundo ensayo realizado, donde se trabajó con el mismo oleaje irregular antes presentado pero con la cámara en una posición distinta (**P2**), se lograron mejores resultados debido a que al estar la cámara más alejada la interferencia detectada (junto con la velocidad de captura de la cámara) es menor. La velocidad de captura influye mucho a la hora de tomar las imágenes (en donde se genera ese efecto de *blur*): si la velocidad de captura es lenta, las imágenes se ven difusas y los contornos no se distinguen, lo cual introduce un error; al alejar la cámara el movimiento de la ola afecta menos cantidad de píxeles y se pueden detectar mejor los contornos. En la [Figura 7](#) y [Figura 8](#) se pueden observar los resultados obtenidos por ambos métodos.

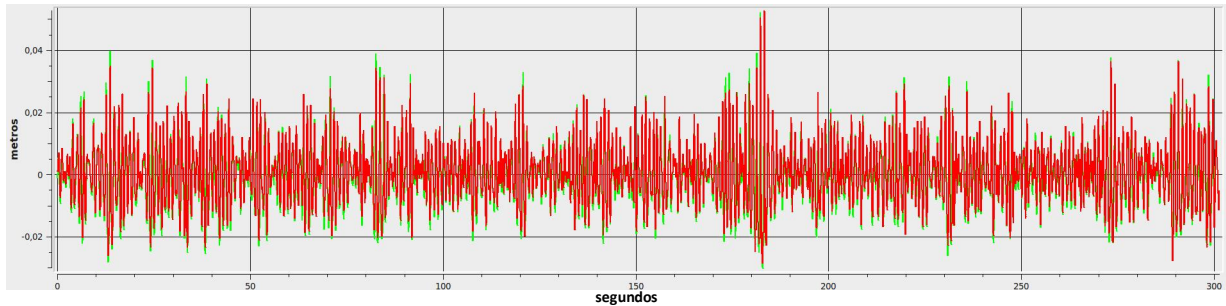


Figura 7: Datos obtenidos por ambos métodos (sensor resistivo en verde y sistema implementado en rojo) durante 5 minutos (cámara en la posición **P2**).

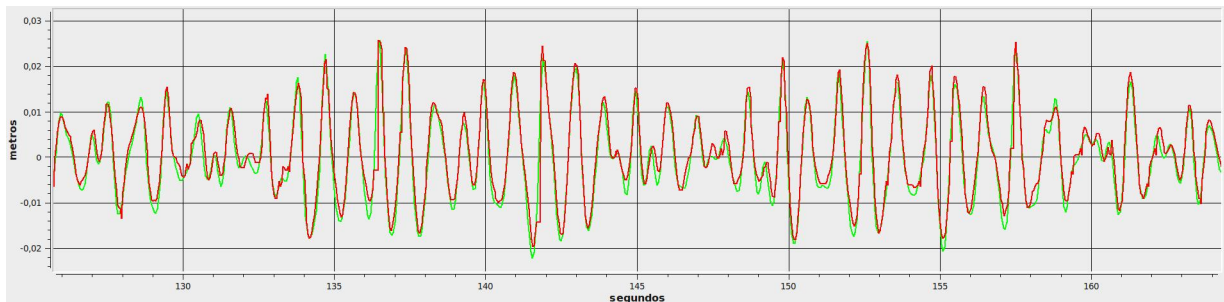


Figura 8: Detalle de los datos obtenidos por ambos métodos (sensor resistivo en verde y sistema implementado en rojo) durante 38 segundos (segundo 126 al 164 aproximadamente).

Se puede ver que los datos coinciden en su gran mayoría (la máxima diferencia encontrada es de 0,0055 metros), además de que no se perciben a grandes rasgos cambios abruptos como en el caso anterior. Observando, ahora, los espectros asociados a los datos (Figura 9) es posible notar semejanzas tanto en simetría como amplitud, pero con un pequeño desfase sobre el eje x (en el espectro de JONSWAP asociado a los datos capturados por el sistema en base al generado por los datos del sensor resistivo) y de menor amplitud (causado por la diferencia en los valles detectados de las olas).

En la Tabla 2 podemos observar nuevamente la buena correspondencia entre los 3 valores de interés.

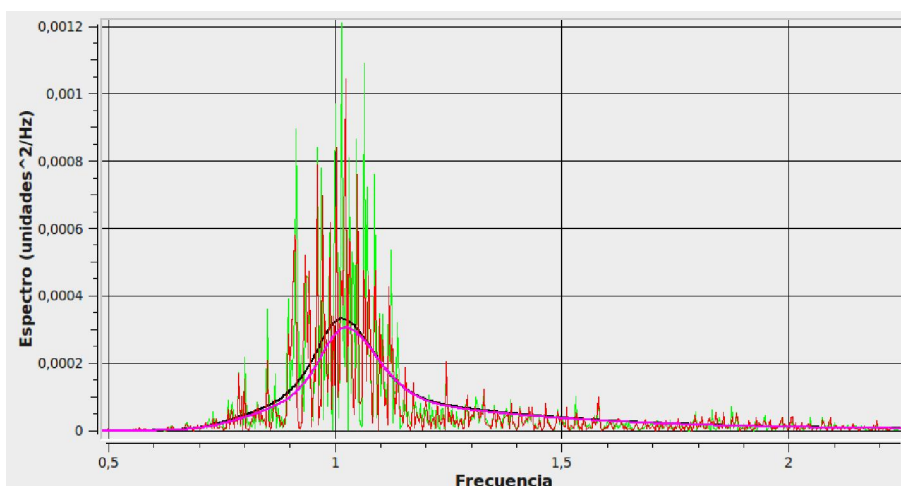


Figura 9: Espectros calculados para los datos obtenidos por ambos métodos (sistema implementado y sensor resistivo) para un oleaje irregular con parámetros $H_s = 5$ cm y $f_p = 1$ Hz durante 5 minutos (cámara en la posición **P2**). En verde (y negro) se observa el espectro resultado (y espectro de JONSWAP respectivamente) de los datos capturados por el sensor resistivo mientras que en rojo (y magenta) el espectro calculado (y espectro de JONSWAP) por el sistema implementado.

| Método | H_s (m) | T_p (s) |
|----------------------|-----------|-----------|
| Sensor resistivo | 0,042 | 0,987 |
| Sistema implementado | 0,041 | 0,979 |

Tabla 2: Comparación de los parámetros H_s y T_p asociados a los datos capturados por el sistema implementado y el sensor resistivo (cámara en la posición **P2**).

6 CONSIDERACIONES GENERALES

6.1 Ángulo y posición de captura

Uno de los posibles errores a la hora de detectar la altura de la superficie libre se debe a la posición y ángulo de la cámara con respecto al nivel en reposo del fluido en el canal. En el caso del primer ensayo, la posición de la lente de la cámara se encuentra casi a la altura de la superficie libre (en reposo). Cuando la pala mecánica comienza a producir el oleaje a medir, la superficie libre se desplaza por encima y por debajo del punto de reposo. En el momento en el que la superficie libre se encuentra por debajo de la altura del lente de la cámara se produce un efecto no deseado que es la captura de la superficie del fluido a lo largo del ancho del canal, la cual genera reflejos e introduce ruido a la hora de detectar los contornos de la ola.

En la [Figura 10](#) se puede observar éste caso, en donde la superficie del fluido está por debajo del lente de la cámara y observándose el ruido introducido en la detección de contornos por medio de una doble línea con trazos de bordes entre las mismas provenientes del reflejo mencionado anteriormente.

Esta situación, en donde se detecta una “doble superficie” puede generar una falsa detección de la altura de ola (más específicamente el seno, valle o parte inferior de la ola) dando como resultado una altura menor a la real como se ve en la [Figura 11](#). Se puede observar que las crestas de las olas son detectadas correctamente mientras que los senos no.

Claramente esto fue lo que sucedió en todos los ensayos realizados con la cámara ubicada en la posición **P1**.

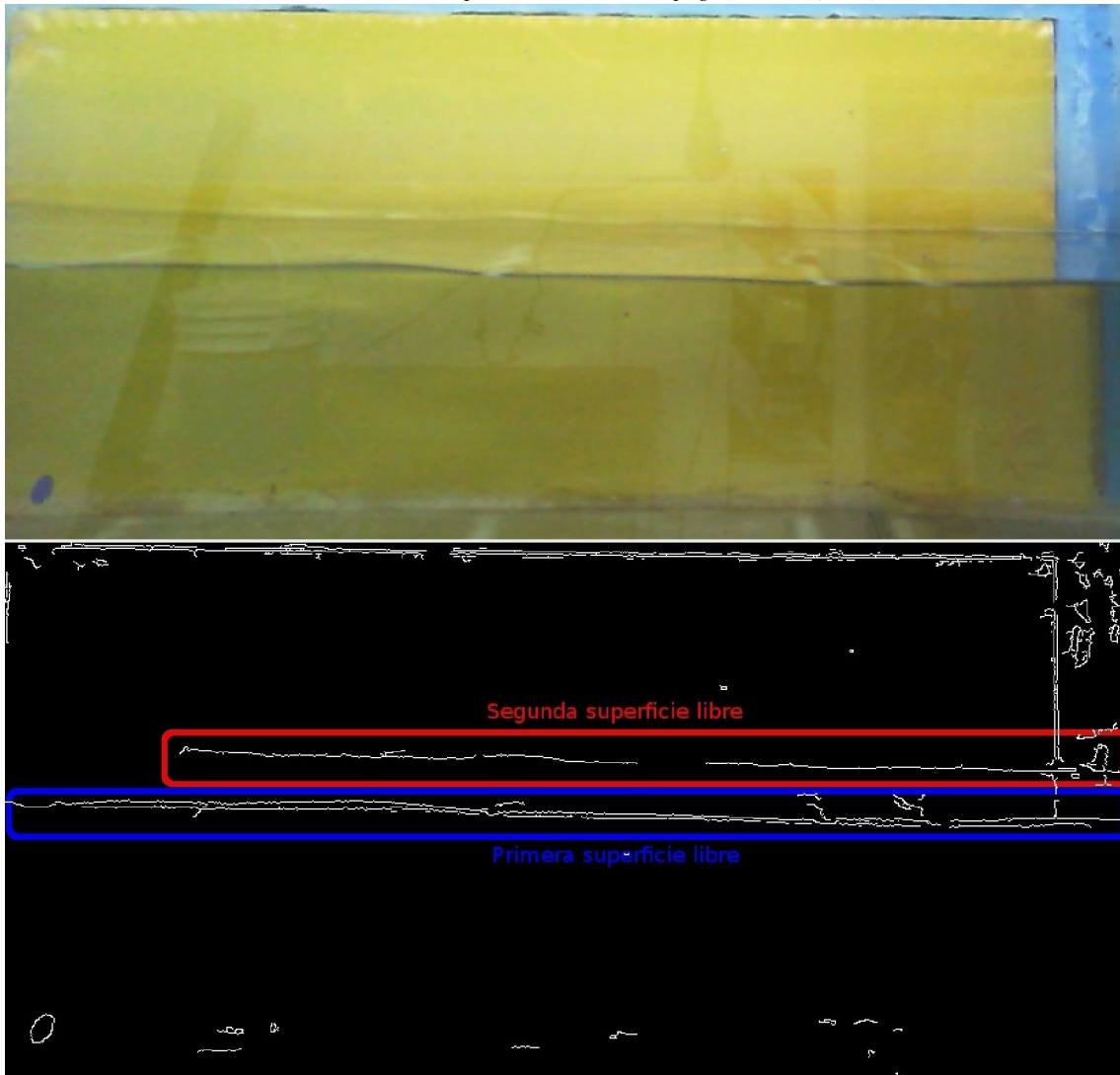


Figura 10: Errores introducidos por la posición y ángulo de la cámara a la hora de detectar los contornos de la superficie libre. Altura de la superficie libre por debajo del lente de la cámara.

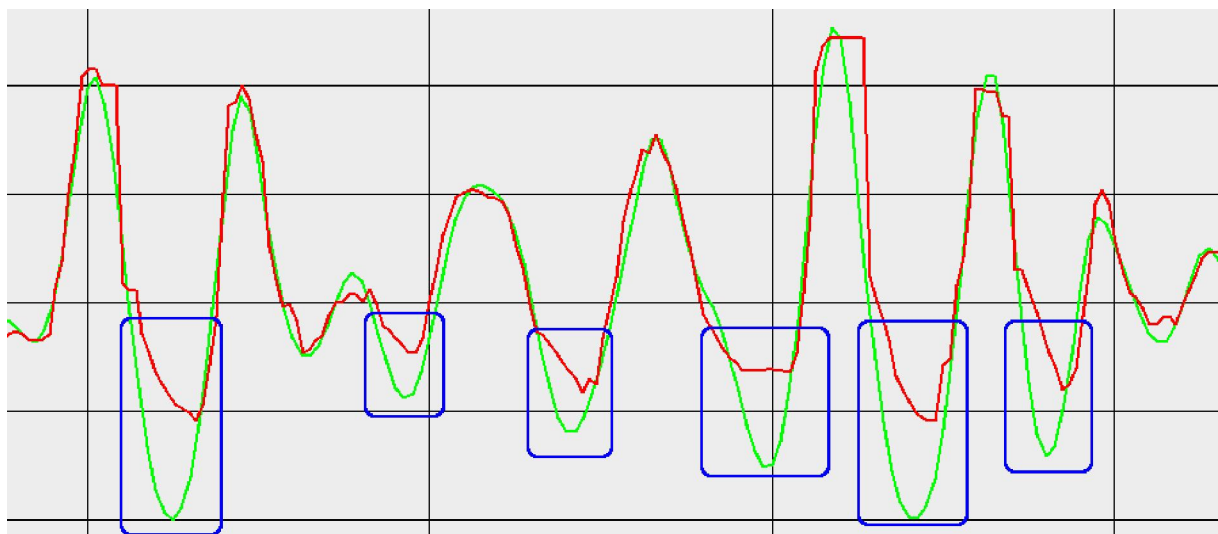


Figura 11: Errores (remarcados en azul) introducidos por la posición y ángulo de la cámara a la hora de detectar los contornos de la superficie libre cuando la altura de la superficie libre se encuentra por debajo del lente de la cámara. En rojo se muestra los datos capturados por el sistema implementado mientras que en verde los del sensor resistivo.

6.2 Velocidad de captura de la cámara

La velocidad de captura de la cámara juega un importante papel a la hora de poder detectar la superficie libre. El problema surge cuando la velocidad de movimiento de la superficie libre es alta, de manera tal que la cámara no logra capturar el contorno de la misma claramente, si no en cambio la toma como un borde difuso. En tal situación el algoritmo de *Canny* es incapaz de detectar la superficie libre (contorno) en distintas partes de la ola.

Como consecuencia de la implementación del algoritmo *Wave Height Analyzer* pueden surgir 2 problemas: 1) La altura puede permanecer constante (Figura 12); 2) La altura puede saltar abruptamente (Figura 13).

Claramente ambos casos (1 y 2) generan grandes diferencias en los valores capturados por el sistema implementado en comparación con los obtenidos por el sensor resistivo. Una posible situación que solventa estos errores es la disminución de los valores umbrales utilizados en el algoritmo de *Canny*. De esta manera el algoritmo es más permisivo y detecta una mayor cantidad de contornos (y por ende mas interferencia o los que denominamos bordes pasivos). Esta interferencia varía mucho a medida que la ola se traslada generando un cambio brusco en los valores de la altura de la superficie libre detectada por el sistema.

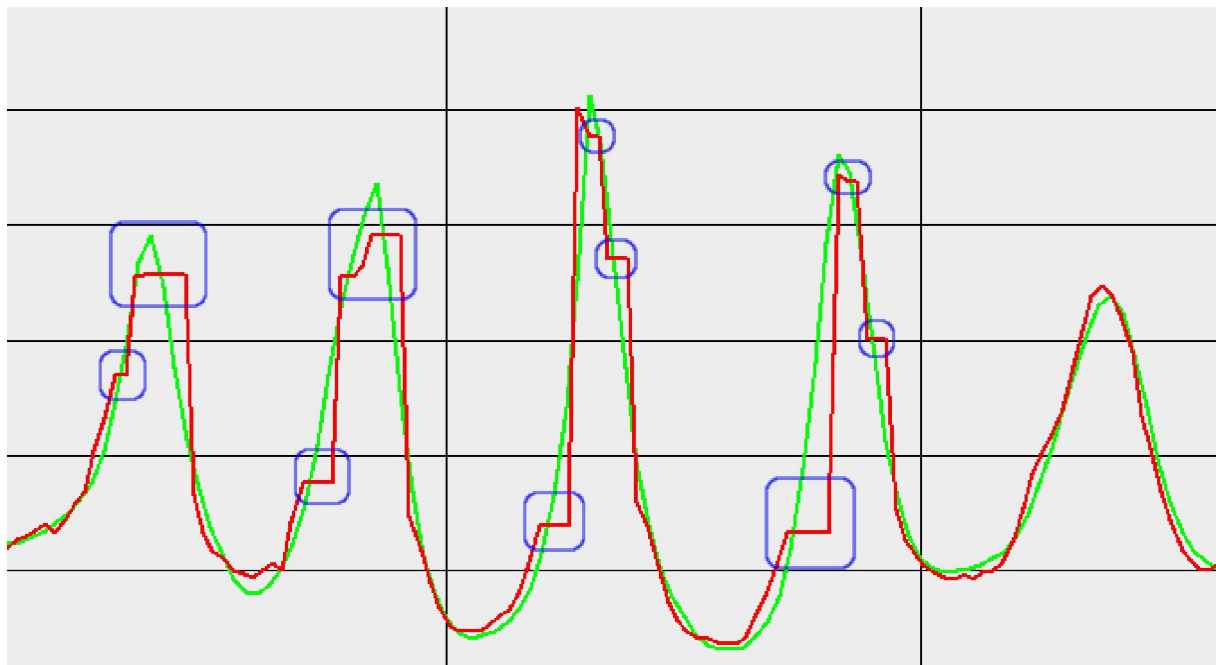


Figura 12: Errores (remarcados en azul) causados por la imposibilidad de detectar la superficie libre, en donde el sistema devuelve durante cierto tiempo la altura de ola detectada en la captura anterior. En rojo se observa los datos tomados por el sistema implementado y en verde se muestra los datos obtenidos por el sensor resistivo.

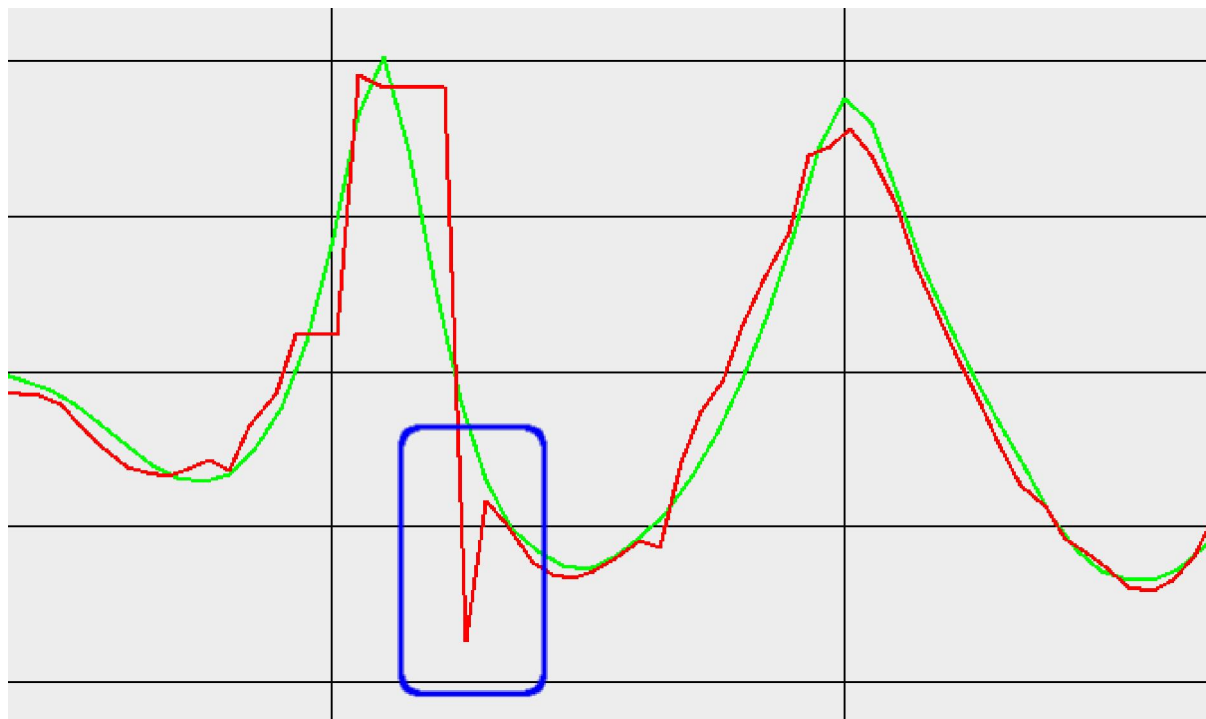


Figura 13: Error (remarcado en azul) causado por la imposibilidad de detectar la superficie libre, en donde el sistema devuelve la altura más cercana al último punto detectado. En rojo se observa los datos tomados por el sistema implementado y en verde los obtenidos por el sensor resistivo.

6.3 Superficie libre

Otro inconveniente que se pudo observar es el hecho de la doble línea (doble contorno) que se detecta al capturar la altura de la superficie libre como lo indica la [Figura 14](#). Este tipo de situación genera en algunos casos (cuando el movimiento de la ola es muy rápido/brusco) un error similar al planteado anteriormente, en donde los 2 bordes pueden ser no detectados de forma intercalada (primero se detecta uno y luego el otro) generando pequeños saltos (que en este caso son de aproximadamente 4 mm) en los datos capturados y por ende en la altura de ola resultado. Probablemente esto pueda ser resuelto utilizando distintos tipos de luces que eviten resaltar el borde de la superficie de esa forma. Otra posible solución (a investigar) podría ser colorear todo el líquido del canal con un tono opaco de forma tal que no se alcance apreciar la línea divisoria entre el agua y el aire.

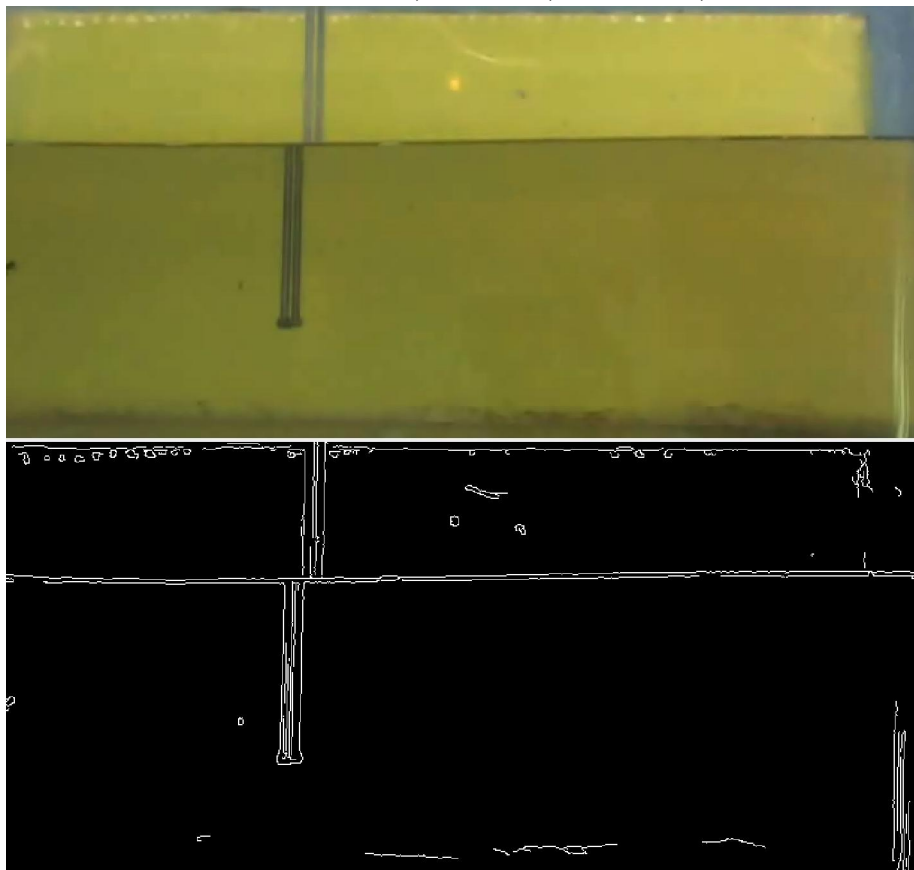


Figura 14: Doble contorno detectado de la superficie libre (separadas aproximadamente por unos 4 mm).

7 CONCLUSIONES

En este trabajo se propuso desarrollar un sistema capaz de capturar la altura de la ola en tiempo real en un canal 2D de laboratorio por medio de una cámara web. Como se pudo observar en los análisis de los resultados, la precisión del sistema está ligada y fuertemente afectada por factores externos como la luz utilizada (intentando buscar siempre aquella que genere mayor contraste entre el fluido y la superficie y al mismo tiempo no genere reflejos sobre el vidrio del canal), la cantidad de partículas tanto en el fluido como en las paredes del canal, la posición y ángulo de la cámara la cual cumple un importante papel, y además los valores de configuración para el algoritmo de detección de contornos, el cual varía dependiendo de los factores recientemente mencionados.

Se puede decir que el sistema logra capturar y analizar la altura de ola en tiempo real y a una velocidad superior a la mínima esperada (alcanzando 25 Hz) mostrando en pantalla al mismo tiempo el proceso del ensayo realizado y los datos obtenidos (lo que quiere decir que puede correrse la interfaz gráfica en simultáneo con el proceso de captura). Aún así no se logró alcanzar una precisión milimétrica para poder realizar mediciones que requieran un error menor a 0,55 cm, donde las causas se debieron tanto por motivos de hardware (velocidad de captura), software (post-procesamiento de datos, utilización de otros algoritmos) como de factores externos (luminosidad, partículas en el fluido y en las paredes del canal, etc). Por otro lado se pudieron realizar mediciones en donde los errores de los valores de importancia como H_s y T_p se encuentran aproximadamente en el orden de los 0,1 cm y 0,009 segundos respectivamente.

Por otro lado se considera altamente viable la posibilidad de aumentar la precisión del sistema por medio de distintos algoritmos en la fase final que complementen los posibles errores introducidos por la velocidad de captura de la cámara, como así también en una etapa posterior a la detección de los datos (post-procesamiento) en la que se filtren y corrijan

aquellos valores que se consideren erróneos.

Fuera del sistema implementado para detectar la altura de la superficie libre, se logró desarrollar además un *framework* (entorno) el cual permitirá crear futuras aplicaciones (que brinden la oportunidad de medir otras variables hidrodinámicas) de una forma más fácil y rápida, dando también la opción de utilizar la GPU como unidad de procesamiento.

REFERENCIAS

- Brady, P.D.M., Boutounet, M. and S. Beecham. *Free Surface Monitoring Using Image Processing*. 15th Australasian Fluid Mechanics Conference. The University of Sydney, Sydney, Australia, 2004.
- Bradski, G., Kaehler A.. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Ed., 2008.
- Canny, J.F.. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-698, 1986.
- Fisher, P., Walker & Wolfart. "Spatial Filters - Laplacian of Gaussian". Retrieved 2010-09-13, 2003.
- Nadernejad E., Sharifzadeh S. and Hassanpour H.. Edge Detection Techniques: Evaluations and Comparisons. *Applied Mathematical Sciences*, Vol. 2, 2008.
- Pérez, J.C., Ortiz, C.A., Osorio, A.F., Mejía, C.E. y R. Medina (2012) Environmental Applications of Camera Images Calibrated by Means of the Levenberg-Marquardt Method, *Computers & Geosciences*, por aparecer.
- Perez Paladini, A., *Software para la detección de variables hidrodinámicas en laboratorio mediante técnicas de vídeo de alta definición*. Tesis de fin de carrera, FAMAF (Facultad de Matemática Astronomía y Física, Universidad Nacional de Córdoba), 2012.
- Sánchez, F.J.. *Medición y Análisis de las Variaciones en el Nivel de un Modelo Físico Empleando Imágenes*. Universidad Autónoma Metropolitana Unidad Azcapotzalco, México, 2010.