



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

TELECOM ESCUELA
TÉCNICA **VLC** SUPERIOR
DE **UPV** INGENIEROS DE
TELECOMUNICACIÓN

MONITORIZACIÓN DE FLOTANTES EN RÍOS USANDO VISIÓN ARTIFICIAL

David Llácer Giner

Tutor: Antonio José Albiol Colomer

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2017-18

Valencia, 1 de Diciembre de 2017

Resumen

En los últimos años la concienciación con el cuidado del medio ambiente ha incrementado de manera exponencial llegando a convertirse en una de las prioridades en investigación, innovación y desarrollo.

En este proyecto se presenta un nuevo algoritmo de detección de basuras flotantes en ríos con el cual poder controlar, y en consecuencia reducir, la cantidad de desechos que pasan arrastrados por la corriente. El tratamiento de imagen ha consistido en escoger una línea del vídeo la cual se va a capturar de cada fotograma y se va a apilar formando una imagen en la que, si algún objeto ha atravesado dicha línea a causa de la corriente, aparecerá dicho objeto. A continuación, se analizarán estas pilas de líneas realizando un promediado vertical con el cuál se modelará estadísticamente el aspecto de los píxeles en cada posición. Una vez obtenido dicho modelado, se realizará un análisis estadístico del tipo de objetos que son arrastrados por la corriente habitualmente (formaciones de espuma generalmente), basándonos en dos de sus características básicas: posición y tamaño, mediante el cálculo de momentos. Con esto obtendremos una matriz de probabilidad de aparición de dicha espuma. A partir de este momento, cada vez que se inicie el proceso de detección acudiremos a la matriz de probabilidades para comprobar si los objetos que se han detectado son realmente objetos o son simplemente irregularidades en la superficie del agua. Los resultados de los experimentos realizados demuestran el funcionamiento del algoritmo realizado, incluso en ríos en los que se ve el fondo del mismo siendo un entorno muy irregular. Además, se comentan las limitaciones de la detección y las propuestas de mejora del algoritmo.

Resum

En els últms anys la concienciació amb l'atenció del medi ambient ha incrementat de manera exponencial, arribant a convertir-se en una de les prioritats tant en investigació, innovació i desenvolupament.

En aquest projecte es presenta un nou algoritme de detecció de residus flotants en rius amb el qual poder controlar, y en conseqüència reduir, la quantitat de desfets que passen arrossegats per la corrent. El tractament d'imatge ha consistit en escollir una línia del video la qual es capturarà de cada fotograma del video i s'apilarà formant una imatge en la qual, si algun objecte ha travessat la línia a causa de la corrent, apareixerà l'objecte. A continuació, s'analitzaran aquestes piles de línies realitzant un amitjanat vertical amb el qual es modelarà estadísticament l'aspecte dels píxels en cada posició. Una vegada obtingut aquest modelat, es realitzarà una anàlisi estadística del tipus d'objectes que són arrossegats per la corrent habitualment (formacions de bromera generalment), basant-nos en dos de les seues característiques bàsiques: posició i grandària, mitjançant el càlcul de moments. Amb aquesta informació obtindrem una matriu de de probabilitat d'aparició de la bromera. A partir d'aquest moment, cada vegada que s'inicie el procés de detecció acudirem a la matriu de probabilitats per comprovar si els objectes que s'han detectat són realment objectes o són simplement irregularitats en la superfície de l'aigua. Els resultats dels experiments realitzats demostren el funcionament de l'algoritme realitzat, inclòs en rius en els que el nivell de l'aigua es baix i es veu el fons fent que siga un entorn molt irregular. A més, es comenten les limitacions de la detecció i les propostes de millora de l'algoritme.

Abstract

During the last years, environment-friendly awareness has increased in a substantial way, becoming one of the main priorities in research and development.

A floating waste detection algorithm focused on rivers is introduced in this project to control, thus, reducing, the amount of waste that is dragged down by the river tides. Image processing has been approached by choosing a video line which is to be captured frame by frame and stacked, resulting in an image in which, if any object crosses such aforementioned line because of the water streams, such item will appear. Moreover, this stacked-in-piles lines will be analysed through a vertical mean average we're going to use with to statistically model the pixel's looking in each position. Once modelling is obtained, a statistical analysis will be undertaken on the kind of objects that are being usually pushed by the water stream (i.e., foam clusters, generally), basing them on two of their main features: position and size, by momentum calculus. With such procedure, we obtain the foam surging probability matrix. From now on, every time the detection process is started, we'll resort to such matrix to test if the detected objects are really objects or simply inconsistencies of the water surface. The ending results of the experiments carried out show the functionality of the algorithm used, even in rivers where the deep bottom can be easily seen, being the latter fairly irregular as well. In addition, some limitations on detection are fully fleshed-out and, also, improvement suggestions about the algorithm's performance are commented.

Índice

Capítulo 1.	Introducción	3
Capítulo 2.	Objetivos	4
Capítulo 3.	Recursos y Software empleado	5
Capítulo 4.	Visión Global	6
4.1.	Formación de la pila	6
4.2.	Enventanado de la pila	7
4.3.	Tipos de ventana	8
4.3.1.	Solución a los objetos incompletos	9
4.4.	Ventajas de procesar una sola línea de cada fotograma	10
Capítulo 5.	Desarrollo	11
5.1.	Aprendizaje. Modelado estadístico de los píxeles	11
5.1.1.	Obtención de la Media y la Desviación Típica	12
5.1.2.	Obtención de la matriz de probabilidades	13
5.2.	Proceso de detección	19
5.3.	Tipos de algoritmo	21
5.3.1.	Algoritmo offline	21
5.3.2.	Algoritmo online	21
Capítulo 6.	Resultados	22
6.1.	Grabación de vídeo	22
6.2.	Colocación de la cámara	24
6.3.	Detección de objetos	26
6.4.	Fiabilidad del algoritmo	28
6.5.	Situaciones a tener en cuenta	29
6.5.1.	Problemática de los objetos transparentes	29
6.5.2.	Profundidad del río	30
6.6.	Forma de aumentar la robustez del algoritmo	31
Capítulo 7.	Conclusiones y propuestas de trabajo futuro	33
7.1.	Conclusiones	33

7.2. Propuestas de trabajo futuro	34
Capítulo 8. Bibliografía	35

Capítulo 1. Introducción

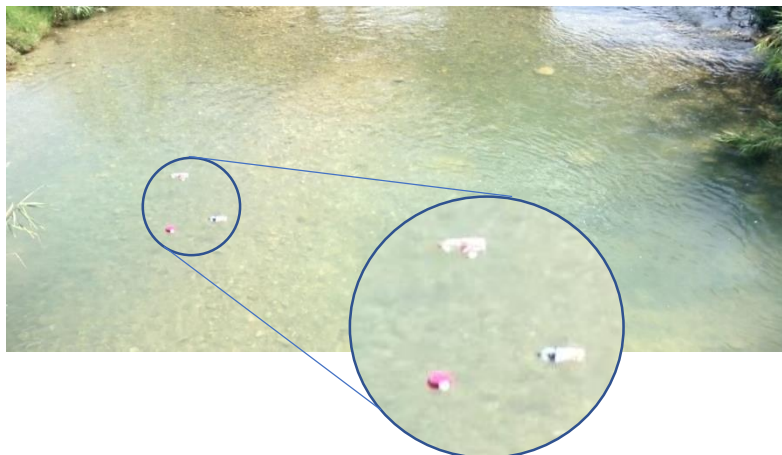
El agua es posiblemente el bien máspreciado que tenemos actualmente en el planeta. Se trata de un bien de primera necesidad e imprescindible para la vida. Además de esto, es un recurso energético natural no renovable, con el que se abastece gran parte de la industria mundial, desde la industria pesada hasta la industria textil, pasando por la metalúrgica, la siderúrgica, la petroquímica y un sinnúmero de empresas, que la utilizan para procesos de calefacción, enfriamiento y limpieza entre otros. Es por esto que el agua es un bien tan necesario y que debemos cuidar con gran prioridad para que el modelo de sociedad industrializada en el que vivimos pueda mantenerse.

La forma de obtención del agua depende sobre todo de cuál va a ser el uso que se le va a dar. Siendo esto así, las principales maneras de obtención de agua son: agua subterránea, agua procedente de la superficie (ríos y lagos), agua reciclada y agua desalinizada.

En este trabajo nos centraremos en el agua que obtenemos de los ríos. Precisamente, al ser una de las principales fuentes de agua que existen, es de crucial importancia que se mantengan en un estado de limpieza muy alto, no solo para las necesidades humanas, sino también para preservar todo el entorno natural que los rodea, es decir, la fauna y la flora que siempre está presente en cualquier cuenca hidrográfica.

El hecho de que el agua sea tan importante para la industria hace que los ríos estén estrechamente ligados a las grandes industrias y no siempre para bien, ya que muchas de ellas los utilizan como vertederos de los residuos que producen. Existen varios tipos de contaminación tales como residuos químicos, residuos metálicos y elementos derivados del petróleo como plásticos y sucedáneos, todos igual de preocupantes ya que no solo afectan a los ríos, si no que acaban desembocando y afectando a los mares, y cada uno de ellos se tratan y se gestionan de forma distinta.

En el siguiente trabajo se propone la monitorización de un río mediante visión artificial, y por consiguiente un algoritmo de detección utilizando tratamiento digital de imagen, de forma que podamos conocer la cantidad de residuos flotantes que pasan por el mismo con la finalidad de poner una solución, y conseguir que esta cantidad disminuya para poder mantener la salud de nuestros ríos en buen estado.



Capítulo 2. Objetivos

El objetivo principal de este trabajo es la realización de un algoritmo capaz de detectar el paso de objetos que flotan por la superficie de un río arrastrados por la corriente del mismo. Para ello el trabajo se divide en:

- Captación de imágenes y simulación del paso de objetos flotantes
- Diseño de un algoritmo para su detección.

El algoritmo de procesamiento deberá ser capaz de detectar los elementos ajenos al río que pasen flotando por el mismo y diferenciarlos del resto de elementos de la escena como puede ser la superficie del río. Además, deberá ser capaz de calcular datos objetivos tales como la velocidad de la corriente del río y el tamaño de los objetos detectados, así como la cantidad de objetos localizados.

Por tanto, de forma secuencial, los objetivos principales son:

- Búsqueda de localizaciones en las que grabar y realizar los ensayos.
- Grabación de los ensayos realizados.
- Realización de un algoritmo de tratamiento de imagen para la detección de objetos flotantes. Se debe tener en cuenta que el programa no sabrá de forma anticipada lo que está buscando, ya que no sabemos lo que puede pasar flotando por la superficie, de modo que deberá ser capaz de diferenciar un objeto partiendo de la base de que éste va a ser distinto de la superficie del río.

Primero, se deberá tener en cuenta que los ensayos se han realizado en distintas localizaciones, cuya característica más determinante a la hora de realizarlos es la profundidad del río, lo que hará que se aprecie o no el fondo en la grabación. Obviamente, el peor caso es en el que se ve el fondo, ya que hará que la superficie por la que pasa el objeto sea muy irregular.

Capítulo 3. Recursos y software empleado

Todo el trabajo se ha desarrollado en el sistema operativo Linux, que cuenta con gran cantidad de software de código abierto lo que facilita mucho la tarea del desarrollador. Además, ofrece la posibilidad de configurarlo y personalizarlo a gusto del usuario [1].

El lenguaje de programación empleado ha sido ‘C++’ y como entorno de desarrollo se ha elegido ‘Eclipse’, entorno de código abierto y multiplataforma [2].



Figura 1. Logos de izquierda a derecha: Linux – Eclipse – C++

Para el tratamiento de imagen se han empleado dos librerías de código abierto: *Clmg* [3] y *OpenCV* [4]. La primera de ellas se ha utilizado en el grueso del tratamiento de imágenes mientras que la segunda se ha utilizado mayormente para la adquisición de imágenes a partir de vídeos. En adición, se ha hecho uso de librerías proporcionadas y utilizadas en la asignatura Tratamiento Digital de Imagen del Grado de Tecnologías y Servicios de Telecomunicación de la Universidad Politécnica de Valencia. También se ha utilizado *Matlab* para graficar datos y visualizarlos [5].



Figura 2. Librerías utilizadas: Clmg y OpenCV.

Para la grabación de vídeo se ha utilizado un iPod 5G de la marca Apple con cámara iSight de 5 Mpx y grabación de vídeo HD1080 y 30 fps. Para poder colocarla de forma que captara los ríos se ha utilizado un trípode Gorilla Pod con las patas modificables para adaptarse a cualquier superficie, y un trípode rígido mediano para superficies planas.

La edición de los vídeos de los ensayos se ha realizado con la herramienta *Vegas Pro 12* de Sony, software profesional de edición de vídeo y audio.



Figura 3. iPod 5G – Trípode Gorilla Pod – Logo de Sony Vegas Pro 12

Capítulo 4. Visión global.

4.1. Formación de la pila.

El programa lee cada fotograma del vídeo, del cual coge una única línea (siempre la misma) elegida por el usuario.

Estas líneas se van apilando de forma ordenada (orden temporal) en lo que llamaremos “pila”, que será una imagen cuyo eje Y (vertical) será el eje del Tiempo y el eje X (horizontal) será el ancho del fotograma. De esta forma, se irá creando una imagen en la que, si algo se ha movido en dirección perpendicular a la línea elegida en el vídeo y ha atravesado la misma, entonces aparecerá dicho objeto en la pila.

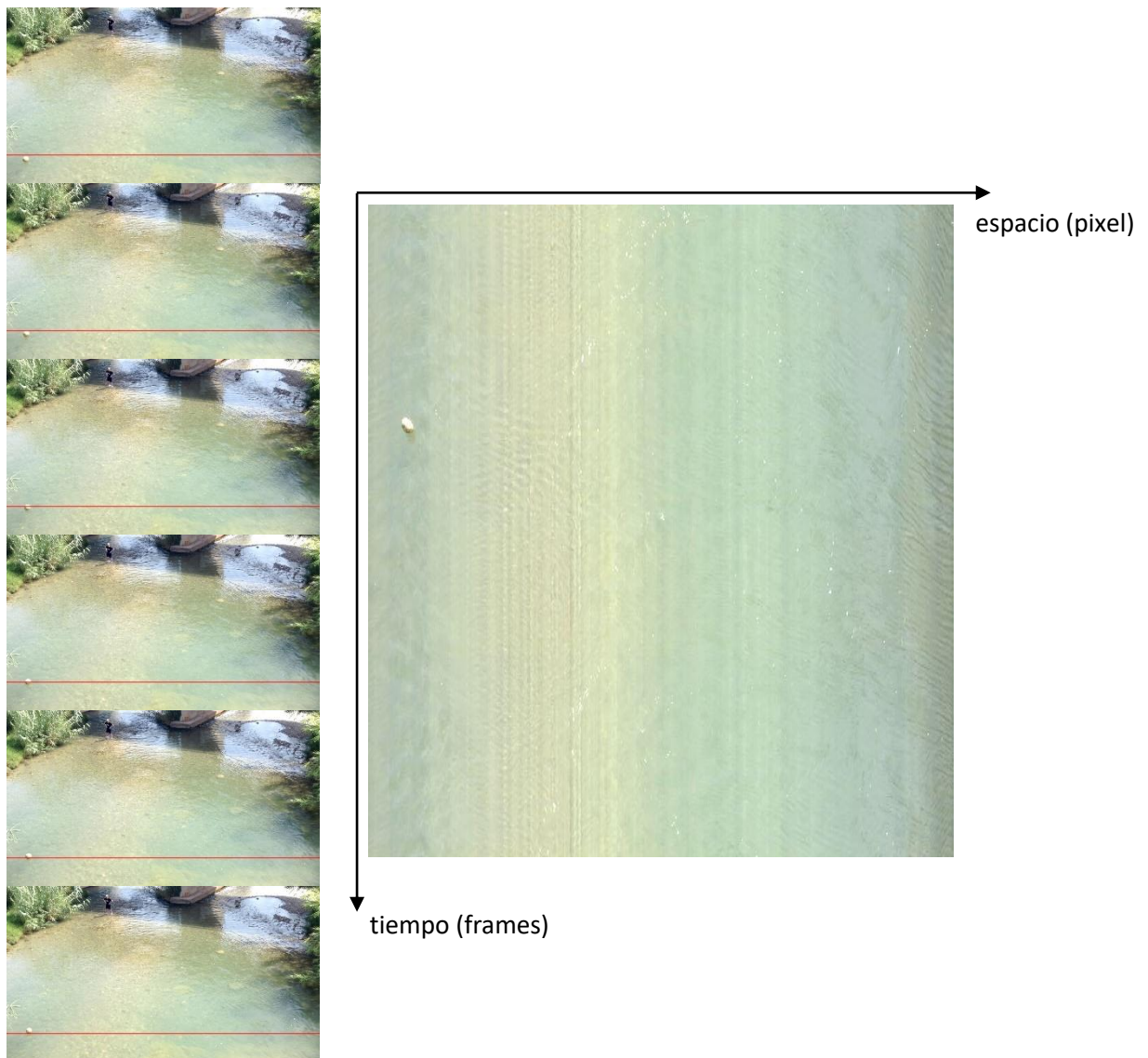


Figura 4. Formación de la pila.

La Pila tiene un tamaño limitado, que viene definido por la velocidad de fotogramas del vídeo (fps) y el tiempo (t) en segundos que el usuario quiera que dure, por ejemplo 60 segundos.

$$\text{Altura_Pila} = \text{fps} * t$$

4.2. Enventanado de la Pila.

La forma de analizar la Pila será la siguiente. Cada t fotogramas se cogerán un número t de líneas de la pila que formarán una imagen, de altura t y anchura el ancho del fotograma, a la que llamaremos ventana. Estas ventanas son las que se le pasarán tanto al estimador de estadísticas como al algoritmo de detección de detección.

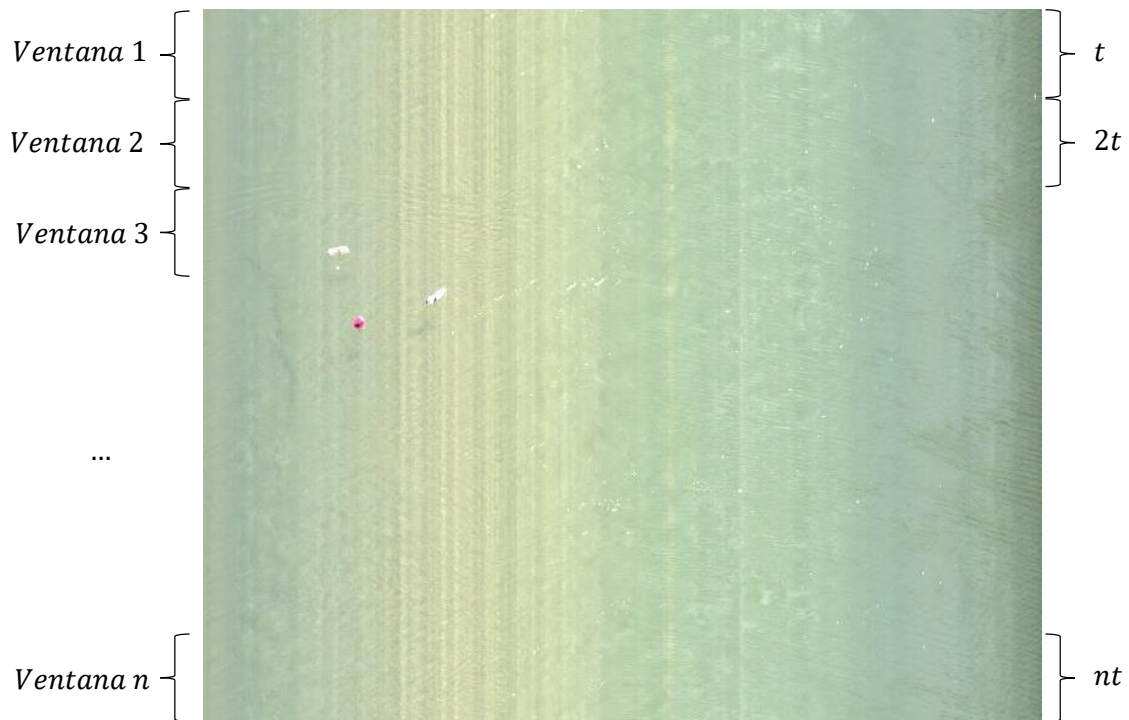


Figura 5. Enventanado de la Pila.

La Pila se sobrescribe a si misma cada vez que llega a su altura máxima. Del mismo modo, cuando cojamos la última ventana correspondiente a las t últimas filas volveremos a coger la ventana del principio, es decir, desde la posición 0 a la posición $t - 1$.

4.3. Tipos de ventana.

A la hora de analizar la Pila, podremos encontrarnos con tres tipos de ventana.

1. Ventana vacía.

Serán ventanas en las que no aparece ningún objeto, simplemente captaremos la superficie del río. Este tipo de ventana será seguramente el que más nos encontremos ya que se supone que la frecuencia de paso de objetos no será demasiado alta. Además, serán un tipo de ventana imprescindible en cuanto a caracterización del entorno ya que nos ayudará a conocer cómo es la superficie del río y posteriormente a poder diferenciar los objetos de ésta.



Figura 6. Ventana vacía.

2. Ventana con objeto entero.

Serán ventanas en las que además de la superficie del río aparecerán uno o varios objetos. Estas serán las ventanas interesantes en cuanto a detección ya que son las que contienen los objetos.



Figura 7. Ventana con objeto entero.

3. Ventana con objeto incompleto.

Puede ocurrir que un objeto aparezca partido por el hecho de coger ventanas de un tamaño definido, de forma que nos aparezca la mitad en una ventana y la otra mitad en la siguiente.



Figura 8. Ventana con objeto incompleto.

4.3.1. Solución a los objetos incompletos.

La solución que se ha planteado para resolver este problema es coger las ventanas con cierto solapamiento, de forma que, si en la ventana actual sólo sale la mitad del objeto, en la siguiente aparecerá entero. En este caso, se ha cogido un solapamiento de la mitad, es decir, si cogemos ventanas de tamaño t , el solapamiento entre ventanas será de $t/2$.



Figura 9. Esquema del inventariado con solapamiento.

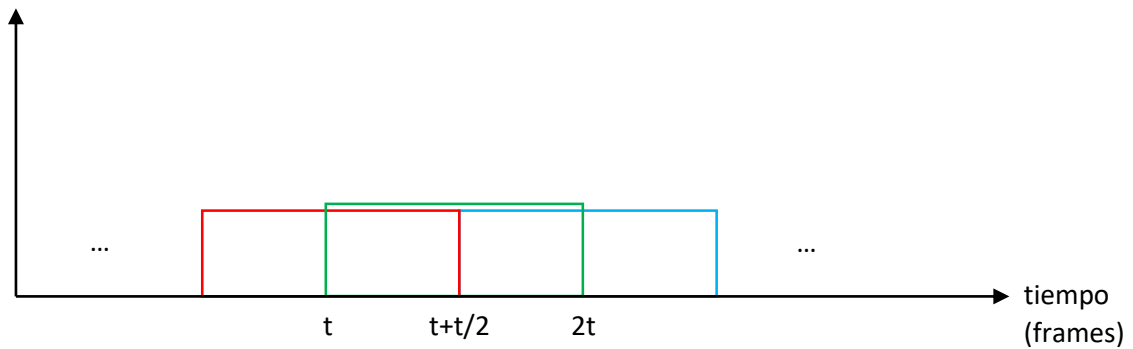


Figura 10. Esquema del solapamiento entre ventanas.

En la siguiente figura se puede observar un ejemplo de un objeto que aparece partido y cómo la mitad inferior de la ventana anterior se corresponde con la mitad superior de la ventana actual.

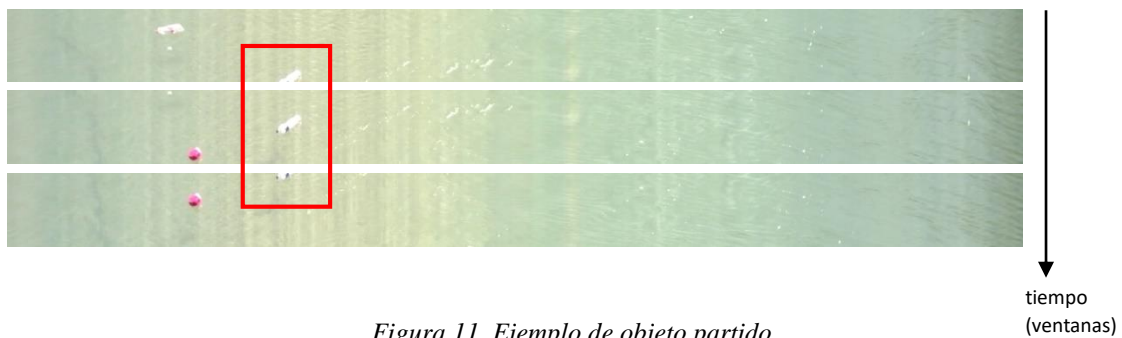


Figura 11. Ejemplo de objeto partido.

4.4. Ventajas de procesar una sola línea de cada fotograma.

En primer lugar, la detección y seguimiento de objetos en movimiento suele ser un problema planteado en 3 dimensiones: x , y y *tiempo*; además suele exigir un alto rendimiento de procesado y un consumo de recursos bastante grande. La propuesta de elegir una línea del fotograma reduce este problema a 2 dimensiones: x y *tiempo*, haciendo que la exigencia de recursos y rendimiento sea menor y consiguiendo el mismo objetivo: detectar objetos en movimiento.

Un ejemplo del ahorro considerable de recursos es la cantidad de píxeles de cada fotograma que se analizan. Por ejemplo, si el vídeo tiene un tamaño de 1270x720, esto suman un total de 914.400 píxeles, mientras que una única línea está formada por 1270 píxeles, lo que es un 0.1389% del total, es decir, utilizamos un 99.8611% menos de información.

Además, coger una sola línea nos permite seleccionar la parte de la imagen que nos interese y sea más beneficiosa para la detección. Como se puede ver en la Figura 5, en el vídeo no sólo aparece el agua del río, sino que también se ve la orilla y otros elementos que no nos interesan, por tanto, escoger una línea nos da la posibilidad de elegir un entorno favorable para la detección.



Figura 12. Fotograma del vídeo.

Capítulo 5. Desarrollo.

5.1. Aprendizaje. Modelado estadístico de los píxeles.

5.1.1. Obtención de la Media y la Desviación Típica.

Para poder detectar objetos diferentes de la superficie del río se debe conocer primero cómo es dicha superficie, es decir, caracterizarla. Para ello, se recopilan una serie de estadísticas que se utilizarán posteriormente en el algoritmo de detección.

Esta primera fase se debe realizar con un vídeo en el que estemos seguros de que no aparezcan objetos de forma que no quede registrado ningún objeto en las estadísticas, sino que únicamente quedará la superficie del río.

Para el cálculo de dichas estadísticas se necesita calcular tanto la media vertical de los píxeles de las ventanas como la desviación típica de los mismos.

Cada vez que se obtiene la ventana correspondiente de la Pila, que es una imagen RGB, es decir, tiene 3 canales: Rojo (R), Verde (G) y Azul (B); se cogerá únicamente el canal G debido a que es el que más relación tiene con la Luminancia (Y) como podemos ver en su respectiva fórmula.

$$Y = 0.299 * R + 0.587 * G + 0.114 * B \quad (1)$$

Una vez se tiene el canal verde de la ventana, esto es, una imagen con un solo canal, se procederá a realizar los pasos de obtención de las estadísticas.

En primer lugar, se realiza un promediado vertical de cada columna de la ventana, de forma que se obtendrá un vector de tamaño el ancho de la ventana. A este vector lo llamaremos $\bar{x}(n)$, es decir, será el vector promedio de las columnas de cada ventana, donde n es la ventana que se está analizando.

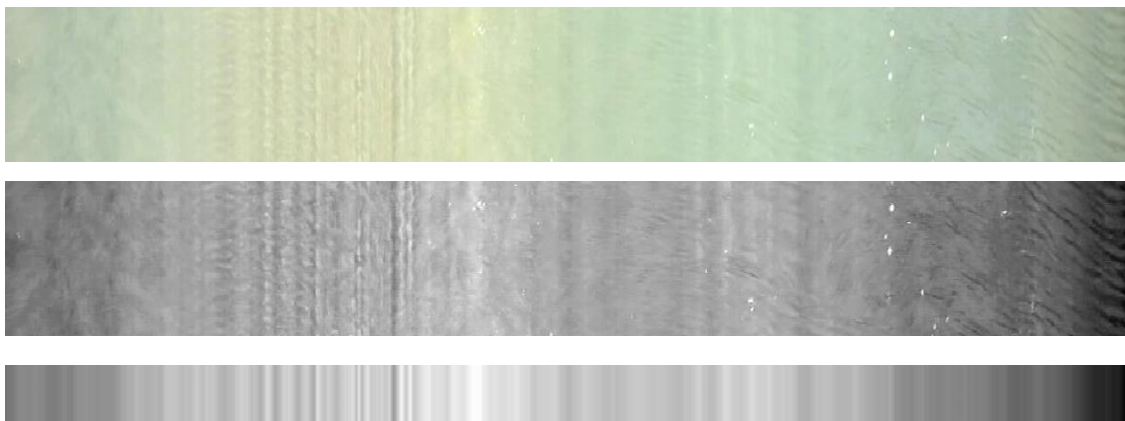


Figura 13. En orden de arriba abajo: Ventana - Canal G de la ventana - Media vertical $\bar{x}(n)$

$$\bar{f}(x) = \frac{\sum_{t=0}^{T-1} f(x, t)}{T} \quad (2)$$

Donde $f(x, t)$ es cada uno de los píxeles de la ventana y T es la altura de la ventana.



Figura 14. Ejes de una ventana.

Al mismo tiempo, obtendremos la media al cuadrado, que será utilizada posteriormente para calcular la desviación típica de la distribución.

$$\bar{f}^2(x) = \frac{\sum_{t=0}^{t=T-1} f(x, t)^2}{T} \quad (3)$$

Una vez se ha calculado la media de cada columna de cada ventana, se realiza un promediado de las distintas ventanas para obtener la *Media Total* (μ) y la *Media Cuadrática Total* (μ_2). Es en este momento cuando se calcula la *Varianza* y a continuación la *Desviación Típica*.

$$\mu = \frac{\sum_{i=0}^{i=N-1} \bar{f}(x, t)}{N} \quad (4)$$

$$\mu_2 = \frac{\sum_{i=0}^{i=N-1} \bar{f}^2(x, t)}{N} \quad (5)$$

$$\sigma = \sqrt{Var} = \sqrt{\mu_2 - \mu^2} \quad (6)$$

Donde N es el número total de ventanas que se han analizado.

Obteniendo estos datos se acaba de caracterizar el sistema como una distribución normal, es decir, cada columna de la ventana tendrá su media y su desviación típica.

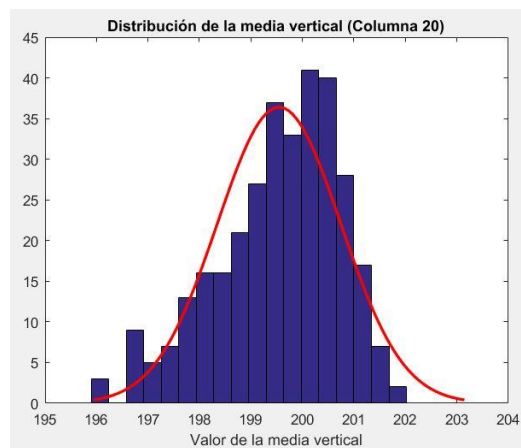


Figura 15. Ejemplo de distribución de una de las columnas.

Una vez obtenidos los vectores Media vertical (μ) y Desviación típica (σ), el siguiente paso es obtener la estadística de objetos que aparecen en las ventanas.

5.1.2. Obtención de la matriz de probabilidades.

En primer lugar, obtendremos el canal G de la ventana al igual que se ha explicado anteriormente. A cada fila de la ventana le restaremos la Media (μ) y la dividiremos entre la Desviación Típica (σ). Haciendo esto, se convierte la distribución normal de media μ y desviación típica σ , en una distribución normal con media 0 y desviación típica 1, de modo que aquellos píxeles cuyos valores que no sean muy diferentes de la media estarán cercanos a 0.

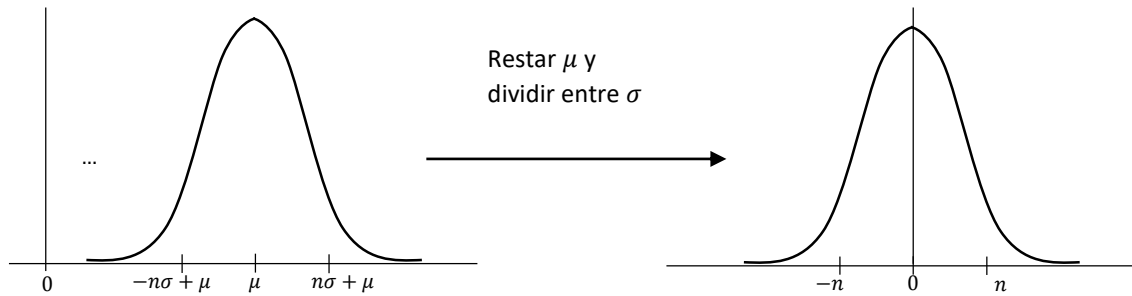


Figura 16. Representación del desplazamiento y normalización de la función de distribución.

Para hacer más sencillo el análisis se calcula el valor absoluto de los valores haciendo que únicamente se tengan valores en la parte positiva del eje x.

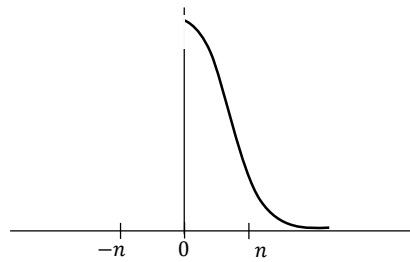


Figura 17. Valor absoluto de la distribución normal.

Al aplicar este procedimiento a las ventanas se obtienen las imágenes que aparecen a continuación.



Figura 18. Aplicación de la normalización a una ventana.

En orden descendente: Canal G – Diferencia – Diferencia Normalizada – Valor absoluto de la Diferencia Normalizada

Como se puede observar, en la última imagen aparecen los píxeles que más se parecen a la media en un tono más oscuro, mientras que aquellos que son muy diferentes aparecen mucho más brillantes. Esto es porque son objetos que pasan flotando por el río y que no pertenecen al fondo. Sin embargo, tampoco son objetos que nos interese diferenciar del fondo ya que no son botellas, bolsas, envases o cualquier otro objeto de este tipo que son los que realmente queremos detectar. Estos puntos brillantes suelen ser espuma que se forma debido principalmente a la corriente del río, como también pueden ser cualquier otro tipo de elemento flotante propio del río pero que no forma parte del fondo estático del río.

El siguiente paso es recopilar información sobre todos estos píxeles más brillantes. Para ello, se binariza la imagen “*Valor absoluto de la Diferencia Normalizada*”.

Para binarizar la imagen se necesita un umbral. Dicho umbral se ha elegido en base a la Función Error, que permite conocer cuál es el porcentaje de valores de la distribución normal está a una distancia, múltiplo de la desviación típica, de la media. Concretamente, el área bajo la curva campana viene dada por la siguiente expresión:

$$\Phi_{\mu,\sigma^2}(\mu + n\sigma) - \Phi_{\mu,\sigma^2}(\mu - n\sigma) = \Phi(n) - \Phi(-n) = 2\Phi(n) - 1 = \operatorname{erf}\left(\frac{n}{\sqrt{2}}\right) \quad (7)$$

Si se da valores a n se obtendrá el porcentaje de valores que habrá debajo de la curva a una distancia $n\sigma$ de la media. Gracias a que se ha realizado el procedimiento de normalización de la distribución, como se ha explicado anteriormente, el valor n es directamente la distancia a la que se encuentran los valores de la distribución del 0. En la siguiente tabla se encuentran algunos valores de n y sus respectivos porcentajes [6].

n	$\operatorname{erf}(n/\sqrt{2})$
1	0,682689492137
2	0,954499736104
3	0,997300203937
4	0,999936657516
5	0,999999426697

Tabla 1. Valores de la Función Error.

Para este caso se ha elegido el un umbral de $n = 2$, es decir, el 95,45% de los valores se considerará que no están muy lejos de la media y el 4,55% restante se considerarán objeto. Dicho de otra manera, los píxeles de la imagen “*Valor absoluto de la Diferencia Normalizada*” cuyo valor sea superior a 2 se considerará que no forman parte del fondo estático del río.

La imagen resultante después de la binarización se ve en la Figura 13.



Figura 19. Ventana binarizada.

Como se puede observar, únicamente quedan aquellos píxeles con valor por encima de 2.

Llegados a este punto, el siguiente paso es clasificar los objetos que aparecen en las ventanas binarizadas. Para ello se han caracterizado dichos objetos mediante dos propiedades: posición y área.

La posición de un objeto se corresponderá a las coordenadas del píxel que se considere centro de dicho objeto mientras que el área será el número de píxeles que conforman el mismo.

Una vez se conocen estos dos descriptores de cada objeto se procede a clasificarlos dentro de un histograma tridimensional cuyo eje vertical se corresponderá con el eje de las áreas y el horizontal con el eje de posiciones, y cuya tercera dimensión será la cantidad de objetos que tienen misma área y misma posición.

A la hora de colocar un objeto en el histograma se ha dividido la ventana en un número de intervalos $n_{intervalos}$, en este caso se ha elegido $n_{intervalos} = 254$, de forma que para la posición diremos que un objeto cuyo centro está, por ejemplo, en la columna 20 y otro que esté en la columna 22, se considerará que está dentro de la franja $x = 4$, suponiendo que el ancho de ventana es 1270. Por tanto, la posición dentro del histograma se calculará mediante la siguiente fórmula:

$$x = \frac{columna}{ancho\ ventana} * n_{intervalos} \quad (8)$$

Por otro lado, el eje de las áreas es un eje logarítmico. Esto es debido a que no es igual de significativa la diferencia de tamaño entre un objeto de área 2 y un objeto de área 3; y un objeto de área 200 y otro de área 201, es mucho más significativa entre objetos pequeños de áreas pequeñas.

Se establece un área mínima (A_{min}) y un área máxima ($A_{máx}$) del eje del histograma para delimitarlo. El área mínima será 1 ya que el objeto más pequeño que nos podemos encontrar tiene tamaño 1 píxel. El área máxima se ha delimitado en 600 píxeles.

Para conocer la posición en el eje de áreas se utiliza la siguiente expresión (9).

$$A = \frac{\log(\text{área}_{objeto}) - \log(\text{área}_{mínima})}{\log(\text{área}_{máxima}) - \log(\text{área}_{mínima})} * n_{intervalos\ de\ área} \quad (9)$$

Por tanto, cada vez que aparezca un objeto y conozcamos su centro y su área, sumaremos una unidad en la posición (x, A) correspondiente del histograma.

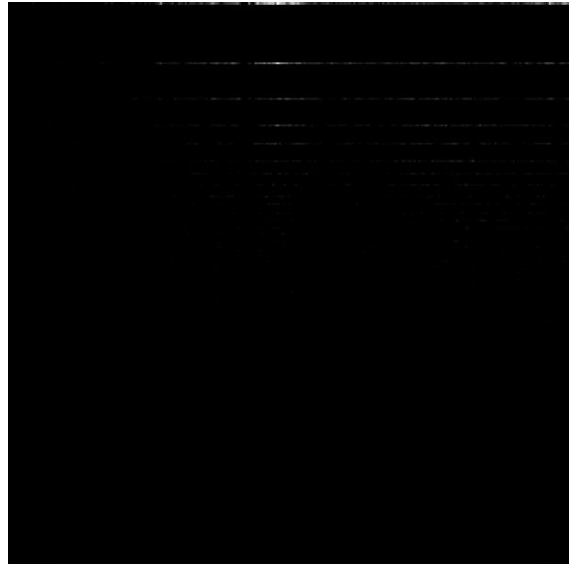


Figura 20. Histograma.

Si aparece un objeto cuya área es mayor que el área máxima definida, se supondrá que el área de dicho objeto será igual al área máxima.

$$\text{área} = \min(\text{área}_{objeto}, A_{máx}) \quad (10)$$

Una vez se tiene completo el histograma, se procede a realizar un suavizado del mismo con el objetivo de aproximar la Función Densidad de Probabilidad (FDP). Para ello se ha seguido el método de estimación de la FDP propuesta por Emanuel Parzen en 1962, que consistía en aplicar un filtrado gaussiano sobre cada una de las muestras de una variable aleatoria, de forma que la suma de la contribución de cada gaussiana en cada muestra resultará en la función densidad de probabilidad de la distribución. Cuantas más muestras tuviera la distribución más correcta sería la aproximación de la FDP [7].

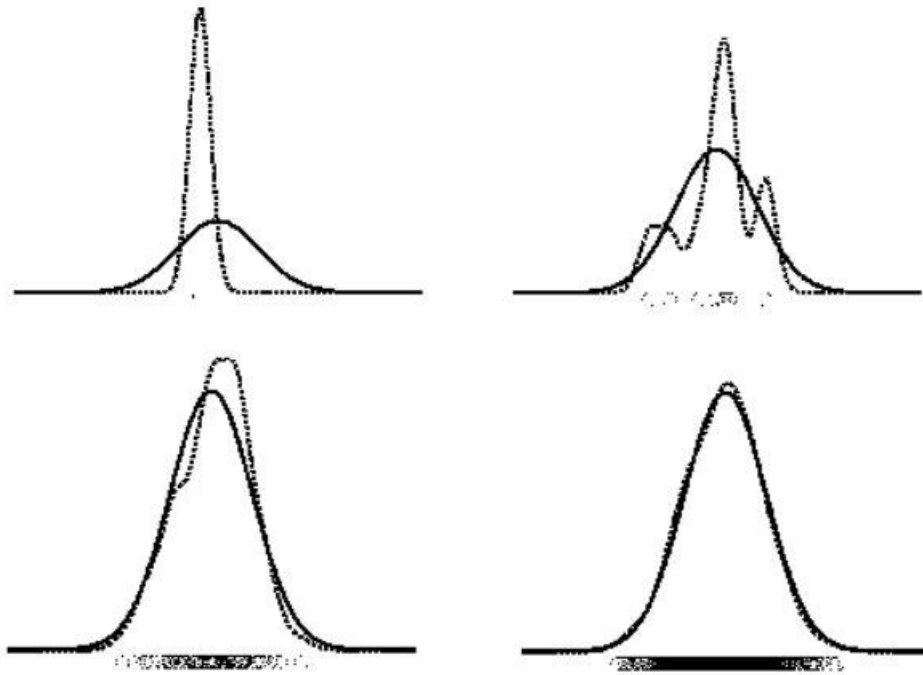


Figura 21. Aproximación de la FDP mediante ventana de Parzen.

En orden de izquierda a derecha y de arriba abajo: 1 muestra - 10 muestras - 100 muestras - 1000 muestras. [8]

Para conseguirlo, se realizará un suavizado gaussiano tanto en el eje de las posiciones como en el eje de las áreas, eso sí, en cada eje la máscara tendrá su respectiva desviación típica (σ_A y σ_x).

Para calcular σ_x tendremos en cuenta la influencia que queremos que tengan los objetos que hayamos encontrado en sus vecindarios, es decir, si se ha encontrado un objeto O en x , consideraremos que en $x + \Delta$ y $x - \Delta$ probablemente también aparezcan objetos del mismo tamaño que el de O . Esta influencia que queremos que tenga en los píxeles de alrededor será σ_x . Sabiendo que la anchura de la máscara es aproximadamente 6σ tendremos que igualarlo a su equivalente en píxeles, como observamos en (10).

$$6\sigma = n_{\text{píxeles}} \quad (11)$$

donde $n_{\text{píxeles}}$ es el número de píxeles vecinos de x sobre los cuales queremos que tenga influencia.

Teniendo en cuenta que las x en el histograma corresponden al ancho de franja (ancho_x) de la ventana original, un píxel del histograma representa a ancho_x píxeles de la imagen original, por tanto, la sigma calculada resulta

$$\sigma_x = n_{\text{píxeles}}/6 \quad (12)$$

De este modo, el filtrado afectará a los píxeles situados en el intervalo $[x - ancho_x, x + ancho_x]$.

Para el cálculo de σ_A se sigue el mismo razonamiento que para σ_x , teniendo en cuenta que el eje de las áreas es logarítmico, lo que resulta la siguiente expresión.

$$\sigma_A = \frac{1}{6} * \log \frac{A}{B} * \frac{n_{intervalos}}{\log(A_{m\acute{a}xima}) - \log(A_{m\acute{i}nima})} \quad (13)$$

Donde el cociente A/B representa el vecindario sobre el que queremos que tenga influencia la máscara.

El resultado de aplicar el suavizado al hisotgrama se puede observar en la Figura 15.

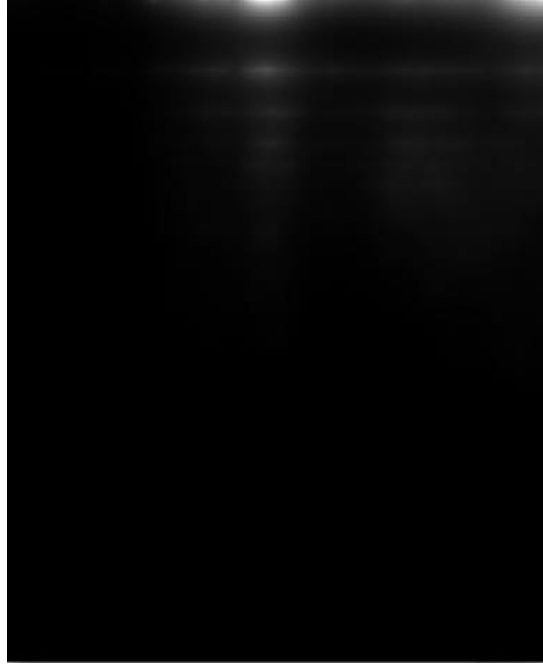


Figura 22. Histograma suavizado.

Una vez hemos estimado la FDP y la normalizamos dividiendo por la suma total de la matriz, obtenemos la probabilidad de aparición de un objeto con un área y una posición condicionada a que es fondo, y no objeto. A esta matriz la llamaremos $P(A, x | F)$.

Sin embargo, la probabilidad que nos interesa es justamente la contraria, es decir, la probabilidad de ser fondo condicionada a un área y una posición ($P(F | A, x)$). Si acudimos al teorema de Bayes [9], la expresión para obtener dicha probabilidad sería,

$$P(F | A, x) = \frac{P(A, x | F) * P(F)}{P(A, x | F) * P(F) + P(A, x | \bar{F}) * P(\bar{F})} \quad (14)$$

Como no conocemos cuál es la probabilidad de ser fondo $P(F)$ ni la de que no serlo $P(\bar{F})$, suponemos que son iguales. De este modo podemos simplificar (13) obteniendo la expresión,

$$P(F | A, x) = \frac{P(A, x | F)}{P(A, x | F) + P(A, x | \bar{F})} \quad (15)$$

Al no conocer $P(A, x | \bar{F})$ la supondremos uniforme, de modo que si $P(A, x | \bar{F}) \gg P(A, x | F)$ entonces $P(F | A, x) \approx 0$, mientras que si $P(A, x | \bar{F}) \ll P(A, x | F)$ entonces $P(F | A, x) \approx 1$.

Por tanto, al conocer la probabilidad de que sea fondo $P(F | A, x)$ también podremos conocer la probabilidad de objeto que simplemente será

$$P(O | A, x) = 1 - P(F | A, x) \quad (16)$$

Llegados a este punto ya hemos obtenido los tres datos que necesitamos para proceder a la detección: la Media (μ), la Desviación Típica (σ) y la matriz de probabilidades $P(O | A, x)$.

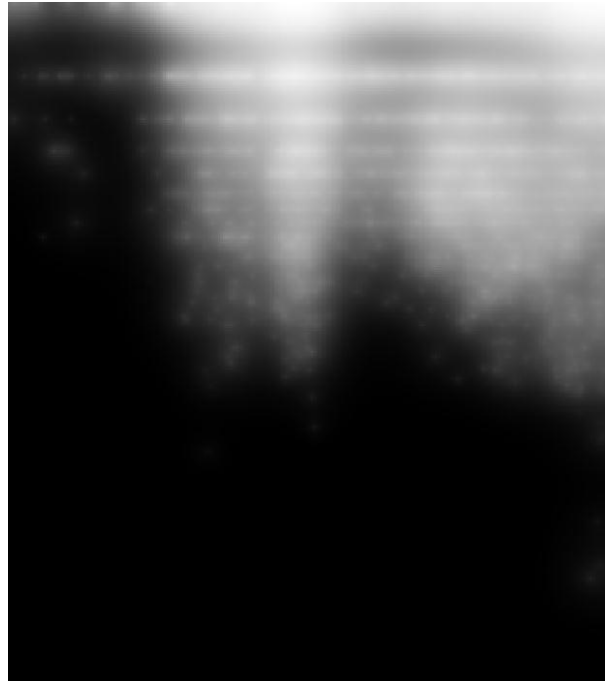


Figura 23. Matriz de Probabilidades.

5.2. Proceso de detección.

Una vez tenemos la matriz de probabilidades (ρ) se puede empezar a procesar imágenes en las que sí aparezcan objetos. Para ello, primero se deben tener a mano los vectores Media (μ) y Desviación Típica (σ) que se han obtenido en el proceso de caracterización del entorno.

El proceso de detección comienza del mismo modo que el de obtención de la matriz de probabilidades. En primer lugar se obtiene el canal G de la ventana y se calcula la imagen Valor absoluto de la diferencia normalizada utilizando μ y σ . A continuación, se binariza la imagen utilizando el mismo umbral que se había utilizado anteriormente, en este caso $umbral = 2$. A partir de aquí es donde comienza el proceso real de detección.

El primer paso es localizar todos los objetos que aparecen en la imagen binaria y conocer de cada uno de ellos su centro y su área.

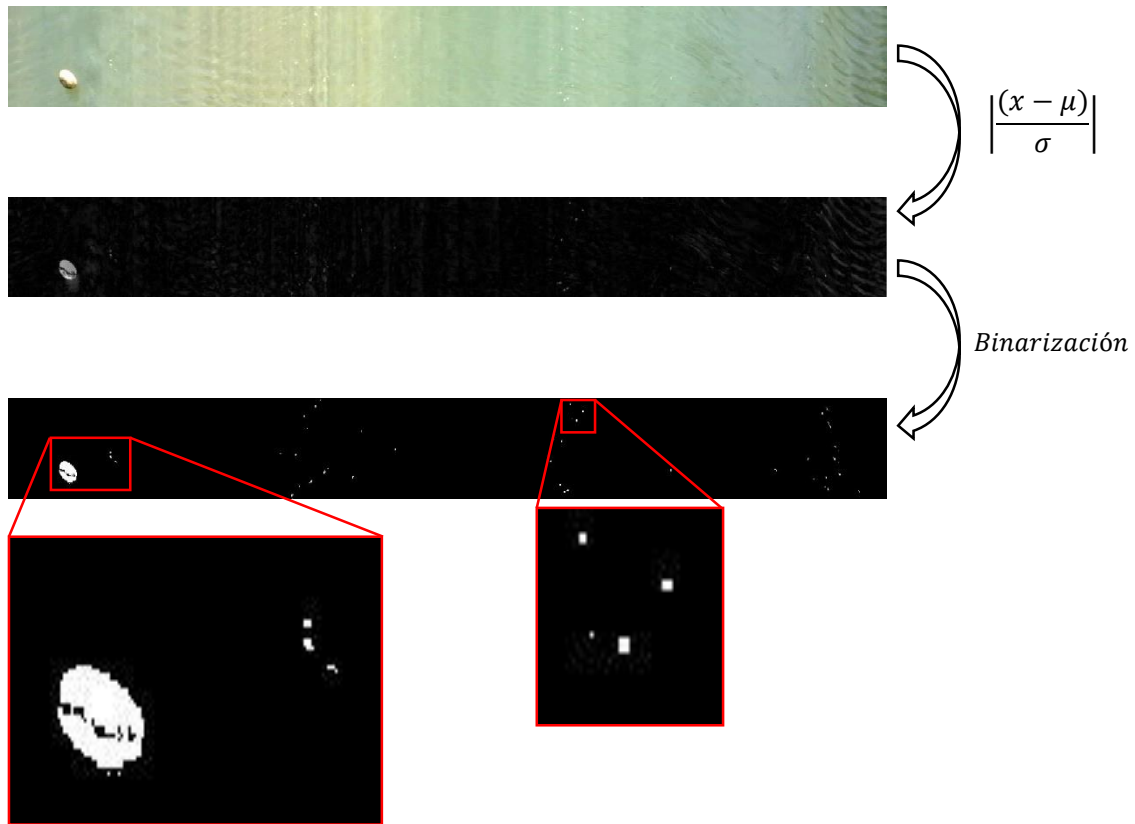


Figura 24. Proceso hasta el binarizado y detalle de los objetos.

Sabiendo el centro y el área de un objeto se acude a la matriz de probabilidades para conocer cuál es la probabilidad de que dicho objeto sea o no un objeto definitivamente. De modo que se crea una nueva imagen igual que la binaria con la única diferencia de que los píxeles de los objetos no valdrán 1 sino que tomarán el valor de la probabilidad de que dicho objeto sea un objeto. Esto se puede apreciar en la Figura 22.

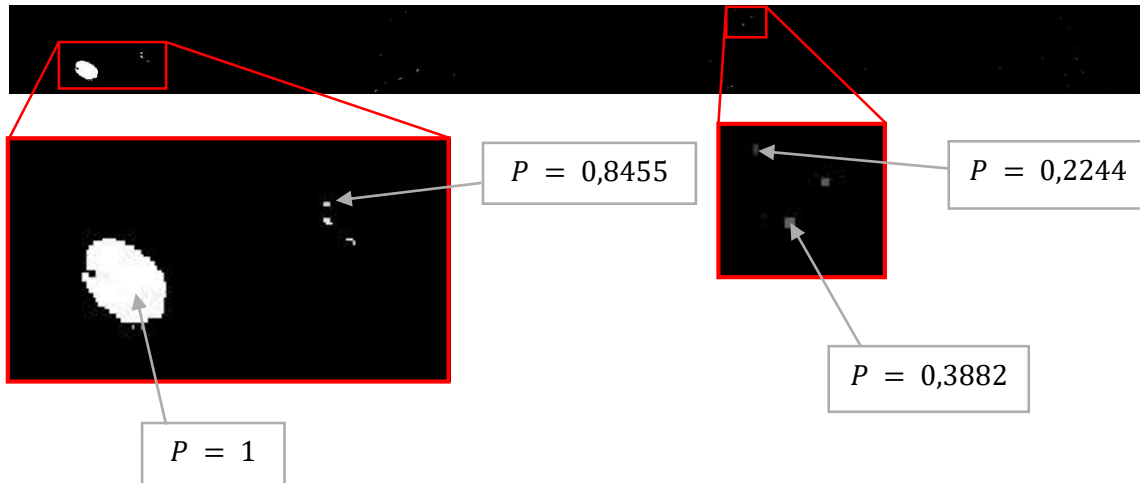


Figura 25. Detalle de la probabilidad de cada objeto.

Llegados a este punto, es el momento de elegir un umbral para filtrar aquellos objetos que son espuma u otros elementos que no son objetos y quedarnos únicamente con los objetos. Este umbral se ha decidido mediante ensayo y error, viendo qué valores de probabilidad han ido teniendo los distintos objetos que han pasado flotando. Se ha concluido, en base a los ensayos realizados y al tiempo de aprendizaje que ha tenido el programa, que el umbral de probabilidad óptimo para este caso es $Umbral_p = 0,96$.

Hay que tener en cuenta que este umbral se ajusta mejor cuando mayor es el tiempo de aprendizaje ya que se caracteriza mucho mejor el sistema debido a que la cantidad de datos recogidos es mayor y por tanto la FDP se estima mucho mejor.

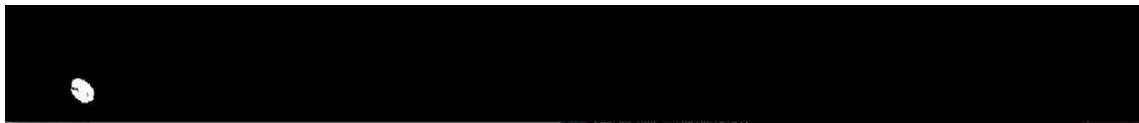


Figura 26. Resultado de aplicar el umbral de probabilidad.

Como se puede observar, la imagen resultado es una imagen binaria en la que todos aquellos píxeles que hayan superado el umbral de probabilidad valen 1 y aquellos que no lo han superado pasan a formar parte del fondo tomando el valor 0.

Por último, una vez tenemos una imagen binaria en la que sabemos con seguridad que los elementos que aparecen son objetos, se actualiza el contador de objetos detectados sumándole el número de objetos que aparecen en la ventana.

5.3. Tipos de algoritmo.

5.3.1. Algoritmo offline.

El algoritmo offline se utiliza para analizar videos ya grabados. Para poder hacer funcionar este algoritmo correctamente necesitaremos un vídeo en el que únicamente se vea el río, es decir, que no pasen objetos; con el cual estimar las estadísticas y obtener la matriz de probabilidades. Para esto habrá que realizar un primer análisis del vídeo con el que obtendremos os vectores μ y σ , y posteriormente un segundo análisis del mismo vídeo para sacar la matriz de probabilidades.

Una vez obtenidas las matrices, se aplica el algoritmo de detección al vídeo en el que aparecen objetos.

Este es el algoritmo que se ha seguido en este proyecto, debido a que no ha sido viable la realización de experimentos en tiempo real ni la instalación de ningún dispositivo en los ríos.

5.3.2. Algoritmo online.

También podríamos referirnos a este algoritmo como algoritmo en tiempo real ya que, a diferencia del online, en este no utilizamos vídeos grabados, sino que tomamos las imágenes en tiempo real y se van analizando los fotogramas conforme se van capturando.

El proceso de inventanado de la Pila es el mismo, la principal diferencia entre algoritmos es la obtención de las matrices μ , σ y ρ .

En el caso online, desde el momento en que se pone en marcha la captura de vídeo, hay un primer periodo de tiempo que se emplea para la caracterización del entorno obteniendo las matrices Media y Desviación Típica. A continuación, sucede un segundo periodo empleado para calcular la matriz de Probabilidades. Una vez obtenidos los datos necesarios se procede a la detección y aquí es donde aparece la diferencia con el algoritmo offline.

Como el objetivo final de este algoritmo es el trabajo en tiempo real, llegará un momento en el que las estadísticas obtenidas en un primer instante quedarán inservibles ya que ha pasado mucho tiempo desde su obtención.

Para solucionar este problema, lo que se hace es refrescar las estadísticas con cada ventana que analicemos y recalculamos la matriz de probabilidades cada vez. Este refresco lo haremos aplicando un factor de olvido pequeño, de forma que las nuevas medias que obtengamos influyan en la media total menos que todas las medias obtenidas anteriormente.

$$\mu_n = \alpha * \bar{f}(x) + (1 - \alpha) * \mu_{n-1} \quad (17)$$

Donde α es el factor de olvido.

Esta fórmula se aplica del mismo modo a la media cuadrática, de forma que cada vez que actualicemos las medias recalcularemos la Desviación Típica y la matriz de Probabilidades.

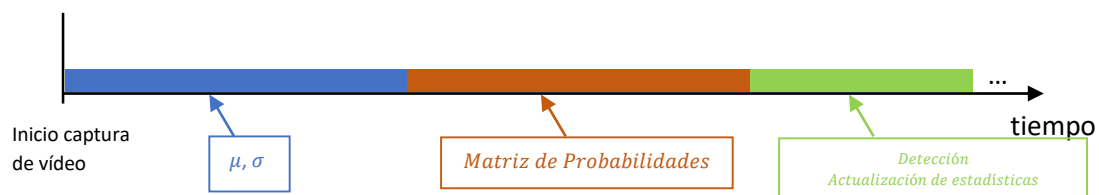


Figura 27. Diagrama temporal del algoritmo online

Capítulo 6. Resultados.

6.1. Grabación de vídeo.

Para probar el algoritmo realizado se han hecho varias grabaciones de vídeo en distintas localizaciones y desde diferentes posiciones. Los ríos elegidos para ello han sido el Túria y el Júcar ya que ofrecían la posibilidad de realizar los experimentos permitiendo recoger los objetos lanzados, además de ser ríos accesibles y dentro de la provincia en la que se ha realizado el estudio.

Los ensayos consistieron en lanzar diferentes tipos de objetos que flotaran en el agua de forma que fueran arrastrados por la corriente y fuesen captados por la cámara.





Figura 28. Fotogramas con objeto.

Los vídeos de los ensayos realizados están accesibles en YouTube a través de los siguientes links:

Ribaroja 1
<https://youtu.be/WrF8o9ECp4k>

Ribaroja 3
https://youtu.be/_r7BEykhfg

Albalat - Orilla
<https://youtu.be/5kuME04uCI>

Albalat - Puente
https://youtu.be/O_HddA70sxo



6.2. Colocación de la cámara.

Se probaron dos posiciones en las que colocar la cámara a la hora de grabar:

- Encima del río desde un puente, de forma que la cámara capture toda la anchura del río desde arriba y la corriente avance en sentido vertical.



Figura 29. Vista del río desde arriba.

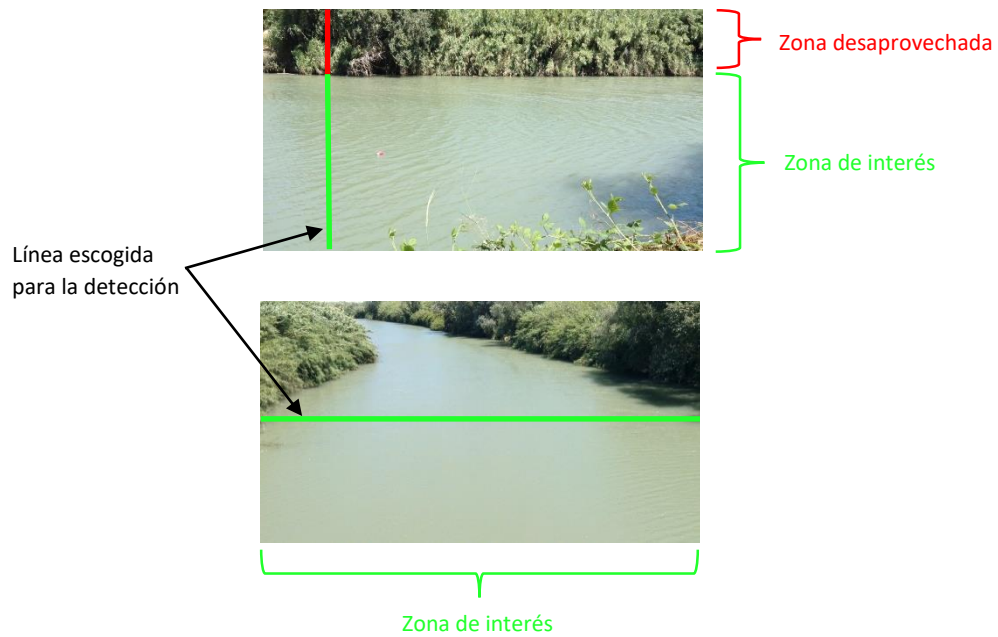
- En una orilla del río, viéndolo de lado y con la corriente avanzando de izquierda a derecha o viceversa.



Figura 30. Vista del río desde la orilla.

Al colocar la cámara a un lado del río, en lugar de coger líneas horizontales como hemos hecho hasta ahora, se deben coger líneas verticales por el hecho de que la corriente va en dirección horizontal. El problema de formar la pila a partir de líneas verticales situando la cámara en la orilla es que la porción de superficie del río que aparece en la línea disminuye considerablemente con respecto a la vista desde arriba (Figura 31). Por tanto, aparecen en la Pila partes que no son interesantes para la detección, como puede ser

vegetación, parte del cielo, la otra orilla, o cualquier otro elemento que no es el agua del río. Además, las orillas suelen ser bastante inaccesibles por tanto es mucho más difícil colocar un trípode o cualquier estructura para sujetar la cámara.



*Figura 31. Comparativa de colocación de la cámara.
Arriba: Vista desde la orilla. Abajo: Vista desde arriba.*

En resumen, se ha decidido como mejor opción grabar desde arriba del río por tres razones:

- Vista completa del ancho del río.
- Se evita la superposición de unos objetos con otros.
- Facilidad de colocación del equipo de grabación.

6.3. Detección de objetos.

A continuación, se muestran los resultados obtenidos de aplicar nuestro algoritmo de detección a una serie de vídeos por el que pasan diferentes objetos.

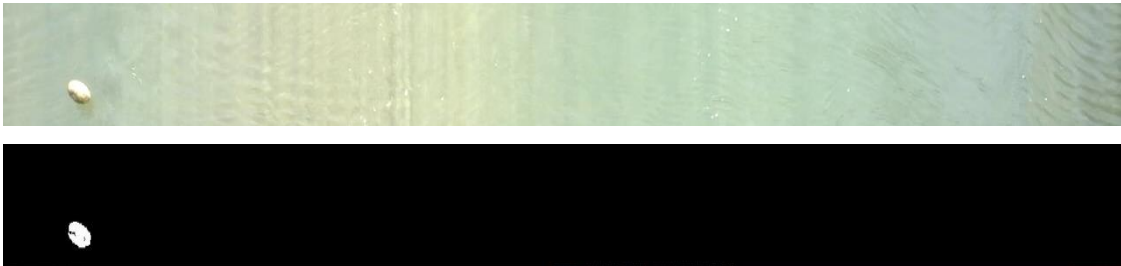


Figura 32. Pelota.

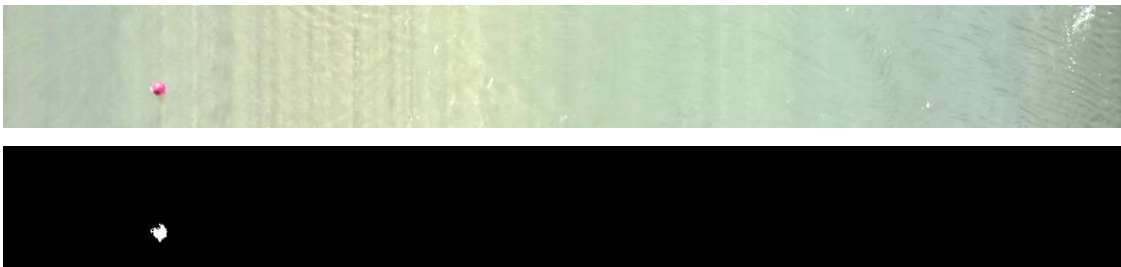


Figura 33. Bote de plástico opaco.

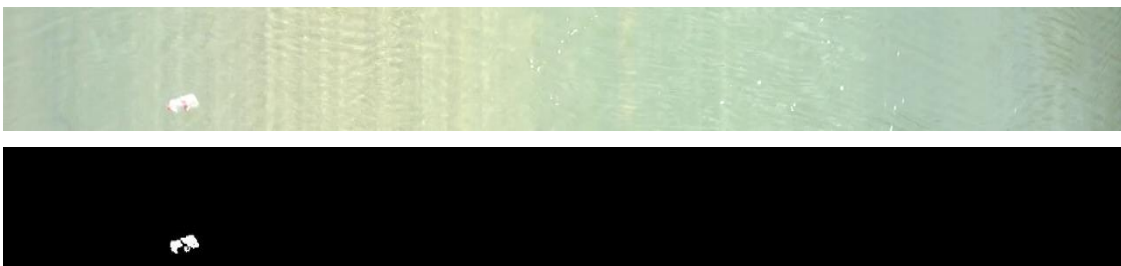


Figura 34. Botella de plástico transparente.

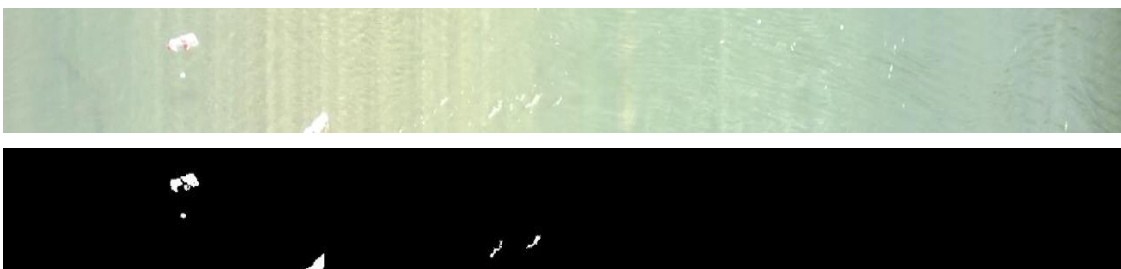


Figura 35. Botellas de plástico transparente.



Figura 36. Bote de plástico opaco y botella de plástico transparente.

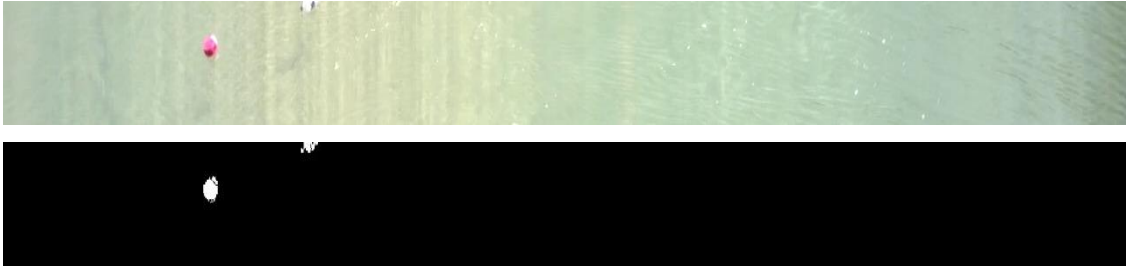


Figura 37. Bote de plástico opaco y botella de plástico transparente.

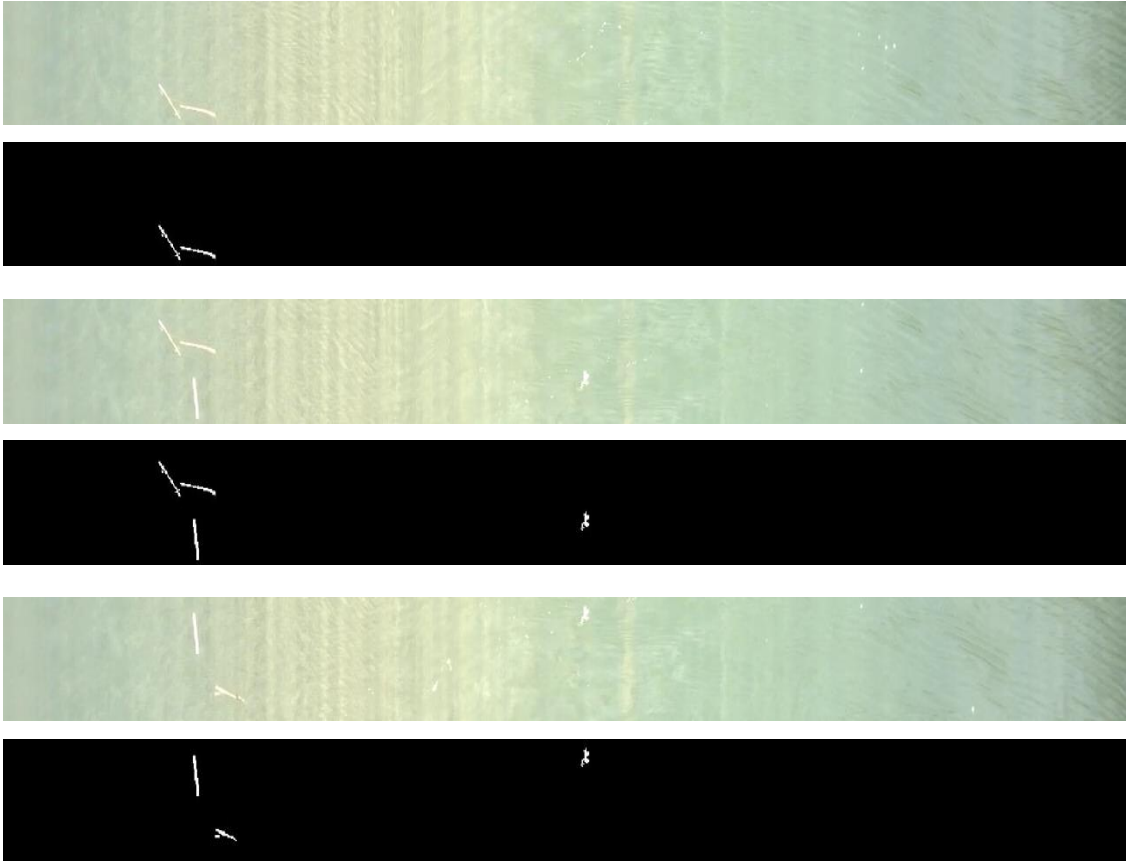


Figura 38. Palos de madera.

6.4. Fiabilidad del algoritmo.

En algunas de las imágenes (Figuras 32 a 38) podemos ver que aparecen elementos que no son objetos pero que se diferencian lo suficiente del fondo como para ser detectados. Esto se puede deber a tres razones:

- Se necesita más tiempo de caracterización del entorno para que puedan aparecer elementos como espuma, hojas, reflejos del agua, etc., y queden registrados en las estadísticas, de forma que cuando aparezcan luego tengamos unas estadísticas sólidas y obtenidas a partir de muchas muestras.
- Al realizar los ensayos lo que se ha hecho ha sido lanzar los objetos desde una altura considerable, lo que puede provocar que al caer el objeto sobre el agua genere espuma de forma que esta espuma cruce la línea al mismo tiempo que los objetos y sea detectada como uno más, o también que se creen reflejos artificiales en el agua debido a las ondas producidas por el objeto al impactar en el agua. Si vemos ventanas en las que no aparecen objetos podemos observar que no encontraremos conjuntos tan grandes de espuma.



Figura 39. Ventanas estándar sin objetos.

En un caso real esto no ocurriría ya que los objetos no serían lanzados, sino que llegarían flotando y por tanto no provocarían ningún tipo de distorsión sobre la superficie del agua.

- Que aparezca espontáneamente alguna irregularidad del tipo espuma o reflejo del agua que no había aparecido nunca y por tanto no podamos prever.

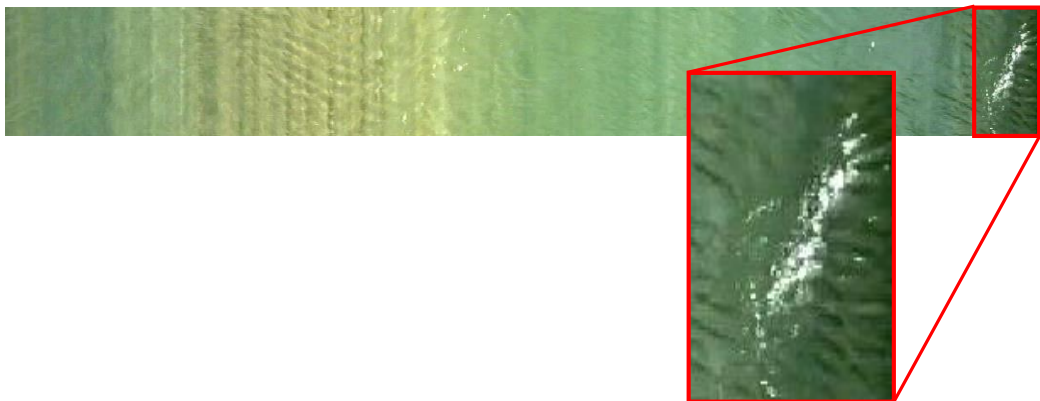


Figura 40. Formación de espuma espontánea.

6.5. Situaciones a tener en cuenta.

6.5.1. Problemática de los objetos transparentes.

Gran parte de los residuos flotantes de los torrentes de agua suelen ser plásticos, y muchos de ellos pueden ser transparentes. Si nos encontramos este tipo de objetos arrastrados por la corriente puede dar lugar a dos situaciones:

- Que de la casualidad que al atravesar el objeto la línea del fotograma refleje la luz que le incide y se queme la imagen, es decir, que el objeto en lugar de ser transparente sea total o parcialmente blanco. Si esto ocurre el objeto se detectará correctamente ya que no tendrá el color de la superficie del agua.
- Que al cruzar la línea no refleje la luz y por tanto sea transparente y se vea la superficie del agua impidiendo que se detecte dicho objeto. Es muy difícil que todo el objeto aparezca totalmente transparente en la imagen, normalmente siempre habrá parte de él o partes que reflejen la luz y aparezcan blancas, de modo que se detecte el objeto de forma incompleta.

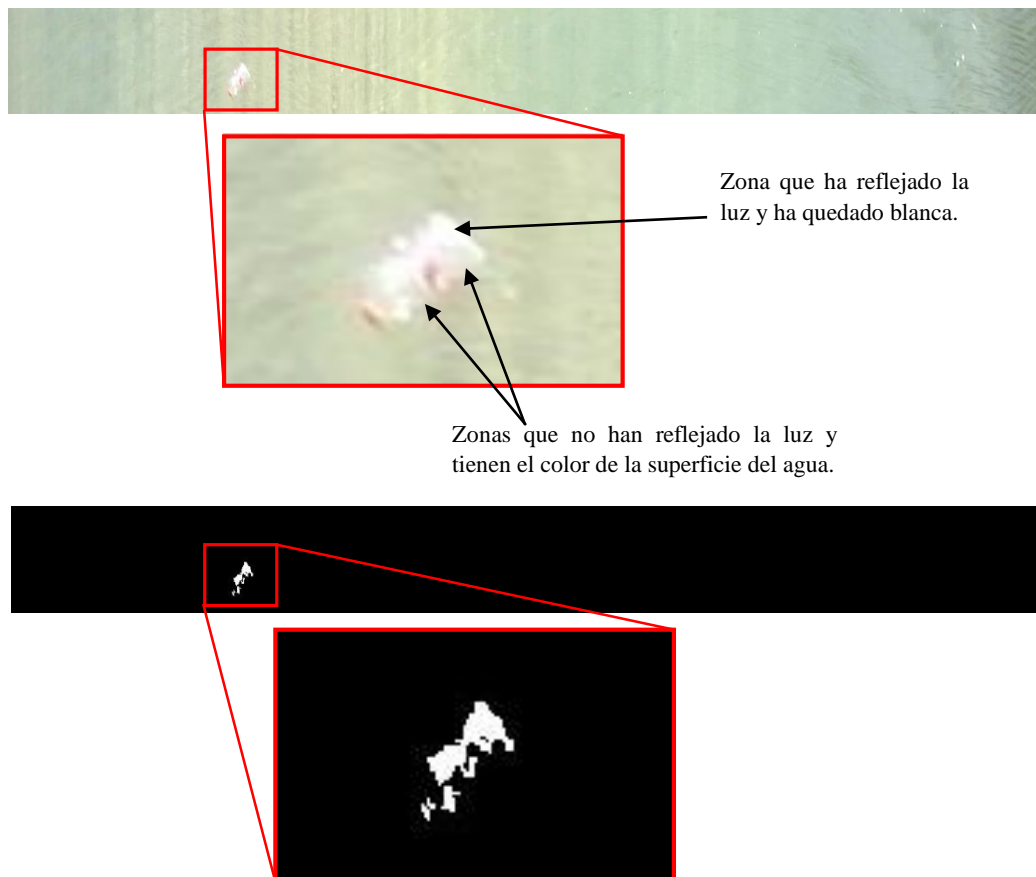


Figura 41. Arriba: Ventana con objeto transparente y detalle de las zonas transparentes.
Abajo: Resultado de la detección.

6.5.2. Profundidad del río.

A la hora de buscar una localización en la que realizar la detección y aplicar el algoritmo se deberá tener en consideración la profundidad del río. Podemos encontrarnos en dos situaciones.

- Profundidad alta. Este es el caso más favorable, debido a que el fondo no se vería, haciendo que la superficie del agua sea muy regular y facilitaría la detección.
- Profundidad baja. En este caso, se apreciaría el fondo del río haciendo que se captara todo lo que hubiese en él como piedras o diferentes elementos de forma que haría que la superficie captada por la cámara fuese irregular. Sin embargo, este no es el peor de los problemas de una profundidad baja. Cuando un objeto viaja sobre el agua en una zona de baja profundidad se proyecta la sombra del objeto sobre el fondo del río y es captada por la cámara (Figura 42). Si la sombra es leve no debería suponer un gran problema, sin embargo, si es muy pronunciada y con una luminancia muy diferente a la superficie podría llegar a detectarse como un objeto.

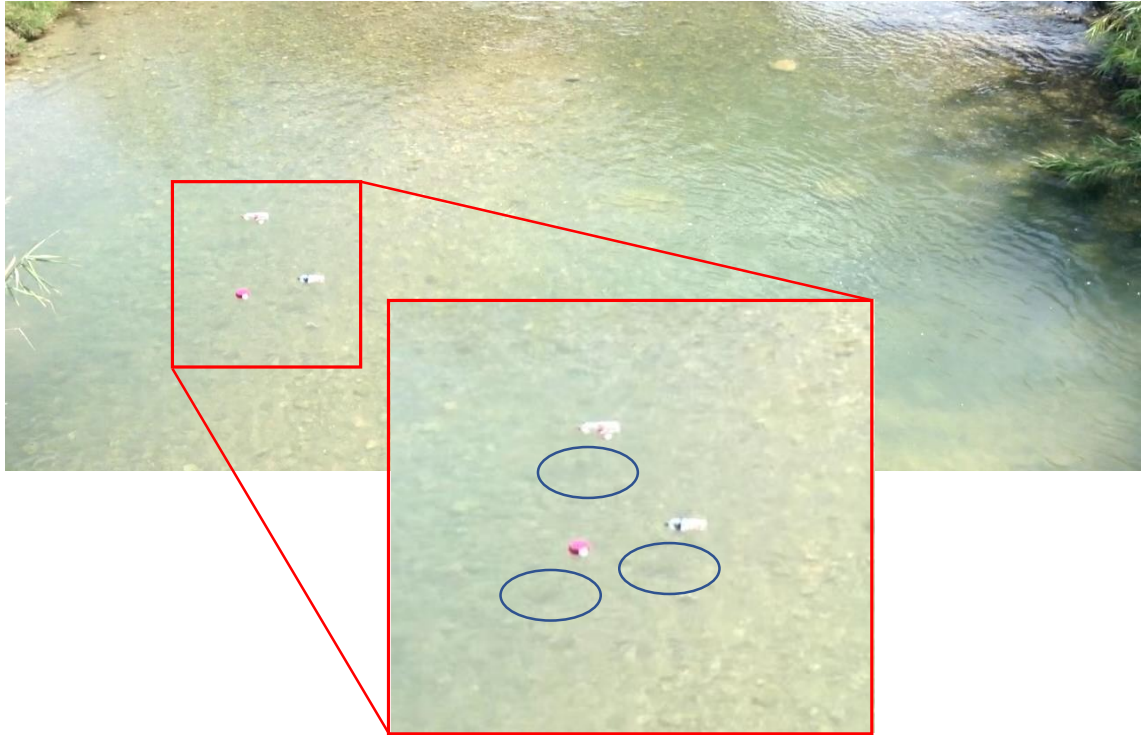


Figura 42. Detalle de las sombras proyectadas por los objetos sobre el fondo.

6.6. Aumentar la robustez del algoritmo.

Observando las grabaciones y los resultados nos podemos dar cuenta de que las formaciones de espuma o reflejos del agua, son aleatorios, pudiendo aparecer y desaparecer antes o después de la línea que hayamos escogido.

Teniendo esto en cuenta, para hacer más robusto el algoritmo se propone, en lugar de utilizar una sola línea de cada fotograma, utilizar dos líneas. De esta forma si aparece un objeto en la Pila de la línea 1 y también aparece en la Pila de la línea 2, aseguraremos que se trata de un objeto y no cualquier formación de espuma o reflejo del agua cuya formación varíe aleatoriamente.



Figura 43. Fotograma con las líneas escogidas indicadas.

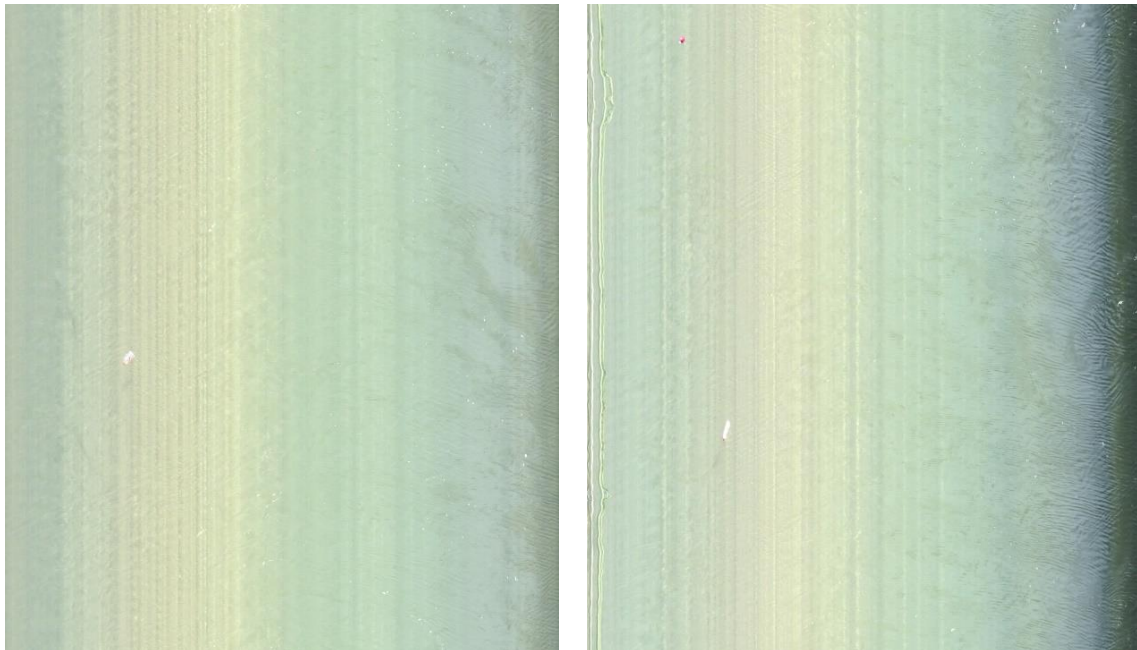


Figura 44. Comparativa de las pilas de cada línea.

Además, utilizar dos líneas nos va a permitir calcular características del río como por ejemplo la velocidad de la corriente. Si conocemos la distancia (en píxeles) entre las dos líneas elegidas y además sabemos que la duración en segundos de una línea (1/30 s), cuando aparezca un objeto simplemente tendremos que dividir la diferencia de altura que hay de línea a línea entre la diferencia de altura en la pila del punto en el que aparece (o termina) un objeto detectado.

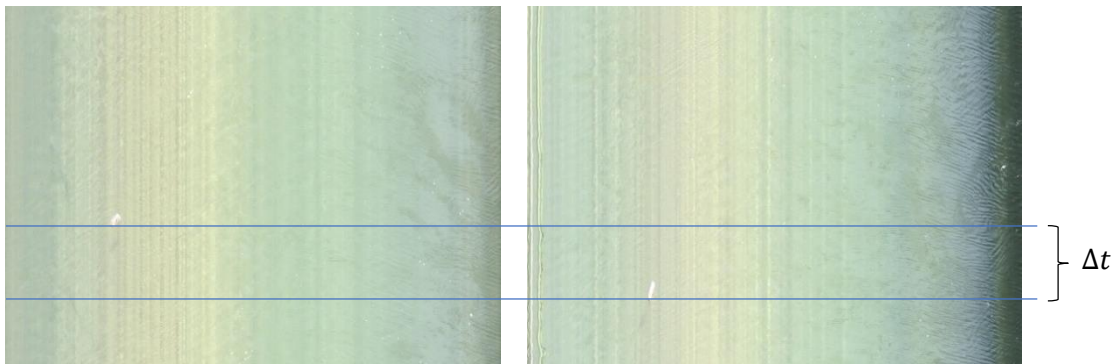
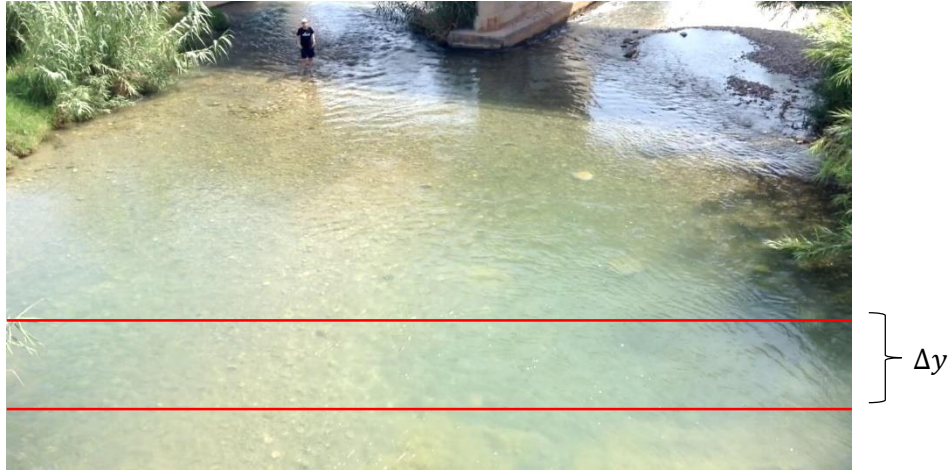


Figura 45. Incrementos de posición y tiempo.

De la Figura 44 podemos deducir la expresión de la velocidad de la corriente en píxeles/segundos.

$$v = \frac{\Delta y}{\Delta t}$$

Capítulo 7. Conclusiones y propuestas de trabajo futuro.

7.1. Conclusiones.

Los objetivos marcados para este proyecto eran los siguientes:

- Conseguir un algoritmo innovador para la detección de objetos flotantes en ríos.
- Búsqueda de localizaciones y grabación de ensayos para la puesta a prueba del algoritmo.

Con el algoritmo realizado se ha conseguido modelar estadísticamente los píxeles de una de las líneas de los vídeos grabados, logrando de este modo detectar cuando un objeto ha sido captado por la cámara, utilizando únicamente una única línea del fotograma, y consiguiéndolo con un coste computacional mucho más bajo que los algoritmos de detección de movimiento convencionales.

Además, se ha conseguido estudiar y encontrar una colocación de la infraestructura de grabación para una toma de imágenes favorable a su posterior análisis.

Otra conclusión importante a destacar es el conocimiento de las limitaciones del algoritmo y cuáles pueden ser soluciones a las mismas. Si un objeto es transparente y tenemos la mala suerte de que justo al pasar por la línea no refleja la luz, este objeto tendrá el color del fondo por tanto no será detectado, o no al menos totalmente. Por otra parte, la aparición espontánea de irregularidades como formaciones grandes de espuma o reflejos en el agua algunas veces serán eventos que no podremos predecir y puede que sean detectados como objetos. Aun así, se pueden prevenir estos sucesos aumentando el tiempo de aprendizaje del algoritmo para que las estadísticas recopiladas tengan una base más robusta

Así pues, se ha logrado desarrollar un algoritmo basado en la estadística y la probabilidad, con el que se ha conseguido modelar estadísticamente los píxeles de una grabación de la superficie de un río. También se ha reducido el coste de procesado de los algoritmos de detección de movimiento, favoreciendo la posible implementación del algoritmo en un dispositivo autónomo.

7.2. Trabajo futuro.

Hay varios aspectos en los que se podría trabajar para la mejora o extensión del algoritmo, con el objetivo de hacerlo aplicable a una situación real.

- En primer lugar y el paso más obvio sería la implementación del algoritmo en un dispositivo autónomo, como pudiera ser una Raspberry Pi, de forma que se pudiese instalar en un puente o cualquier otra infraestructura habilitada, y que estuviese conectada a un servidor en el que se almacenaran los datos de los objetos detectados. Habría que estudiar cuál sería la calidad mínima que necesitaría la cámara de la Raspberry Pi para que detectara objetos lo suficientemente pequeños. Además, se debería realizar un estudio de cuál sería el impacto visual que tendría la instalación de dicho dispositivo.
- En cuanto a la mejora del algoritmo, se podría aumentar el número de cámaras utilizadas y por tanto la cantidad de puntos de vista, de forma que tendríamos una mejor visión y la detección mejoraría notablemente.
- Por último, para la colocación de los dispositivos, se debería realizar un estudio de las mejores localizaciones a lo largo del río en las que las condiciones de grabación y detección fueran las más favorables.

Capítulo 8. Bibliografía.

[1] LINUX. UBUNTU. <<https://www.ubuntu.com/>>.

[2] ECLIPSE. <<https://eclipse.org/>>.

[3] LIBRERÍA Y DOCUMENTACIÓN CIMG. <<http://cimg.eu/reference/>>.

[4] LIBRERÍA Y DOCUMENTACIÓN OPENCV. <<http://docs.opencv.org/>>.

[5] DOCUMENTACIÓN MATLAB. <<https://es.mathworks.com/help/matlab/>>.

[6] FUNCIÓN ERROR Milton Abramowitz and Irene A. Stegun, eds. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York: Dover, 1972. (Capítulo 7)

[7] Parzen, E. (1962) “On Estimation of a Probability Density Function and Mode”. The Annals of Mathematical Statistics.

[8] <https://www.cs.utah.edu/~suyash/Dissertation_html/node11.html> [Consulta: 1 de Diciembre de 2017]

[9] Bayes, Thomas (1763). “An essay towards solving a Problem in the Doctrine of Chances”. Philosophical Transactions of the Royal Society of London.