



Universidad Autónoma del Carmen



Facultad de Ingeniería

**Desarrollo del Sistema de Detección de los Objetos
Flotantes sobre Superficie Marítima usando Visión
Artificial**

Tesis de Maestría

Presenta:

Ing. David Emmanuel Kantún Marín

Ciudad del Carmen, Campeche, México. 2018



Universidad Autónoma del Carmen



Facultad de Ingeniería

**Desarrollo del Sistema de Detección de los Objetos
Flotantes sobre Superficie Marítima usando Visión
Artificial**

Tesis de Maestría

Presenta:

Ing. David Emmanuel Kantún Marín

**Director de tesis:
Dr. Victor Golikov**

**Co-director de tesis:
Dr. Hussain Alazki**

Ciudad del Carmen, Campeche, México. 2018

*Dedicado a mis padres y a mis hermanos, que siempre estuvieron
apoyándome en mis decisiones desde el primer día*

*Sin olvidar a mi director y codirector de tesis, quienes siempre
me aconsejaron para mejorar en mi trabajo*

Índice general

Lista de acrónimos	7
Índice de figuras	8
Índice de tablas	12
Resumen	13
Abstract	14
1. Introducción	15
1.1. Historia	15
1.2. Planteamiento y justificación	16
1.3. Objetivo de la tesis	17
1.3.1. Objetivo general	17
1.3.2. Objetivos específicos	17
1.4. Metodología	17
2. Estado del arte	20
2.1. Evolución de las cámaras	20
2.2. Introducción a la visión artificial	21
2.3. La visión artificial en la robótica	22
2.4. Definición y representación de imágenes digitales	23
2.5. Filtrado	25
2.5.1. Filtros secuenciales	26
2.5.2. Filtros espaciales	27
2.5.3. Histograma	28
2.6. Color	29
2.6.1. Espacio RGB	29

2.6.2. Espacio HSI	30
2.7. Sensores utilizados en la formación de una imagen digital	31
2.8. Conversión analógico-digital	32
2.9. Muestreo de vídeo	32
2.10. Breve historia del vídeo digital	32
2.11. Proceso de imágenes y métodos	33
2.12. Ruido	34
2.13. Etapas básicas de una aplicación en visión artificial	34
2.14. Acción y/o decisión	35
2.15. Justificación	35
2.16. Objetivo de la tesis	38
2.16.1. Objetivos generales	38
2.16.2. Objetivos específicos	38
2.17. Metodología	38
2.18. Conclusiones	40
3. Desarrollo de algoritmos de detección de objetos en la superficie del mar	41
3.1. Distribución normal o gaussiana	41
3.2. Pruebas de hipótesis estadísticas	44
3.3. Planteamiento del problema de detección	46
3.4. Criterio de Neyman-Pearson y razón de verosimilitud	47
3.5. Desarrollo de algoritmos de detección de la señal en presencia de ruido gaussiano blanco	48
3.6. Desarrollo de algoritmos de detección de la señal en presencia de ruido coloreado	50
3.7. Razón de verosimilitud generalizada	51
3.8. Algoritmos de detección de la señal con potencia del fondo dependiente de la señal	53
3.9. Conclusiones	56
4. Pruebas de desempeño	58
4.1. Simulación de una secuencia de imágenes con ruido blanco	58
4.2. Simulación de una secuencia de imágenes con ruido correlacionado	66
4.3. Conclusiones	70
5. Pruebas experimentales	71
5.1. Muestras seleccionadas	71

5.2. Análisis de pruebas estadísticas de los videos en tercera dimensión	74
5.3. Análisis de pruebas estadísticas de los videos en primera y segunda dimensión	80
5.4. Conclusiones	81
6. Conclusiones finales	82
6.1. Trabajo a futuro	83
Referencias	84
Anexos	88

Lista de acrónimos

GLRT General Likelihood Ratio Test.

LRT Likelihood Ratio Test.

MMSD Modified Matched Subspace Detector.

MSD Matched Subspace Detector.

PDF Probability Density Function.

SBR Signal to Background Ratio.

Índice de figuras

1.1. Esquema del proceso de detección del objeto	18
2.1. Diagrama a bloques de las etapas de un sistema de visión artificial	23
2.2. Matriz de pixeles	24
2.3. Histograma de una imagen digital. En este ejemplo se muestra la imagen de prueba Lenna con su histograma en escala de grises	28
2.4. Cubo de espacio de color RGB	30
2.5. Representación del espacio de color HSI	31
2.6. Esquema básico de la sustracción del fondo	36
2.7. Ejemplos de sustracción del fondo	36
2.8. Dos barcos a una distancia de 2 km y sus picos de voltaje. Se observan los pulsos de los barcos y un pico falso que representa el reflejo del mar.	37
2.9. Dos barcos a una distancia de 4 km y sus picos de voltaje.	37
2.10. Esquema del proceso de detección del objeto	39
3.1. Curva normal	42
3.2. Ejemplo de distribución normal bivariada	43
3.3. Muestra gráfica de las pruebas de hipótesis estadísticas	44
3.4. Diagrama a bloques del detector acoplado	49
3.5. Diagrama a bloques del detector acoplado cuando se tiene ruido correlacionado	51
3.6. Esquema de fondo con objeto multipixel	55
4.1. Señal útil modelada con el comando <i>ones</i> de MATLAB.	61
4.2. Señal útil modelada con el comando <i>rand</i> de MATLAB.	61
4.3. Señal útil modelada con el comando <i>ones</i> de MATLAB.	61
4.4. Señal útil modelada con el comando <i>rand</i> de MATLAB.	61
4.5. Señal útil modelada con el comando <i>ones</i> de MATLAB más fondo decorrelacionado y ruido del canal.	62

4.6. Señal útil modelada con el comando <i>rand</i> de MATLAB más fondo decorrelacionado y ruido del canal.	62
4.7. Señal útil modelada con el comando <i>ones</i> de MATLAB más fondo decorrelacionado y ruido del canal, vista desde el plano en tres dimensiones.	62
4.8. Señal útil modelada con el comando <i>rand</i> de MATLAB más fondo decorrelacionado y ruido del canal, vista desde el plano en tres dimensiones.	62
4.9. Espectro de fondo decorrelacionado	63
4.10. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01, p = 4$	63
4.11. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01, p = 8$	63
4.12. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. Parámetros $P_{FA} = 0.01, p = 4$	64
4.13. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01, p = 8$	64
4.14. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01, p = 4$	65
4.15. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01, p = 8$	65
4.16. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01, p = 4$	65
4.17. P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01, p = 8$	65
4.18. Señal útil modelada con el comando <i>ones</i> de MATLAB más fondo correlacionado y ruido del canal.	67
4.19. Señal útil modelada con el comando <i>rand</i> de MATLAB más fondo correlacionado y ruido del canal.	67
4.20. Señal útil modelada con el comando <i>ones</i> de MATLAB más fondo correlacionado y ruido del canal, visto en tercera dimensión.	67
4.21. Señal útil modelada con el comando <i>rand</i> de MATLAB más fondo correlacionado y ruido del canal, visto en tercera dimensión.	67
4.22. Espectro de fondo correlacionado	68
4.23. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01, p = 4$	68
4.24. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01, p = 8$	68

4.25. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 4$	69
4.26. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 8$	69
4.27. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 4$	69
4.28. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 8$	69
4.29. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 4$	70
4.30. P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 8$	70
5.1. Nadador sosteniendo dos bolas, una blanca y una negra	72
5.2. Nadador con flotador cruzando el mar	72
5.3. Objetos flotando en el mar	72
5.4. Representación gráfica del análisis de video	73
5.5. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.6	75
5.6. Nadador con dos bolas a su alrededor con una ventana de análisis de 5×5 píxeles	75
5.7. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.8	75
5.8. Nadador con dos bolas a su alrededor con una ventana de análisis de 10×10 píxeles	75
5.9. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.10	75
5.10. Nadador con dos bolas a su alrededor con una ventana de análisis de 20×20 píxeles	75
5.11. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.12	76
5.12. Nadador cruzando el mar con una ventana de análisis de 5×5 píxeles	76
5.13. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.14	76
5.14. Nadador cruzando el mar con una ventana de análisis de 10×10 píxeles	76
5.15. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.16	77

5.16. Nadador cruzando el mar con una ventana de análisis de 20×20 pixeles . . .	77
5.17. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.18	77
5.18. Barco en el mar con una ventana de análisis de 5×5 pixeles	77
5.19. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.20	78
5.20. Barco en el mar con una ventana de análisis de 10×10 pixeles	78
5.21. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.22	78
5.22. Barco en el mar con una ventana de análisis de 20×20 pixeles	78
5.23. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.24	79
5.24. Objetos en el mar con una ventana de análisis de 5×5 pixeles	79
5.25. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.26	79
5.26. Objetos en el mar con una ventana de análisis de 7×7 pixeles	79
5.27. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.28	79
5.28. Objetos en el mar con una ventana de análisis de 10×10 pixeles	79
5.29. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.24	80
5.30. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.26	80
5.31. Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.28	81

Índice de tablas

3.1. Situaciones posibles al probar una hipótesis estadística	45
-------------------------------------------------------------------------	----

Resumen

En este trabajo se presenta el desarrollo y comparación de dos algoritmos de detección de objetos flotantes en la superficie marítima. Ambos algoritmos utilizan parámetros estadísticos, los cuales presentan un mejor desempeño ante condiciones de fondo fluctuante que los algoritmos clásicos de videovigilancia. Asimismo se debe señalar que uno de los algoritmos es una mejora del anterior, el cual considera parámetros como la energía de los píxeles del fondo. Ambos algoritmos son puestos a prueba en condiciones similares tanto simuladas como en video real para mostrar su desempeño y la aportación del nuevo algoritmo frente al clásico.

Abstract

This work presents the development and comparison of two algorithms used in detection of floating objects on sea surface. Both algorithms use statistical parameters, which present a better performance in fluctuating background conditions than classical ones used in video surveillance. Also, it should be noted that one of the algorithms is a modification of an older one, which considers parameters like background pixel energy. Both ones are tested in similar conditions like simulated environments as well as real video conditions in order to show the improvement and performance of the new algorithm respect to the older one.

Capítulo 1

Introducción

1.1. Historia

El término “teledetección”, o “remote sensing” en inglés, hace referencia a la obtención de la información sobre un objeto, área o fenómeno sin un contacto directo con el mismo. Este término, acuñado a principios de los años 60, si bien se designó en un principio a la fotografía clásica como principal sensor, en la actualidad también comprende otras maneras de obtener a distancia información del objeto deseado, ya sea a través de cámaras de vídeo o infrarrojas, o incluso a través de radares y sonares [9, 33].

En un sentido más amplio, la teledetección no engloba solo los procesos que permiten obtener una imagen desde el aire o desde el espacio, sino también su posterior tratamiento, en el contexto de una determinada aplicación [9, 23].

La teledetección ha sido una ciencia aplicada y muy dependiente del desarrollo tecnológico de la época en cuestión, siendo los antecedentes de la teledetección el desarrollo de las primeras cámaras y sensores fotográficos [23].

Los conflictos bélicos han sido un factor importante en el desarrollo de dichos sensores y cámaras fotográficas. Siendo vistos en un primer momento como un objeto de recreación, fue durante la Gran Guerra que se desarrolla la primera cámara aérea. Es ahí cuando la importancia estratégica de obtención de imágenes se evidencia. Pero es en la Segunda Guerra Mundial cuando el desarrollo científico de esta área se enfoca a mejorar la óptica de las cámaras de reconocimiento, así como a explorar otros rangos del espectro electromagnético no visible

como el infrarrojo [23, 19].

Sin embargo, en la década de 1960 fue cuando, unido al desarrollo de las computadoras, las plataformas satelitales fueron reemplazando a las aéreas, y las cámaras fotográficas sustituidas por sensores electrónicos que cumplían una función muy similar. Había nacido la era de la teledetección espacial [23].

Se hace notar que en la actualidad la palabra “teledetección” está más enfocada a la teledetección espacial, a través de imágenes de la superficie terrestre tomadas por sensores montados en un satélite. Sin embargo, no es la única forma de teledetección ya que, como se mencionó anteriormente, la teledetección es la detección de objetos a distancia a través de sensores.

Todas esas innovaciones, en un principio reservadas a usos militares, se encuentran disponibles para usos civiles. Otras áreas de las ciencias se han beneficiado del desarrollo de la teledetección: Geografía, Geología, Ingeniería Civil, Robótica, entre otros.

1.2. Planteamiento y justificación

Existen muchos tipos de sistemas de detección de objetos en movimiento. El principal problema con estos sistemas de detección es que funcionan bien en eventos donde el objeto de interés es el único que se encuentra en movimiento sobre un fondo fijo. Para efectos prácticos identificaremos como la señal útil el objeto de interés, y el ruido el fondo. En presencia de fondos fluctuantes, es decir, en fondos en movimiento, muchos de los algoritmos de detección que funcionaban sin ningún problema en un escenario ideal comienzan a fallar. El reto es encontrar algoritmos de detección de eventos remotos que permitan minimizar los efectos del ruido del fondo fluctuante, así como del ruido del canal.

Es en este tipo de casos en que nuestro sistema de detección debe detectar la presencia del objeto en cuanto aparezca, aún si el ojo humano no puede percibirlo por las condiciones del ambiente. La información del objeto de interés dependerá de la aplicación: detección de barcos, derrames de crudo, etc.

En la literatura se encuentran diferentes soluciones para la reducción de los efectos ocasionados por el mismo fondo fluctuante y demás ruidos. Los algoritmos investigados requieren información exacta sobre forma de las señales reflejadas desde las superficies de los

objetos de interés.

El fondo fluctuante que estudiaremos a lo largo de este trabajo es la reflexión de la superficie marina con olas. Existen otros tipos de fondos fluctuantes, como el movimiento de las hojas de los árboles, de reflexiones de luz, e inclusive la presencia de la lluvia puede provocar fluctuaciones en el ambiente.

1.3. Objetivo de la tesis

1.3.1. Objetivo general

El objetivo del presente trabajo es realizar una comparación de diferentes algoritmos para sistemas de detección de objetos en presencia de un fondo fluctuante, utilizando diferentes escenarios de vídeos, y ajustar sus parámetros con el fin de cumplir con el criterio de Neyman-Pearson.

1.3.2. Objetivos específicos

- Simular vía software un escenario con un objeto y un fondo, variar sus parámetros espectrales y evaluar los algoritmos elegidos.
- Analizar los parámetros espectrales de la superficie del mar y de los objetos deseados, así como del ruido del canal de propagación con datos experimentales.
- Prueba de los algoritmos propuestos con diferentes escenarios bajo diferentes condiciones.
- Se mostrarán los resultados obtenidos y se dará un veredicto de qué algoritmo o algoritmos funcionan mejor para cada escenario.

1.4. Metodología

Las herramientas a utilizar serán las siguientes:

- Vídeos de diferentes escenarios de mar y con diversas cámaras
- Computadora con MATLAB

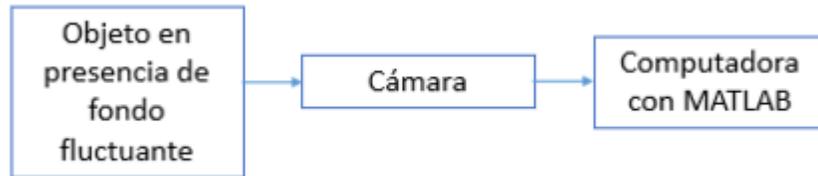


Figura 1.1: Esquema del proceso de detección del objeto

Como se podrá notar, aunque las herramientas son pocas, la diversidad de los escenarios reside en la cantidad de vídeos que se puedan tomar con diferentes condiciones en el ambiente, tales como la reflexión e intensidad de la luz o la fluctuación de las olas del mar.

Además de contar con vídeos que ya se han estudiado y analizado en tesis previas que han abordado el tema, se planea tomar nuevos vídeos para comprobar los resultados obtenidos y para mejorar los parámetros de detección.

MATLAB es una herramienta de software matemático con un entorno de programación utilizado por millones de ingenieros y científicos del mundo. Su versatilidad y su propio lenguaje de programación M, optimizados para resolver problemas matemáticos, de ingeniería y científicos, permiten crear algoritmos complejos con pocas líneas. Además incluye herramientas que permiten crear interfaces gráficas para que el usuario final pueda interactuar con el propio algoritmo.

La razón para elegir MATLAB y su lenguaje de programación M es debido a que, a diferencia de C, los algoritmos matemáticos y estadísticos pueden ser implementados con facilidad de manera nativa. Su tratamiento de la información a través de matrices permitirá trabajar con la misma de una manera a la que estamos acostumbrados matemáticamente.

La principal desventaja de MATLAB es que no permite trabajar en línea, por lo que en este trabajo no se realizarán pruebas en vivo. Otras desventajas incluyen la cantidad de recursos en computadora que consume el entorno de programación, o su limitación de soporte a arquitecturas de procesamiento Intel o AMD de 32 y 64 bits dejando de lado computadoras educativas de bajo costo como la Raspberry Pi.

La razón para elegir algoritmos estadísticos basados en la razón de verosimilitud generalizada, o GLRT (del cual se hablará con más detalle en el capítulo 3.7) reside en que en la literatura y tesis previas, se ha demostrado que estos algoritmos son los más eficientes a la hora de detectar objetos en presencia de un fondo fluctuante [3, 16, 22, 34].

Capítulo 2

Estado del arte

Para mejorar la calidad de los detectores, es importante entender el proceso que existe detrás de la adquisición tanto de imágenes como de video en formato digital. En esta sección será presentado el estado del arte del problema de detección de objetos usando visión artificial, así como los diferentes procesos de detección y sus principales problemas.

2.1. Evolución de las cámaras

Uno de los instrumentos importantes para la detección ha sido partícipe de una activa evolución: la cámara fotográfica. Las primeras versiones de esta cámara fueron las llamadas “cámaras oscuras”. Esta fue la base para obtener las primeras imágenes llamadas “daguerrotipos” [7, 12].

Los problemas que presentaban estas cámaras era que, los agujeros pequeños de estas cámaras, necesarios para obtener imágenes detalladas, requerían tiempos de exposición de hasta horas. Una solución fue colocar lentes ópticas que sustituían al agujero. El problema de las lentes es que, a pesar de resolver el tiempo de exposición altamente prolongado, traían un nuevo problema: las aberraciones, inherentes a estos elementos, ya sea por distorsiones en la forma de los objetos (aberración geométrica), como en la de las diferentes longitudes de onda que componen la luz (aberración cromática). Grandes científicos han trabajado mejorando las lentes para tratar de minimizar estas aberraciones y mejorar el desempeño de las cámaras fotográficas [8, 12].

La Segunda Guerra Mundial propició avances importantes en las ciencias, en particular en los campos de la fotografía y la óptica. La investigación en los espectros de la luz no visible al ojo humano, como la luz infrarroja y la ultravioleta, permitieron crear sistemas que pudieran detectar presencia enemiga que en el espectro visible sería difícil o casi imposible de detectar [12, 23].

La Guerra Fría y la carrera espacial requerían de una mejor manera de obtener fotografías de objetos remotos. Es aquí cuando nacen detectores de imagen de estado sólido como los de carga acoplada (CCD) y los de semiconductor-metal-óxido (MOS), requeridas para obtener imágenes remotas en tiempo real. La masificación de equipos fotográficos que incluían estos sensores, y el descenso del costo de fabricación de los mismos, permitió que el consumidor promedio pudiera adoptar estas tecnologías en sus equipos de consumo diario como teléfonos celulares y computadoras [12].

2.2. Introducción a la visión artificial

La visión por computadora consiste en la construcción de descripciones sobre objetos físicos a partir de imágenes para su posterior procesamiento. La creación de estas descripciones es un requisito previo para poder reconocer un objeto, manipularlo y pensar sobre él [14].

La visión por computadora se enfrenta a los siguientes retos [14]:

- ¿Cuál es la información válida que se puede extraer de una escena a partir de una imagen?
- ¿Qué tipo de proceso debe realizarse para extraer dicha información?
- Una vez extraída, ¿qué tipo de proceso se debe realizar?
- ¿Cómo utilizamos el conocimiento almacenado anteriormente?
- ¿Cómo representamos nuestro conocimiento y modelos sobre el mundo?

Las aplicaciones son muy amplias. Algunos ejemplos son los siguientes:

1. Robótica.

2. Imágenes aéreas.
3. Astronomía.
4. Medicina.
5. Química.
6. Física.

La formación de la imagen tiene lugar cuando un sensor registra la radiación que ha interactuado con otros objetos físicos. La lista de sensores utilizados en la formación de imágenes prácticamente no tiene fin y de ellos se hablará un poco más adelante.

2.3. La visión artificial en la robótica

En un ser humano, un sentido es una fuente de información sobre el mundo que lo rodea, para que en base a dicha información el cerebro tome una decisión. Extrapolando esa misma idea a un robot o a cualquier otro sistema artificial, existen sensores que pueden dotar de “sentidos” al mismo. Como en realidad es caro e impráctico dotar a un robot con todos los sentidos con los que cuenta el humano, normalmente se inclina hacia su función primaria para elegir adecuadamente un sensor que le proporcione información sobre el ambiente. La información que suele requerirse son dimensiones físicas, por lo que la predilección del sentido a replicar suele ser la vista [31].

La visión artificial no se debe confundir con el procesamiento digital de imágenes. A pesar de que la primera se apoya del segundo, de ninguna forma son lo mismo. El proceso de obtener información a través de la visión artificial para la toma de una decisión puede esquematizarse como se muestra en la figura 2.1 [14, 31].



Figura 2.1: Diagrama a bloques de las etapas de un sistema de visión artificial

2.4. Definición y representación de imágenes digitales

Si se representa en el plano de escala de grises, una **imagen** es una función bidimensional de la intensidad de luz y

$$f : Z \times Z \rightarrow Z$$

cuyos valores se han obtenido muestreando la intensidad sobre una retícula rectangular.

Una imagen se puede denotar como $f(x, y)$, donde x e y son las coordenadas espaciales y el valor de f en cada punto (x, y) es proporcional a la intensidad de luz (nivel de gris) de ese punto [11].

Podemos decir que una imagen $f(x, y)$ está formada por dos componentes: una es la cantidad de luz incidente en la escena y la otra es la cantidad de luz reflejada por los objetos. Estas dos componentes se llaman: **iluminación**, que denotaremos por $i(x, y)$ y **reflectancia**, que denotaremos por $r(x, y)$. Entonces

$$f(x, y) = i(x, y)r(x, y)$$

con $0 < i(x, y) < \infty$ y $0 < r(x, y) < 1$. La iluminación $i(x, y)$ está determinada por las

características de la fuente que emite la luz y la reflectancia $r(x, y)$ por las características del objeto [5, 11].

Llamaremos nivel de gris, l , a la intensidad de luz de una imagen f . El rango de variación de l será:

$$L_{min} \leq l \leq L_{max}$$

Llamaremos al intervalo $[L_{min}, L_{max}]$ escala de grises y normalmente se desplaza al intervalo $[0, L]$, donde $l = 0$ se considera negro y $l = L$ se considera blanco. El resto de valores son variaciones de grises que varían de forma continua desde el negro hasta el blanco.

Una imagen digital es una imagen que ha sido discretizada en el espacio y en los valores de intensidad que puede tomar. Podemos considerar una imagen digital como una matriz en la que la fila y la columna representan un punto en la imagen y el valor del elemento de la matriz corresponde al nivel de gris de ese punto. Los elementos de la matriz se llaman puntos o píxeles [5].

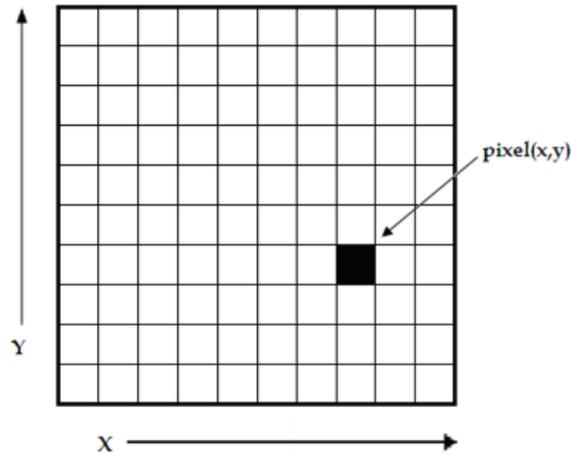


Figura 2.2: Matriz de píxeles

Supongamos que la imagen es muestreada espacialmente en las dos direcciones x e y de manera que tenemos una matriz $N \times N$, donde los elementos toman valores discretos.

$$f(x, y) = \begin{pmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N - 1) \\ \vdots & \vdots & \vdots & \vdots \\ f(N - 1, 0) & f(N - 1, 1) & \dots & f(N - 1, N - 1) \end{pmatrix} \quad (2.1)$$

Normalmente se toma valores de N y para el número de niveles de gris G potencias de 2.

$$\begin{aligned} N &= 2^n = 64, 128, 256, 512, 1024 \\ G &= 2^m = 2, \dots 256 \end{aligned} \tag{2.2}$$

Por tanto el número de bits para almacenar una imagen es:

$$b = N \times N \times m \tag{2.3}$$

Si la imagen es acromática, solo se presenta la luminancia, esto es, los niveles de grises. La función $f(x, y)$ retornará el nivel de gris del pixel mencionado. En caso de que la imagen sea en color, $f(x, y)$ devolverá un vector. Normalmente suele expresarse como una proyección del color sobre el sistema RGB (Red-Green-Blue) [11].

La imagen puede ser de tipo 3D, por ejemplo, en resonancia magnética, etc. Éstas se presentan como una pila de imágenes 2D. Se les añade otro índice, denominado k o z , que indica el orden de la rodaja de imagen 2D dentro de la pila [14].

La secuencia temporal de imágenes estáticas da lugar al vídeo. Existen diferentes valores de fotogramas por segundo utilizados para formar una secuencia de video. Gracias a la inercia visual del ojo humano, esta secuencia rápida de fotogramas da la sensación de continuidad de los objetos [5].

2.5. Filtrado

Existen diferentes tipos de ruidos que se producen por múltiples circunstancias: los sensores, ruidos eléctricos, perturbaciones en el medio de transmisión, efectos térmicos, campos electromagnéticos, etc. [5, 17]

La mayoría de las implementaciones de filtros se realizan en dos dominios: espacial y frecuencial.

Los métodos basados en el dominio espacial hacen referencia a la manipulación directa de la luminancia de los pixeles, mientras que los modelos basados en el dominio frecuencial modifican indirectamente la luminancia de cada pixel utilizando como factores de pondera-

ción los valores de los otros píxeles de la imagen o del entorno del punto y las relaciones numéricas entre ellos.

Las operaciones espaciales se pueden entender como la modificación directa del espectro de frecuencias de la imagen. Por supuesto, este espectro de frecuencias se definen como frecuencias espaciales para una imagen. El proceso consistiría en obtener el espectro de la imagen y del filtro a aplicar, y multiplicar ambos para realizar el filtrado.

$$G(u, v) = F(u, v) \cdot H(u, v) \quad (2.4)$$

siendo $H(u, v)$ la transformada de Fourier del filtro en cuestión, y $F(u, v)$ y $G(u, v)$ las transformadas de las imágenes original y filtrada respectivamente.

2.5.1. Filtros secuenciales

Se basan en la Transformada Directa de Fourier (DFT) y la Transformada Directa de Fourier Inversa (IDFT). Estas técnicas se han desarrollado como un caso particular de la Transformada de Fourier clásica o continua, ya que se debe recordar que en tecnologías informáticas, se trabaja con señales discretas [17].

- **Transformada directa de Fourier (DFT):** Mediante esta transformada se obtiene el espectro frecuencial de la imagen en cada eje, obteniendo a unas determinadas frecuencias el módulo y argumento de cada componente frecuencial de la imagen.
- **Transformada discreta de Fourier inversa (IDFT):** Mediante esta transformada se obtiene una imagen del formato de la imagen original, a partir del espectro frecuencial de esa misma imagen.
- **Transformada rápida de Fourier (FFT):** Es una herramienta muy potente para el análisis frecuencial de señales discretas. La FFT permite convertir una señal que está en el dominio del tiempo al dominio de las frecuencias. Es un *ingenio* matemático que permite obtener de una señal digital, muestreada a lo largo del tiempo, los armónicos que la componen, en amplitud y desfase.

Existen otros algoritmos matemáticos que también son muy usados, como la transformada directa del coseno utilizada para la compresión de imágenes JPEG (estáticas) y MPEG (vídeo) [20].

- **Filtros FIR y Filtros IIR:** El filtrado digital de señal mediante filtros FIR (filtros de respuesta impulsional finita) e IIR (filtros de respuesta impulsional infinita) consisten en aplicar la convolución de una matriz bidimensional, formada por los coeficientes del filtro, con la matriz de la imagen a filtrar. En realidad, los filtros IIR son poco utilizados debido a que son bastante inestables.

2.5.2. Filtros espaciales

Este tipo de filtros suelen consistir en recorrer toda la imagen, pixel a pixel, y realizar alguna operación aritmética con un número concreto de pixeles vecinos. A este conjunto de pixeles se le llama *ventana*, la cual puede ser cuadrada. Algunos de estos filtros son los siguientes [2, 17]:

- **Promediadores de área:** Son filtros pasa bajas que se utilizan fundamentalmente para reducir el ruido de alta frecuencia que se produce en una imagen.
- **Filtro de media o de suavizado:** Sustituye cada pixel por la media aritmética de los pixeles vecinos. Se toma una ventana de $N \times N$ puntos, donde el punto a sustituir es el central, se obtiene la media de la suma de todos los valores de los pixeles de la ventana y se sustituye el pixel en la nueva imagen por el valor obtenido. El efecto del filtro aumenta a medida que lo hace el tamaño de la ventana.
- **Filtro de mediana:** Consiste en obtener la mediana de todos los valores de los pixeles de la ventana. Es muy usado para eliminar el ruido impulsivo de una imagen.
- **Filtro de moda:** Consiste en calcular el valor más repetido dentro de todos los pixeles de una ventana (moda).
- **Normalización:** Consiste en aplicar un mapa de transformación de luminancias para extender o disminuir estas. Por ejemplo, si se quiere convertir una imagen RGB con 256 niveles por cada canal de color (R, G y B) a una imagen en escala de grises, el proceso más común consiste en hacer un promedio del valor de cada canal en un pixel, y repetir el mismo procedimiento en todos los pixeles de la imagen.
- **Filtros realzadores de contorno:** Son utilizados para hacer que los bordes sean más marcados.

- **Filtros de búsqueda de gradiente:** Este tipo de filtros buscan y acentúan aquellos puntos donde el gradiente es mayor. El gradiente es una función que da como resultado un vector cuya dirección indica la dirección de máxima diferencia entre puntos (pendiente) y cuyo módulo indica el valor de la pendiente en esa dirección.
- **Filtros detectores de bordes mediante gradientes con promediado:** Estos operadores obtienen el gradiente en una dirección y promedian en la perpendicular. Los métodos más usados de extracción de bordes son Kirsch, Prewitt y Sobel.

2.5.3. Histograma

Un histograma de nivel de gris de una imagen digital es una función que nos da la frecuencia de ocurrencia de cada nivel de canal en la imagen, Por ejemplo, si se dispone de un número n de niveles de gris, el valor de histograma para un particular nivel de gris es el número de píxeles en la imagen que tienen dicho nivel de gris [14].

En la figura 2.3 se puede apreciar el histograma de nivel de gris de una imagen digital, En este caso el número de niveles de gris es 256 (8 bits).

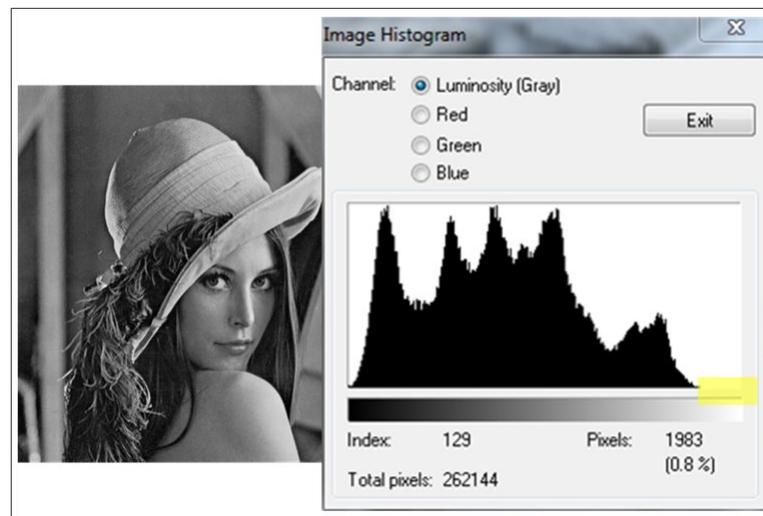


Figura 2.3: Histograma de una imagen digital. En este ejemplo se muestra la imagen de prueba Lenna con su histograma en escala de grises

El histograma genera información interesante que puede ser útil para el proceso de imágenes. Por ejemplo, se puede saber si la imagen es clara u oscura, si tiene contraste bajo o alto, e incluso ser decisiva a la hora de identificar piezas en un ambiente industrial.

2.6. Color

Un espacio de color es un método por el que se puede especificar, crear o visualizar cualquier color. Dependiendo del tipo de sensor se han creado diferentes espacios de colores que se utilizan para la adquisición, transmisión de señales, impresión o espacios que tratan de imitar la percepción humana.

2.6.1. Espacio RGB

Se basa en la combinación de tres señales de luminancia cromática distinta: rojo, verde y azul. La manera más sencilla es determinar la cantidad de rojo, verde y azul combinadas para conseguir un color en específico. En ese caso se realiza la suma de los componentes [10]

$$X = R + G + B$$

Aunque el espacio RGB es el más intuitivo de todos, presenta el inconveniente de que en sus tres valores mezcla la información del color (tono y saturación) y la intensidad [10].

La figura 2.4 muestra un cubo que representa este espacio de color.

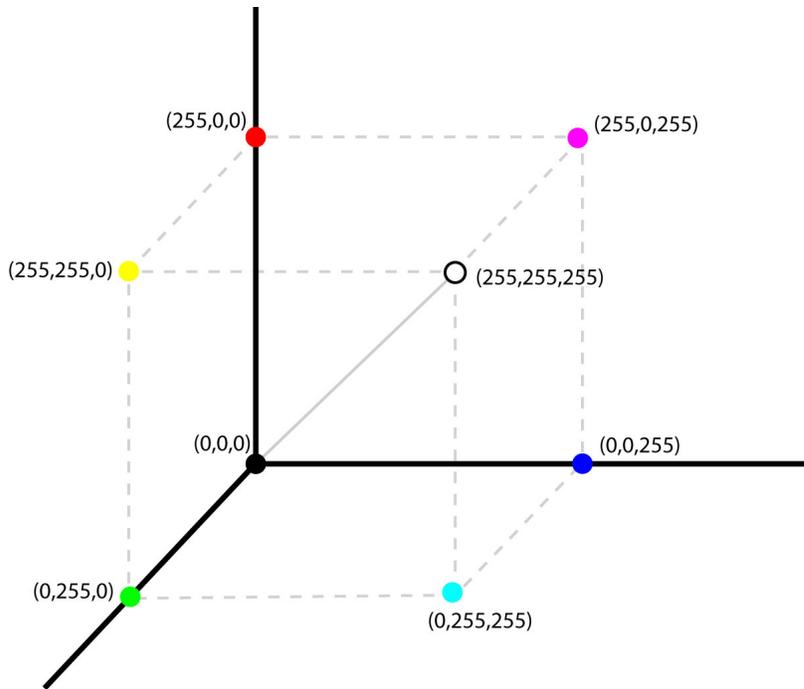


Figura 2.4: Cubo de espacio de color RGB

2.6.2. Espacio HSI

Se basa en el modo de percibir los colores que tenemos los seres humanos. Caracteriza el color en términos de tono o tinte (Hue), saturación o cromatismo (Saturation) y brillo (Intensity). Componentes que se muestran favorables de cara a realizar segmentaciones de la imagen en atención al tono o tinte de color [10].

Las ecuaciones que permiten transformar de RGB a HSI son las siguientes:

$$\begin{aligned}
 I &= \frac{R + G + B}{3} \\
 H &= \arctan \left(\frac{\sqrt{3}(G - B)}{(R - G) + (R - B)} \right) \\
 S &= 1 - \frac{\text{mín}(R, G, B)}{I}
 \end{aligned}
 \tag{2.5}$$

La figura 2.5 muestra un rombo que representa este espacio de color.

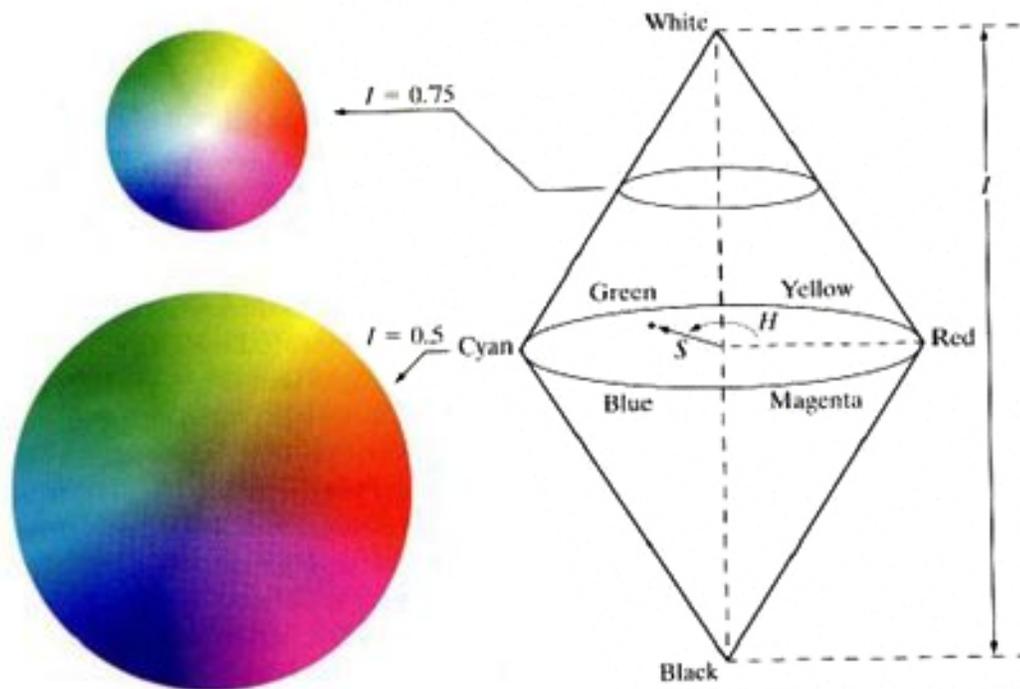


Figura 2.5: Representación del espacio de color HSI

2.7. Sensores utilizados en la formación de una imagen digital

En la actualidad existe una gran cantidad de métodos para la obtención de una imagen digital por computadora. La mayoría de los sensores son de tipo analógico y su señal de salida debe ser digitalizada antes de ser usada por la computadora. Algunos de los métodos empleados en la formación de imágenes son los siguientes [14]:

- Digitadores de imágenes (Scanners).
- Cámaras de televisión.
- Tubo disector de imagen.

- Transferencia de carga.
- CCD Acoplamiento de carga (Charge-coupled device).
- CID Inyección de carga (Charge-injection device).

2.8. Conversión analógico-digital

La función del convertidor analógico-digital es la de recibir como entrada un voltaje como el de una señal de vídeo y producir como salida una representación digital de ese voltaje. La calidad de un convertidor A/D se mide mediante la velocidad a la que realiza la conversión y la precisión de su salida digital [5, 14].

2.9. Muestreo de vídeo

Una muestra de vídeo puede ser representada como una imagen variante en el tiempo $f_c(x, y, t)$, la cual es una función de dos dimensiones espaciales x y y , usualmente observadas en una ventana espacial rectangular \mathcal{W} sobre un intervalo de tiempo \mathcal{T} . La región espacio-temporal $\mathcal{W} \times \mathcal{T}$ se denota como $\mathcal{W}_{\mathcal{T}}$. La ventana espacial es de dimensión $pw \times ph$, donde pw es el ancho de la imagen, mientras que ph es el alto. La imagen f_c puede ser muestreada en una, dos o tres dimensiones. Casi siempre se muestrea en la dimensión temporal, lo que produce una secuencia de imágenes [5].

Un ejemplo muy común de una imagen muestreada solo en la dimensión temporal es una película de vídeo.

2.10. Breve historia del vídeo digital

El primer intento de digitalización de la señal cromática de televisión se remonta a finales de los 50 con el trabajo de R. L. Carbrej. Sin embargo, en años posteriores el trabajo se centra más en la digitalización de señales monocromáticas. No es sino hasta 1971 que se retoma la cromatización [27].

Entre los principales motores estuvieron la conversión de estándares para la transmisión e intercambio de programas de televisión, principalmente entre Europa y América, debido a la diferencia entre estándares de transmisión de señales de televisión [27].

En 1986, Sony introdujo comercialmente el vídeo digital con el formato D1. Su siguiente formato, D2, al igual que el anterior, no utilizaba compresión de vídeo. Ambos formatos serían eventualmente reemplazados por otros formatos más económicos que sí aplicaban dicha compresión como el muy popular DV y MiniDV. Esto conlleva un mejor aprovechamiento del almacenamiento para crear video con mayor cantidad de pixeles por cuadro [28].

A pesar de ser de baja calidad en un principio, la adopción del vídeo digital provocó el despliegue de formatos que permitieran reducir el ancho de banda, y a la vez mejoraran la calidad de la imagen.

2.11. Proceso de imágenes y métodos

El término proceso de imágenes hace referencia al análisis y alteración de imágenes, con el fin de conseguir realzar una serie de características que no son fáciles de identificar en la imagen original y que, posteriormente, pueden ser usadas para lograr, nuestros objetivos.

Fundamentalmente existen tres formas de proceso:

- **Proceso óptico.** Este proceso implica el uso de elementos ópticos.
- **Proceso analógico.** Una señal de vídeo puede ser considerada como una onda eléctrica. Hace referencia a la posibilidad de alterar dicha señal para conseguir nuestros fines.
- **Proceso digital.** Es el desarrollado por computadoras digitales.

En el campo de la visión artificial, el digital es el proceso de interés. Cualquier otro proceso requiere de una etapa de digitalización [14].

2.12. Ruido

Todas las imágenes tienen una cierta cantidad de ruido, valores distorsionados, inherentes al sensor de la cámara o al medio de transmisión de la señal. El ruido puede clasificarse en cuatro tipos [6]:

- **Gaussiano:** Ruido estadístico que tiene una función de distribución de probabilidad igual a la distribución normal. Produce pequeñas variaciones en la imagen. Se considera que el valor final del pixel es la ideal más una pequeña ganancia del error, que se puede describir como una variable gaussiana. Sus fuentes principales son el ruido del sensor, perturbaciones en la transmisión y la digitalización de la imagen.
- **Impulsional:** El pixel toma un valor máximo, causado por una saturación del sensor, o mínimo si se ha perdido su señal. También puede encontrarse si se trabaja con objetos a altas temperaturas, debido a que las cámaras tienen una ganancia en el espectro infrarrojo del que carece el ojo humano.
- **Frecuencial:** La imagen es una suma entre la ideal y otra señal (interferencia) que se caracteriza por una senoide de frecuencia determinada.
- **Multiplicativo:** La imagen que se obtiene es producto de una multiplicación de dos señales.

2.13. Etapas básicas de una aplicación en visión artificial

Las etapas básicas de una aplicación en visión artificial se pueden resumir en los siguientes puntos [14]:

- **Construcción del sistema:** Su objetivo es realzar mediante técnicas fotográficas las características visuales de los objetos.
- **Preprocesado:** Su objetivo es mejorar la calidad de la imagen adquirida. Algunas de las operaciones que se pueden realizar para la mejora de las imágenes pueden ser la regularización de imágenes, mejora de contraste, realce de bordes, mejora de la relación señal-ruido (SNR), etc.
- **Segmentación:** Es la fase donde se particiona la imagen en áreas con significado.

- **Extracción de las características:** Estas pueden ser de tipo morfológico (área, perímetro, inercia, forma, etc.) o de características basadas en textura o en color.

2.14. Acción y/o decisión

Una vez que el robot recibe la información de la etapa de visión artificial, se procede a la toma de decisiones. Dependiendo del tipo de robot y su función, se realizará la acción para la que fue construido [31].

2.15. Justificación

El ambiente marítimo representa un escenario retador para la vigilancia de vídeo automática debido a:

1. Cambios de iluminación graduales y repentinos (por ejemplo, nubes)
2. Cambios de movimiento (por ejemplo, ruido)
3. Objetos de alta frecuencia en el fondo (por ejemplo, olas y gotas de agua)
4. Reflexiones

Una de las técnicas más comunes para la detección de objetos se conoce como sustracción del fondo. Es utilizada para detectar objetos como la diferencia entre un pixel actual y otro de referencia, llamado **modelo de fondo**. Las áreas donde la diferencia es significativa indican la presencia de un objeto.

$$|fondo - frame| > umbral \quad (2.6)$$

En la figura 2.6 se puede ver un esquema básico de esta técnica. La sustracción del fondo es una aproximación común a la detección de objetos en secuencias de vídeo tomadas con una cámara estática [36]. Como se puede ver, es una limitante muy importante debido a que esta técnica no funciona con otras condiciones en movimiento (cámara, fondo, etc.). En la figura 2.7 se puede ver el efecto de los objetos del fondo en la detección de esta técnica.

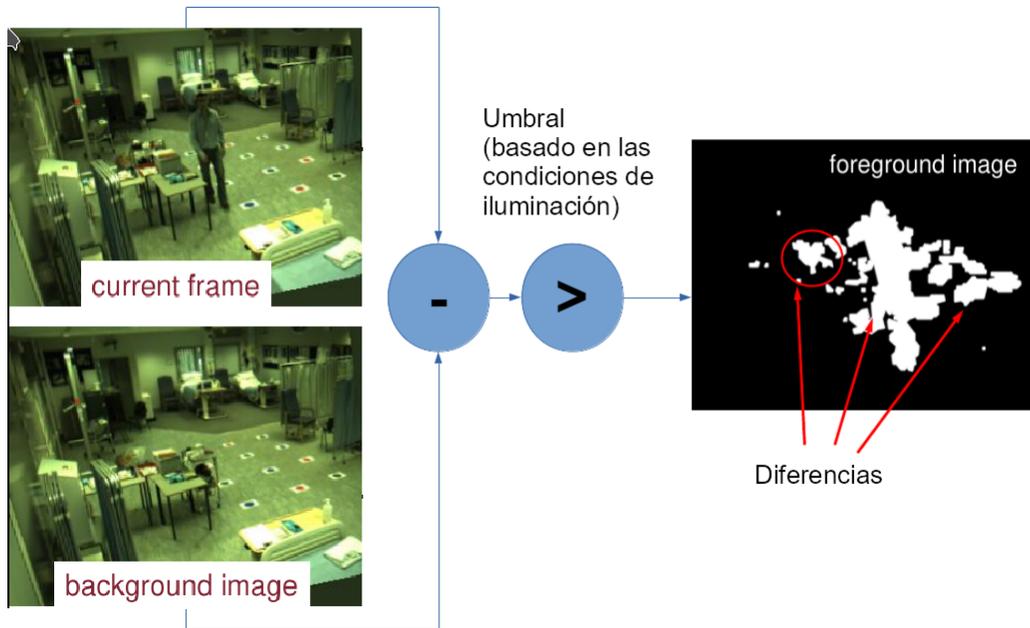


Figura 2.6: Esquema básico de la sustracción del fondo



Figura 2.7: Ejemplos de sustracción del fondo

Para este tipo de problemas se pueden utilizar técnicas basadas en algoritmos estadísticos utilizando el criterio de detección de Neyman-Pearson. En este criterio se requiere establecer a priori la probabilidad de falsa alarma fija y pequeña, posteriormente se maximiza la probabilidad de detección usando la razón de verosimilitud.

Si la razón de verosimilitud contiene parámetros desconocidos se usa la razón de verosimilitud en forma general donde en lugar de parámetros desconocidos se necesita sustituir sus

estimaciones.

Este criterio es común para muchas tareas de detección en sistemas de radares, sonares y vídeo (infrarrojos). Por lo cual será el criterio que se empleará para la solución del problema planteado.

En las figuras 2.8 y 2.9 se pueden observar algunos ejemplos de trabajos realizados con algoritmos estadísticos.

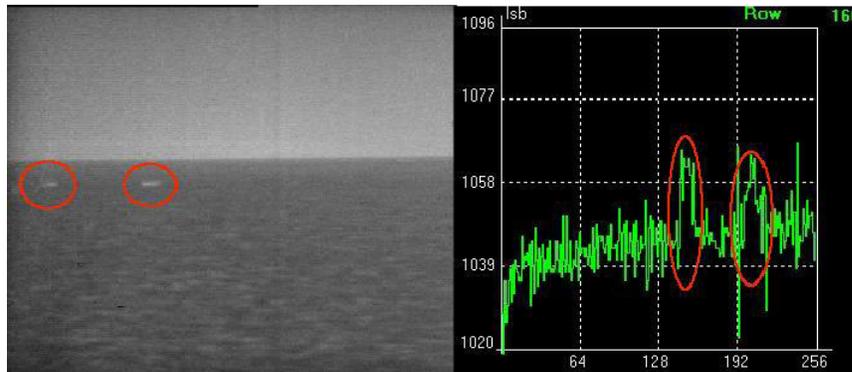


Figura 2.8: Dos barcos a una distancia de 2 km y sus picos de voltaje. Se observan los pulsos de los barcos y un pico falso que representa el reflejo del mar.

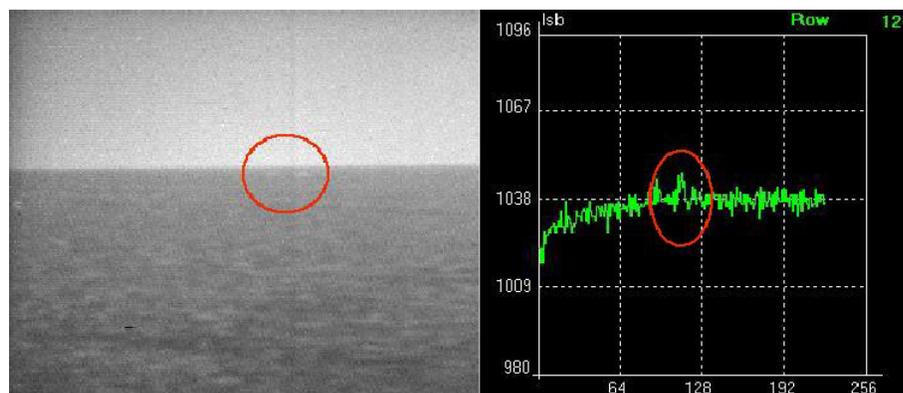


Figura 2.9: Dos barcos a una distancia de 4 km y sus picos de voltaje.

2.16. Objetivo de la tesis

2.16.1. Objetivos generales

El objetivo del presente trabajo es realizar una comparación de diferentes algoritmos para sistemas de detección de objetos en presencia de un fondo fluctuante, utilizando diferentes escenarios de vídeos, y ajustar sus parámetros con el fin de cumplir con el criterio de Neyman-Pearson.

2.16.2. Objetivos específicos

- Analizar los parámetros espectrales de la superficie del mar y de los objetos deseados, así como del ruido del canal de propagación con datos experimentales.
- Prueba de los algoritmos propuestos a través de simulaciones bajo diferentes condiciones.
- Prueba de los algoritmos propuestos con diferentes escenarios de video real bajo diferentes condiciones.
- Se mostrarán los resultados obtenidos y se dará un veredicto de qué algoritmo o algoritmos funcionan mejor para cada escenario.

2.17. Metodología

Las herramientas a utilizar serán las siguientes:

- Vídeos de diferentes escenarios de mar y con diversas cámaras
- Computadora con MATLAB

En la figura 2.10 se muestra un esquema del proceso del trabajo a realizar.

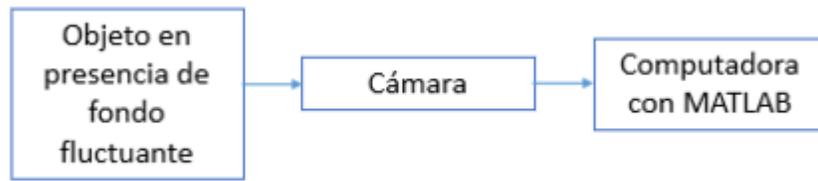


Figura 2.10: Esquema del proceso de detección del objeto

Como se podrá notar, aunque las herramientas son pocas, la diversidad de los escenarios reside en la cantidad de vídeos que se puedan tomar con diferentes condiciones en el ambiente, tales como la reflexión e intensidad de la luz o el movimiento de las olas del mar.

Además de contar con vídeos que ya se han estudiado y analizado en tesis previas que han abordado el tema, se planea tomar nuevos vídeos para comprobar los resultados obtenidos y para mejorar los parámetros de detección.

MATLAB es una herramienta de software matemático con un entorno de programación utilizado por millones de ingenieros y científicos del mundo. Su versatilidad y su propio lenguaje de programación M, optimizados para resolver problemas matemáticos, de ingeniería y científicos, permiten crear algoritmos complejos con pocas líneas. Además incluye herramientas que permiten crear interfaces gráficas para que el usuario final pueda interactuar con el propio algoritmo.

La razón para elegir MATLAB y su lenguaje de programación M es debido a que, a diferencia de C, los algoritmos matemáticos y estadísticos pueden ser implementados con facilidad de manera nativa. Su tratamiento de la información a través de matrices permitirá trabajar con la misma de una manera a la que estamos acostumbrados.

La principal desventaja de MATLAB es que no permite trabajar en línea, por lo que en este trabajo no se realizarán pruebas en vivo. Otras desventajas incluyen la cantidad de recursos en computadora que consume el entorno de programación, o su limitación de soporte a arquitecturas de procesamiento Intel o AMD de 32 y 64 bits dejando de lado computadoras educativas de bajo costo como la Raspberry Pi.

Estas deficiencias pueden ser atacadas en trabajos futuros portando los resultados obtenidos en este estudio a otros lenguajes de programación como C++ y la librería OpenCV (Open Source Computer Vision Library).

2.18. Conclusiones

En esta sección se presentaron breves historias de los componentes necesarios para la realización de este trabajo, así como la problemática que se abordará en la siguiente tesis y los problemas que se podrían presentar.

También se presentan algunos métodos comunes y sus principales desventajas, así como un método basado en algoritmos estadísticos que pretende funcionar mejor para ciertas condiciones de fondo y objeto fluctuantes. De la misma forma se presenta una visión preliminar de los pasos que se realizarán a lo largo de esta tesis.

Capítulo 3

Desarrollo de algoritmos de detección de objetos en la superficie del mar

3.1. Distribución normal o gaussiana

La distribución de probabilidad continua más importante en el área de la estadística es la **distribución normal**. Su gráfica denominada **curva normal**, es la curva con forma de campana que se muestra en la figura 3.1, la cual describe muchos fenómenos que ocurren en la naturaleza, la industria y la investigación de forma aproximada. Esta distribución también se conoce como **distribución gaussiana** en honor a Karl Friedrich Gauss (1777-1855), aunque fue Abraham DeMoivre quien en 1733 desarrolló la ecuación matemática de dicha curva [18].

Una variable aleatoria continua X que tiene una distribución en forma de campana, como se muestra en la figura 3.1 se denomina **variable aleatoria normal**. Su ecuación matemática para la distribución de probabilidad depende de los parámetros μ (media) y σ (desviación estándar, indica qué tan dispersos están los datos con respecto a la media) [18].

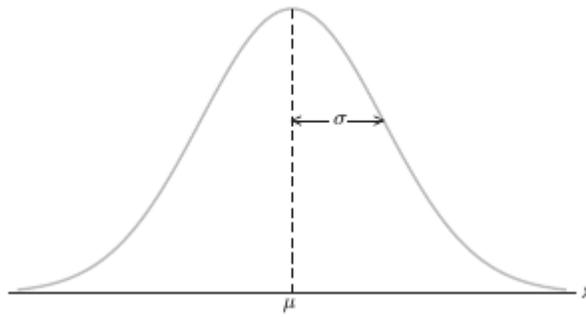


Figura 3.1: Curva normal

La distribución normal se emplea tan ampliamente que a menudo se recurre a la notación abreviada

$$X \sim N(\mu, \sigma) \quad (3.1)$$

para indicar que la variable aleatoria X se distribuye normalmente con media μ y desviación estándar σ .

Aunque algunos autores prefieren trabajar con la notación $X \sim N(\mu, \sigma^2)$, donde σ^2 denota la varianza de la variable aleatoria X , desde este punto y hasta el final de la tesis, se utilizará la notación (3.1) debido a que MATLAB utiliza dicha notación.

La función de densidad de la variable aleatoria normal X , con media μ ($-\infty < \mu < \infty$) y varianza $\sigma^2 > 0$, es

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2} \left[\frac{(x - \mu)}{\sigma} \right]^2, \quad -\infty < x < \infty \quad (3.2)$$

Las propiedades de la curva normal son [18]:

- La moda, que es el punto sobre el eje horizontal donde la curva tiene su punto máximo, ocurre en $x = \mu$.
- La curva es simétrica alrededor de un eje vertical a través de la media μ .
- La curva tiene sus puntos de inflexión en $x = \mu \pm \sigma$.
- La curva normal se aproxima al eje horizontal de manera asintótica, conforme se aleja de la media en cualquier dirección.

- El área total bajo la curva es igual a uno.

Una distribución normal multivariada es aquella en la que el vector \mathbf{x} sigue una distribución normal multivariada si su función de densidad conjunta es de la forma

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \det \mathbf{R}^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{R}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right] \quad (3.3)$$

donde \mathbf{R} es una matriz de dimensiones $n \times n$ no singular, simétrica y definida positiva, y

$$\mathbf{R} = E \left[(\mathbf{x} - \boldsymbol{\mu}) (\mathbf{x} - \boldsymbol{\mu})^T \right] \quad (3.4)$$

donde E es la esperanza matemática, la cual también puede definirse como valor esperado, esperanza, media poblacional o media [18].

Un caso particular de dicha distribución es la **distribución normal bivariada**. En este caso $n = 2$, y \mathbf{x} es un vector que contiene a las variables x_1 y x_2 con sus medias respectivas μ_1 y μ_2 . La matriz \mathbf{R} tiene la forma

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & \rho \\ \rho & \sigma_2^2 \end{bmatrix} \quad (3.5)$$

donde $\rho = \frac{\sigma_{12}}{\sigma_1 \sigma_2}$, y σ_{12} es la covarianza (grado de variación conjunta de dos variables aleatorias respecto a sus medias) entre X_1 y X_2 .

La figura 3.2 muestra gráficamente el comportamiento de la distribución normal bivariada.

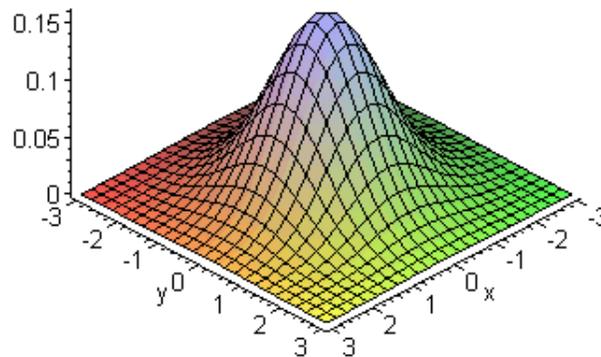


Figura 3.2: Ejemplo de distribución normal bivariada

3.2. Pruebas de hipótesis estadísticas

Una hipótesis estadística es una aseveración o conjetura respecto a una o más poblaciones [37]. El objetivo de la teoría de prueba de hipótesis es elegir entre dos posibilidades excluyentes relativas al valor de un parámetro poblacional, a partir de la información proporcionada por los datos muestrales. A una de ellas se le llamará “hipótesis nula”, y a la otra la llamaremos “hipótesis alternativa” [18].

Sea X_1, \dots, X_n una muestra aleatoria de una variable aleatoria X con función de distribución F_θ , donde $\theta \in \Theta$.

Dada una partición del espacio paramétrico $\Theta = \Theta_0 \cup \Theta_1$, deseamos decidir, en base a la muestra obtenida, si $\theta \in \Theta_0$ o si $\theta \in \Theta_1$. Es decir, queremos contrastar

$$H_0 : \theta \in \Theta_0 \text{ (hipótesis nula)}$$

$$H_1 : \theta \in \Theta_1 \text{ (hipótesis alternativa)}$$

La decisión para aceptar o rechazar la hipótesis nula se basa en una estadística de prueba calculada a partir de datos en una muestra aleatoria. Cuando se toma una decisión en base a una muestra aleatoria, esta decisión puede estar sujeta a error, en este caso pueden ocurrir dos tipos de errores, llamados “tipo I” y “tipo II” [18].

La figura 3.3 muestra una distribución normal donde se aprecian dos regiones, la zona de aceptación de H_0 , y la región crítica en la cual se rechaza H_0 .



Figura 3.3: Muestra gráfica de las pruebas de hipótesis estadísticas

El error de tipo I (α) se define como la probabilidad de rechazar H_0 dado que es verdadero. También es conocido como nivel o tamaño de significación de la prueba. Por otro lado, el error de tipo II (β) se define como la probabilidad de aceptar H_0 dado que es falso. Aunque a veces es mejor trabajar con la potencia de la prueba, donde

$$Potencia = 1 - \beta = P = \{\text{rechazar } H_0 | H_0 \text{ es falsa}\}$$

Debido a que rechazar H_0 es siempre una conclusión fuerte, en tanto que no rechazar H_0 una conclusión débil a menos que se sepa que β es pequeña, se prefiere construir hipótesis tales que el enunciado en torno al cual se desea una conclusión fuerte, esté en la hipótesis alternativa H_1 [18].

Al probar cualquier hipótesis estadística, hay cuatro situaciones posibles que determinan si nuestra decisión es correcta o errónea. La tabla 3.1 resume estas cuatro situaciones.

	H_0 es verdadera	H_0 es falsa
No rechazar H_0	Decisión correcta	Error tipo II
Rechazar H_0	Error tipo I	Decisión correcta

Tabla 3.1: Situaciones posibles al probar una hipótesis estadística

En las pruebas de hipótesis, un estadístico de prueba, es un valor estandarizado que se calcula a partir de los datos de la muestra durante una prueba de hipótesis [24]. El estadístico compara sus datos con lo que se espera según la hipótesis nula. Un valor crítico es un punto en la distribución de la prueba que se compara con el estadístico de prueba para determinar si puede rechazarse la hipótesis nula. La región crítica es aquella región en la que se cruzó el umbral del valor crítico [37].

Lo que se suele hacer en realidad es lo siguiente:

1. Acotar la máxima probabilidad de error de tipo I.
 - Se fija un nivel de significación $\alpha \in (0, 1)$. Lo típico para esta clase de aplicaciones es un valor de $\alpha = 0.01$.
 - Se define el tamaño de una prueba como la máxima probabilidad de error de tipo I. Es decir: $\max_{\theta \in \Theta_0} \mathbb{P}_\theta (R) = \max_{\theta \in \Theta_0} \beta_n (\theta)$
 - Se busca una región de rechazo R tal que $\max_{\theta \in \Theta_0} \mathbb{P}_\theta (R) \leq \alpha$
2. Minimizar la probabilidad de error de tipo II.

Las propiedades importantes de las pruebas de hipótesis son las siguientes:

1. Los errores tipo I y tipo II están relacionados. Una disminución en la probabilidad de cometer uno da como resultado un aumento en la probabilidad del otro.
2. El tamaño de la región crítica (la probabilidad de cometer un error tipo I) se puede reducir ajustando los valores críticos.
3. Un aumento en el tamaño de la muestra n reducirá α y β de forma simultánea.
4. Si H_0 es falsa, β es un máximo cuando el valor verdadero de un parámetro se aproxima al valor hipotético. A mayor distancia entre ambos valores, menor será β .

Para este tipo de trabajos se puede resumir que se tienen dos casos, una hipótesis H_0 conformada del ruido del fondo y el ruido del canal, y una hipótesis H_1 que consistirá del ruido del fondo, el ruido del canal y la señal útil.

3.3. Planteamiento del problema de detección

Debido a que las señales se describen estadísticamente, la decisión de las dos hipótesis es un ejercicio de teoría de decisión estadística. El análisis debe comenzar con una descripción estadística de la función de densidad de probabilidad (Probability Density Function (PDF) en inglés) que describe la medición que debe probarse en cada una de las dos hipótesis [1].

Si la muestra a ensayar se denota como x , se requieren las siguientes dos *pdfs*:

- $p_x(\mathbf{x}|H_0)$: pdf de x dado que el objetivo no está presente.
- $p_x(\mathbf{x}|H_1)$: pdf de x dado que el objetivo está presente.

Por lo tanto, parte del problema de detección es el desarrollo de modelos para estas dos *pdfs*. Más en general, la detección se basa en N muestras de datos x_n formando un vector columna \mathbf{x} :

$$\mathbf{x} = \begin{bmatrix} x_0 & \dots & x_{N-1} \end{bmatrix}^T \quad (3.6)$$

Las *pdfs* conjuntas N -dimensionales $p_x(\mathbf{x}|H_0)$ y $p_x(\mathbf{x}|H_1)$ se utilizan a continuación.

Suponiendo que las dos *pdfs* están modeladas con éxito, las probabilidades de interés se pueden definir como:

- Probabilidad de detección P_D : La probabilidad de que un objeto de interés es declarado (se elige H_1) cuando un objetivo está presente.
- Probabilidad de falsa alarma P_{FA} : La probabilidad de que un objeto de interés es declarado (se elige H_1) cuando un objetivo no está presente.
- Probabilidad de omisión P_M : la probabilidad de que un objeto de interés no es declarado (se elige H_0) cuando un objetivo está presente.

Si se toma en cuenta que $P_M = 1 - P_D$, entonces P_D y P_{FA} son suficientes para especificar las probabilidades de interés.

Supongamos que la señal discreta consiste de N valores y es accesible para la observación. Para llegar a una decisión, primero se debe formar una función T de los datos \mathbf{x} , y después tomar una decisión basada en su valor.

El problema central de la teoría de detección es determinar la función $T(\mathbf{x})$, cuyo valor es conocido como el **estadístico de prueba**.

3.4. Criterio de Neyman-Pearson y razón de verosimilitud

El uso más común de un criterio de Bayes se conoce como **criterio de Neyman-Pearson**. Este criterio describe de manera clara cuál procedimiento de prueba debemos aplicar cuando la forma de distribución de probabilidad de la muestra aleatoria depende solo del valor de un único parámetro desconocido y las hipótesis nula y alternativa son simples [26].

El criterio de Neyman-Pearson está motivado por el objetivo de obtener el mejor posible rendimiento de detección al tiempo que garantiza que la probabilidad de falsa alarma no exceda un cierto valor tolerable.

Este criterio consiste en rechazar las hipótesis $\Theta = \Theta_0$ tras observar x cuando la razón de las funciones de verosimilitud cumple lo siguiente:

$$H_1 : L(\mathbf{x}) = \frac{p(X|H_1)}{p(X|H_0)} > \lambda \quad (3.7)$$

A esta ecuación se le conoce como prueba de razón de verosimilitud (Likelihood Ratio

Test (LRT) en inglés). Esta ecuación establece que la relación de las dos funciones de densidad de probabilidad, cada una evaluada en particular observando los datos, es comparada con un umbral λ . Si la razón de verosimilitud supera el umbral elegido, se elige la hipótesis H_1 y se asume que el objetivo está presente. De lo contrario, se elige la hipótesis H_0 y se asume que el objetivo no está presente [1].

Bajo el criterio de Neyman-Pearson de optimización, la probabilidad de falsa alarma no puede superar el valor de diseño original P_{FA} .

Se considera el caso en el que el tamaño, forma y posición del posible blanco son desconocidos a priori. De acuerdo al criterio de Neyman-Pearson, la solución óptima al problema de pruebas de hipótesis es la LRT para parámetros conocidos, o la GLRT para parámetros desconocidos, de la cual se hablará en secciones posteriores.

3.5. Desarrollo de algoritmos de detección de la señal en presencia de ruido gaussiano blanco

Supóngase el caso en que el problema de detección está distinguido entre dos hipótesis cuando la señal $s[n]$ es conocida a priori,

$$\begin{aligned} H_0 : \mathbf{x}[n] &= \mathbf{w}[n] \\ H_1 : \mathbf{x}[n] &= \mathbf{s}[n] + \mathbf{w}[n] \end{aligned} \quad (3.8)$$

donde $\mathbf{w}[n] \sim N(0, \sigma)$ es el ruido gaussiano blanco, $\mathbf{s}[n]$ es la señal útil, y $n = 0, 1, \dots, N-1$.

Se puede explicar que la señal útil se suma con la señal de ruido para producir la hipótesis alternativa H_1 .

Regresando a (3.7), de donde

$$\begin{aligned} p(\mathbf{x}|H_1) &= \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp \left[-\frac{1}{2\sigma^2} \sum_{n=0}^{N-1} (\mathbf{x}[n] - \mathbf{s}[n])^2 \right] \\ p(\mathbf{x}|H_0) &= \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp \left[-\frac{1}{2\sigma^2} \sum_{n=0}^{N-1} (\mathbf{x}[n])^2 \right] \end{aligned} \quad (3.9)$$

se tiene

$$L(\mathbf{x}) = \exp \left[-\frac{1}{2\sigma^2} \left(\sum_{n=0}^{N-1} (\mathbf{x}[n] - \mathbf{s}[n])^2 - \sum_{n=0}^{N-1} (\mathbf{x}[n])^2 \right) \right] > \lambda \quad (3.10)$$

Tomando los logaritmos de ambos lados,

$$\frac{1}{\sigma^2} \left(\sum_{n=0}^{N-1} \mathbf{x}[n] \mathbf{s}[n] \right) - \frac{1}{\sigma^2} \sum_{n=0}^{N-1} \mathbf{s}[n]^2 > \ln \lambda \quad (3.11)$$

de donde se conoce $\mathbf{s}[n]$, la razón de verosimilitud simplificada es

$$T(\mathbf{x}) = \sum_{n=0}^{N-1} \mathbf{x}[n] \mathbf{s}[n] > \sigma^2 \ln \lambda + \frac{1}{2} \sum_{n=0}^{N-1} (\mathbf{s}[n])^2 = \lambda \quad (3.12)$$

$$T(\mathbf{x}) = \sum_{n=0}^{N-1} \mathbf{x}[n] \mathbf{s}[n] > \lambda'$$

En un detector acoplado como se muestra en la figura 3.4, la señal recibida de n valores discretos $\mathbf{x}[n]$ se multiplica por el patrón de la señal de interés de n valores discretos $\mathbf{s}[n]$. En la salida del multiplicador aparecen n valores discretos del producto del ruido por la señal para H_0 , o una suma de n valores discretos del producto del ruido por el patrón de la señal y n valores discretos de los cuadrados de los valores de la señal de interés para H_1 . Después, estos n valores discretos se acumulan y pasan a un sistema que los comparará con el umbral establecido [1].

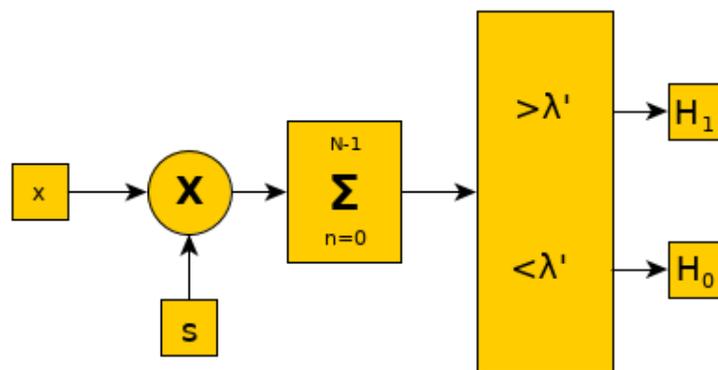


Figura 3.4: Diagrama a bloques del detector acoplado

Este tipo de filtro es muy útil cuando la señal de ruido es de tipo decorrelacionado (blanco) gaussiano aditivo y la señal s es conocida a priori.

3.6. Desarrollo de algoritmos de detección de la señal en presencia de ruido coloreado

Para el caso de ruido coloreado, el primer paso es convertir este tipo de ruido a uno de tipo decorrelacionado. Se puede usar un filtro que multiplica la señal recibida a una matriz $C^{-\frac{1}{2}}$, donde C es la matriz de covarianza del ruido recibido. De forma vectorial se puede representar de la siguiente forma:

$$C = \mathbf{w}\mathbf{w}^T = \mathbf{Q}\mathbf{Q}^T \quad (3.13)$$

Ya que \mathbf{Q} es la raíz cuadrada de C , $C^{-\frac{1}{2}} = \mathbf{Q}$, y el producto $C^{-\frac{1}{2}}\mathbf{w} = \mathbf{w}_1$ que da el ruido decorrelacionado.

Cuando el ruido se modela como una señal gaussiana blanca estacionaria, cuando C tiene la forma especial de una matriz de covarianza de tipo Toeplitz.

$$[C] = \text{cov}(w[m], w[n]) = E(w[m]w[n]) = r_{ww}[m-n] \quad (3.14)$$

$$p(\mathbf{x}|H_0) = \frac{1}{(2\pi)^{\frac{N}{2}} \det^{\frac{1}{2}}(C)} \exp -\frac{1}{2}\mathbf{x}^T C^{-1}\mathbf{x} \quad (3.15)$$

$$p(\mathbf{x}|H_1) = \frac{1}{(2\pi)^{\frac{N}{2}} \det^{\frac{1}{2}}(C)} \exp -\frac{1}{2}(\mathbf{x} - \mathbf{s})^T C^{-1}(\mathbf{x} - \mathbf{s}) \quad (3.16)$$

La razón de verosimilitud se compara con el umbral λ

$$L(\mathbf{x}) = \ln \frac{p(\mathbf{x}|H_1)}{p(\mathbf{x}|H_0)} > \ln \lambda \quad (3.17)$$

Al simplificar dicha razón de verosimilitud, se puede escribir

$$l(\mathbf{x}) = \mathbf{x}^T C^{-1}\mathbf{s} \quad (3.18)$$

Por lo que el algoritmo de detección de la señal \mathbf{s} en presencia de ruido correlacionado es

$$T(\mathbf{x}) = \mathbf{x}^T \mathbf{C}^{-1} \mathbf{s} > \lambda' \quad (3.19)$$

Al comparar las ecuaciones (3.12) y (3.19), se puede observar que el procedimiento del detector cuando el ruido es correlacionado es igual al del ruido decorrelacionado, al que se le agrega un bloque que transforma dicho ruido. Un esquema de este detector se puede observar en la figura 3.5 [1].

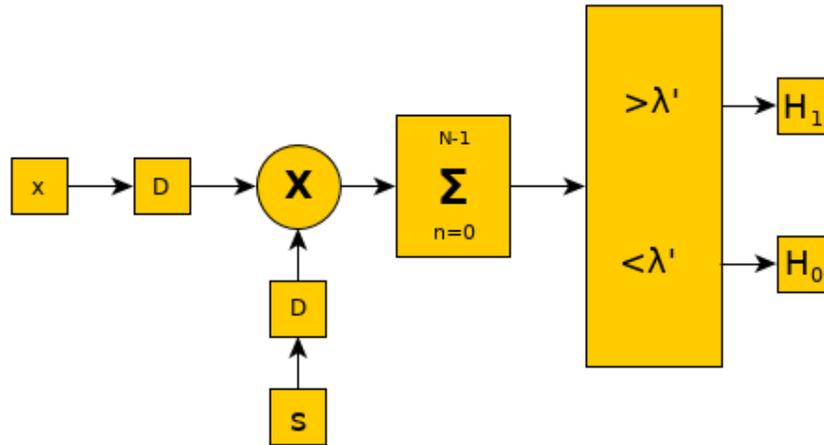


Figura 3.5: Diagrama a bloques del detector acoplado cuando se tiene ruido correlacionado

3.7. Razón de verosimilitud generalizada

En secciones anteriores se habló de hipótesis nulas y alternativas cuando se conocen los parámetros estadísticos de cada una de las señales junto con los ruidos del canal y del fondo.

Ahora, suponiendo que se tiene el seguimiento del problema de hipótesis:

- $H_0 : Y \sim N(0, 1)$ con parámetros estadísticos desconocidos.
- $H_1 : Y \sim N(0, 1)$ con parámetros estadísticos desconocidos.

La razón de verosimilitud generalizada (GLRT) es la siguiente:

$$L_G(\mathbf{x}) = \max_{\theta, \sigma_1^2, \sigma_0^2} \frac{P(\mathbf{x}, \bar{\theta}, \sigma_1^2 | H_1)}{P(\mathbf{x}, \bar{\theta}, \sigma_0^2 | H_0)} \quad (3.20)$$

Ahora supongamos que los parámetros tanto H_0 como de H_1 son desconocidos a priori. En la mayoría de casos la señal que refleja el objeto de interés es diferente para cada nueva prueba recibida. Pero su espectro de Fourier es estable y no cambia de una prueba a otra. Se puede designar la transformada de Fourier como una matriz \mathbf{H}

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_1 & \dots & \mathbf{h}_{p-1} \end{bmatrix} \quad (3.21)$$

cuyos $\mathbf{h}_k = \exp(-j2\pi \frac{kn}{N})$ y donde $k = 0, 1, 2, \dots, p-1$ es la frecuencia discreta y $n = 0, 1, 2, \dots, N-1$ es el tiempo discreto.

La fórmula de la señal particularmente desconocida es $\mathbf{s} = \mathbf{H}\mathbf{a}$, donde \mathbf{a} es el vector de magnitudes de los componentes espectrales que presentan la señal y sus desconocidos. La forma de la señal \mathbf{s} depende de \mathbf{a} , y si el vector \mathbf{a} es desconocido, la señal \mathbf{s} también lo es.

Para describir matemáticamente el procesamiento del detector, se puede usar el espacio multidimensional de todos los vectores posibles y separarlo en dos partes: subespacial donde se puede localizar la señal, y subespacial ortogonal a subespacial de la señal. Se puede introducir la matriz de proyector a ambos espacios.

$$\mathbf{P}_H = \mathbf{H} (\mathbf{H}^H \mathbf{H})^{-1} \mathbf{H}^H \quad (3.22)$$

El producto $\mathbf{P}_H \mathbf{s}$ contiene solamente ciertos armónicos de \mathbf{s} que le permite tener la matriz del proyector \mathbf{P}_H .

Cuando los parámetros de los PDF son desconocidos, en lugar del método de razón de verosimilitud, se usará el método de razón máxima de verosimilitud generalizada.

En este método, en lugar de parámetros \mathbf{a} que son desconocidos a priori, se sustituyen sus estimaciones.

$$L(\mathbf{x}) = \frac{\max_a p(\mathbf{x}|H_1)}{p(\mathbf{x}|H_0)} \quad (3.23)$$

Después de encontrar el numerador $\max_a p(\mathbf{x}|H_1)$ y sustituirlo, el algoritmo del detector subespacial acoplado es

$$\mathbf{x}^T \mathbf{P}_H \mathbf{x} > \lambda \quad (3.24)$$

Para el ruido correlacionado, el operador matricial que puede convertirlo a ruido decorre-

lacionado es $C^{-\frac{1}{2}}$, por lo que se tiene la siguiente fórmula:

$$\mathbf{x}^T \mathbf{C}^{-1} \mathbf{H} (\mathbf{H}^H \mathbf{C}^{-1} \mathbf{H})^{-1} \mathbf{H}^H \mathbf{C}^{-1} \mathbf{x} > \lambda \quad (3.25)$$

Los algoritmos basados en GLRT parten de la premisa de que, a pesar de que los parámetros estadísticos de ambas hipótesis pueden ser desconocidos, las características estadísticas del fondo en la hipótesis H_0 (es decir, el objeto no está presente) no son similares a las de H_1 (el objeto está presente). De hecho, en sistemas de vídeo infrarrojo, el blanco puede cubrir enteramente los píxeles en la superficie fluctuante y, en este caso, la señal recibida puede contener solo la señal del blanco adicionada con la del canal. Por lo tanto, la presencia del blanco elimina la aportación del fondo de la señal recibida [16].

3.8. Algoritmos de detección de la señal con potencia del fondo dependiente de la señal

Vamos a considerar los modelos de las señales recibidas con el fin de detectar la señal de interés dentro de las señales recibidas. En párrafos anteriores se usó el siguiente modelo de las señales recibidas:

- Para la hipótesis nula (H_0), el receptor recibe solamente ruido del canal. El canal es ruido que acumula propagación de luz en espacio libre.
- La hipótesis alternativa (H_1) considera situación cuando el receptor recibe reflexiones desde el objeto de interés y estas reflexiones durante propagación tiene distorsiones y modelo de distorsiones es ruido aditivo blanco gaussiano.

Este modelo es una base de desarrollo de algoritmos de detección en párrafos anteriores (3.12) y (3.19). Estos algoritmos fueron obtenidos aplicando una técnica estadística de creación de algoritmos de detección (LRT y GLRT).

Estos algoritmos garantizan una probabilidad de detección máxima posible con una probabilidad de falsa alarma elegida y fija. Cuando el modelo de la señal recibida cambia, los algoritmos obtenidos en párrafos anteriores no garantizan la probabilidad de detección máxima posible. Para otros tipos de modelos de señales recibidas hay que realizar un desarrollo de algoritmos de detección usando la misma técnica estadística de síntesis de algoritmos.

Se consideran las señales reflejadas desde la superficie marina y se usa una técnica de observación de espacio de los cuadros recibidos desde una cámara digital. Cada cuadro será separado por subespacios en forma de rectángulos con cierto ancho y altura.

El objetivo de proceso de detección es analizar el espacio rectángulo por rectángulo (ventana de análisis), y en cada ventana analizar cierto número de cuadros, procesar con las tres dimensiones (subespacio) y calcular un valor (estadístico), el cual se comparará con un umbral predeterminado. En este caso se puede considerar la hipótesis nula cuando la ventana de análisis no contiene el objeto de interés (completamente o particularmente). La hipótesis estadística alternativa considera la ventana de análisis cuando una parte de la ventana se encuentra total o parcialmente ocupada por el objeto de interés. Es claro que una parte de la ventana ocupada por el objeto no refleja a la cámara la señal del mar, y otra parte de la ventana desocupada refleja luz desde la superficie marítima.

Por lo tanto se puede separar la ventana de análisis por dos subespacios: subespacio ocupado por el objeto de interés (Ω_T), y otra parte de la ventana es subespacio ($\Omega \setminus \Omega_T$) donde no se encuentra el objeto de interés. Mencionamos que todos los subespacios reflejan las señales con cierta distorsión debido a la propagación de la luz, por lo tanto ambos subespacios contienen ruido de canal aditivo gaussiano blanco.

Consideramos el problema en el que se detecta un blanco multipíxel en la secuencia de K imágenes digitales con un fondo y ruido del canal, ambos aleatorios homogéneos y gaussianos. Asumimos que el objeto de tamaño N puede estar presente total o parcialmente en cualquier parte de la subimagen de tamaño L [15].

$$H_0 = \mathbf{x}_l = \mathbf{c}_l + \mathbf{w}_l, \quad l \in \Omega, \\ H_1 = \begin{cases} \mathbf{x}_i = \mathbf{s}_i + \mathbf{w}_i, & l \in \Omega_T, \\ \mathbf{x}_i = \mathbf{c}_i + \mathbf{w}_i, & l \in \Omega \setminus \Omega_T \end{cases} \quad (3.26)$$

donde $\Omega_T \equiv \{1, \dots, N\} \subset \Omega$ ($N \leq L$) es un subconjunto de enteros que indexan los píxeles, que pueden contener un objeto parcialmente desconocido bajo la hipótesis H_1 , y $\Omega \setminus \Omega_T \equiv \{N + 1, \dots, L\}$ es el subconjunto de los píxeles que no contienen al objeto bajo H_1 . Una representación gráfica en dos dimensiones se puede encontrar en la figura 3.6.

El modelo del mar en tres dimensiones es homogéneo, y sus parámetros estadísticos no dependen del número de cuadro ni del número del píxel en la ventana.

También se asume que los vectores homogéneos del fondo \mathbf{c}_l y del ruido del canal \mathbf{w}_l se modelan como vectores aleatorios normales de dimensión K (para K imágenes), por ejemplo

$$\begin{aligned}\mathbf{c}_l &\sim N(0, \sigma_c^2 \mathbf{I}) \\ \mathbf{w}_l &\sim N(0, \sigma_w^2 \mathbf{I})\end{aligned}\quad (3.27)$$

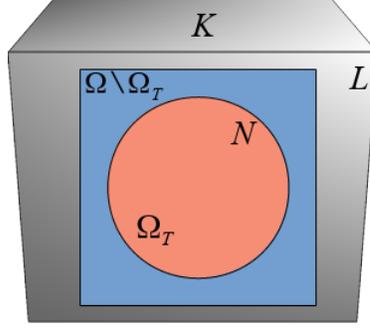


Figura 3.6: Esquema de fondo con objeto multipíxel

Si se supone que $\sigma_{c+w}^2 = \sigma_c^2 + \sigma_w^2$ bajo H_0 , y una varianza desconocida del fondo más el ruido $\hat{\sigma}_{1,i}^2$ bajo H_1 . Además supongamos que los vectores \mathbf{x}_i son independientes entre sí, esto implica que sus *pdfs* pueden escribirse como

$$\begin{aligned}L_1(\mathbf{x}_1, \dots, \mathbf{x}_L; \Omega_T, \hat{\sigma}_{1,i}^2, \boldsymbol{\theta}_i | H_1) &= \frac{c}{\left(\prod_{i=1}^L \det \hat{\sigma}_{1,i}^2 \mathbf{I}\right)^{\frac{1}{2}}} \prod_{i \in \Omega_T} \exp \left[-\frac{1}{2} (\mathbf{x}_i - \mathbf{s}_i)^T (\hat{\sigma}_{1,i}^2 \mathbf{I})^{-1} \right. \\ &\quad \left. (\mathbf{x}_i - \mathbf{s}_i) \right] \times \prod_{j \in \Omega \setminus \Omega_T} \exp \left[-\frac{1}{2} (\mathbf{x}_j^T \hat{\sigma}_{1,j}^2 \mathbf{I})^{-1} \mathbf{x}_j \right]\end{aligned}\quad (3.28)$$

$$L_0(\mathbf{x}_1, \dots, \mathbf{x}_L; | H_0) = \frac{c}{[\det \sigma_{c+w}^2 \mathbf{I}]^{\frac{L}{2}}} \exp \left[-\frac{1}{2} \sum_{i=1}^L \mathbf{x}_i^T (\sigma_{c+w}^2 \mathbf{I})^{-1} \mathbf{x}_i \right] \quad (3.29)$$

donde $\hat{\sigma}_{1,i}^2 = \frac{\mathbf{x}_i^T \mathbf{P}_s^\perp}{K-p}$ para $i \in \Omega_T$, $\hat{\sigma}_{1,i}^2 = \frac{\mathbf{x}_i^T \mathbf{x}_i}{K-p}$ para $i \in \Omega \setminus \Omega_T$, p es el rango del subespacio del blanco \mathbf{H} , $\mathbf{P}_s^\perp = \mathbf{I} - \mathbf{P}_s$ es la matriz de proyección ortogonal de tamaño $K \times K$ en el subespacio ortogonal al subespacio de la señal \mathbf{H} , y \mathbf{P}_s es la matriz de proyección ortogonal de tamaño $K \times K$ en el subespacio de la señal \mathbf{H} .

El algoritmo propuesto, al que se le llamará MMSD¹, se da por el siguiente estadístico:

$$T = \sum_{i=1}^L \left(\frac{\mathbf{x}_i^T \mathbf{x}_i}{K \sigma_{c+w}^2} - \frac{K-p}{K} - \ln \frac{\mathbf{x}_i^T \mathbf{P}_s^\perp \mathbf{x}_i}{(K-p) \sigma_{c+w}^2} \right) \quad (3.30)$$

La razón de expresar todo en términos de una sumatoria se debe a que el producto de una exponencial es igual a la exponencial de una sumatoria.

Se puede reescribir este estadístico de la siguiente forma:

$$T = \sum_{i=1}^L \left(\frac{\mathbf{x}_i^T \mathbf{P}_s \mathbf{x}_i}{K \sigma_{c+w}^2} + \frac{K-p}{K} \frac{\mathbf{x}_i^T \mathbf{P}_s^\perp \mathbf{x}_i}{(K-p) \sigma_{c+w}^2} - \frac{K-p}{K} - \ln \frac{\mathbf{x}_i^T \mathbf{P}_s^\perp \mathbf{x}_i}{(K-p) \sigma_{c+w}^2} \right) \quad (3.31)$$

donde $\mathbf{P}_s^\perp = \mathbf{I} - \mathbf{P}_s$ proyecta la señal recibida al subespacio de armónicos de Fourier donde nunca existe la señal del objeto de interés.

En el caso en el que $p \ll K$, el estadístico se puede reescribir como

$$T \approx \sum_{i=1}^L \left(\frac{\mathbf{x}_i^T \mathbf{P}_s \mathbf{x}_i}{K \sigma_{c+w}^2} + \frac{\mathbf{x}_i^T \mathbf{P}_s^\perp \mathbf{x}_i}{(K-p) \sigma_{c+w}^2} - \ln \frac{\mathbf{x}_i^T \mathbf{P}_s^\perp \mathbf{x}_i}{(K-p) \sigma_{c+w}^2} - 1 \right) = T_{MSD} + T_{ad} \quad (3.32)$$

donde

$$T_{MSD} = \sum_{i=1}^L \frac{\mathbf{x}_i^T \mathbf{P}_s \mathbf{x}_i}{K \sigma_{c+w}^2} \quad (3.33)$$

y

$$T_{ad} = \sum_{i=1}^L \left(\frac{\mathbf{x}_i^T \mathbf{P}_s^\perp \mathbf{x}_i}{(K-p) \sigma_{c+w}^2} - \ln \frac{\mathbf{x}_i^T \mathbf{P}_s^\perp \mathbf{x}_i}{(K-p) \sigma_{c+w}^2} - 1 \right) \quad (3.34)$$

En este caso, se combinan los detectores acoplados subspaceales multipixel T_{MSD} y el detector de cambio de energía del fondo más el ruido multipixel T_{ad} en un esquema único.

3.9. Conclusiones

En este capítulo se mostró el desarrollo de los algoritmos LRT y GLRT, así como los parámetros físicos, matemáticos y estadísticos que estos involucran. También se muestra el

¹El algoritmo originalmente es propuesto como MMMSD, se acorta a MMSD en este documento para una mejor comprensión.

desarrollo de un nuevo algoritmo basado en GLRT que matemáticamente mejora su desempeño considerando la energía de los píxeles que no contienen información útil del objeto.

Se puede adelantar que el algoritmo GLRT y el nuevo algoritmo propuesto en este capítulo serán puestos a prueba, primero en simulación de fondo y objeto y finalmente con muestras de video real del mar.

Capítulo 4

Pruebas de desempeño

En este capítulo se presentarán los resultados experimentales de los algoritmos dados en las secciones anteriores. Se tienen programas que permiten analizar las señales bajo ruido blanco y bajo ruido correlacionado.

4.1. Simulación de una secuencia de imágenes con ruido blanco

Teniendo un programa escrito en MATLAB donde es posible simular las señales del mar, del objeto y el ruido de fondo y del canal, y donde los mismos se pueden variar a voluntad, se pueden probar en un ambiente controlado los algoritmos presentados en la sección anterior. Las gráficas mostradas compararán la probabilidad de detección P_D contra la razón señal a fondo SBR expresada en decibeles.

El detector analiza una subventana de espacio de L pixeles con K cuadros consecutivos que forman una figura en 3 dimensiones de datos (paralelepípedo). Un detector debe procesar con todos los pixeles dentro del paralelepípedo y formar un valor de su salida. Este valor compara con un valor específico (umbral) y tomar una decisión. Si la salida del detector es mayor que el umbral, existe un objeto dentro del paralelepípedo. En caso contrario, el sistema toma la decisión de dentro del paralelepípedo no existe el objeto de interés.

Los parámetros de las señales son los siguientes:

- Señal útil: Señal determinística con p armónicas de Fourier, generada con el comando **ones** o **rand** de MATLAB, fija en tres dimensiones.
- Fondo: Distribución normal generada aleatoriamente (ruido gaussiano blanco) con una cantidad de armónicos determinado por el parámetro K .
- Ruido del canal: Distribución normal generada aleatoriamente (ruido gaussiano blanco) con una cantidad de armónicos determinado por el parámetro K .

En este programa se variarán los siguientes parámetros:

- Tamaño del blanco $N = 2$, $N_1 = 10$ y $N_2 = 20$.
- Razón w de potencia del mar entre potencia del canal
- Número de cuadros K de análisis
- Número de armónicos p de la señal útil
- Tamaño L de la ventana de análisis

Supóngase que, si el blanco ocupa dentro de su ventana un pixel, todos los cuadros de este pixel están ocupados por el blanco dentro del paralelepípedo. También supóngase que si el blanco tiene movimiento, durante K cuadros, este es pequeño. Para un $L = 20$, los tamaños del blanco $N = 2$, $N_1 = 10$ y $N_2 = 20$ significan que el blanco ocupa una décima parte, la mitad y por completo la ventana de análisis respectivamente.

Se debe considerar que el subespacio de la señal útil s siempre es menor que el número K de cuadros. La relación entre K y p se determina por las siguientes condiciones:

- Para valores impares de p , $p < \frac{K+1}{2}$
- Para valores pares de p , $p < \frac{K}{2} + 1$

Por lo que, al proponer valores K de 20 y 50, el número de armónicos máximo debe ser 10 y 25 respectivamente, los cuales representan el número de armónicos de la señal del ruido del mar.

Para comparar calidad de detección usando diferentes algoritmos, se contará con probabilidad de detección (P_D) que garantiza un algoritmo con probabilidad de falsa alarma (P_{FA})

elegida a priori y constante. Cuando observamos un objeto sin distorsiones de ruido, la probabilidad de detección siempre será igual a 1, y la probabilidad de falsa detección siempre será cero. Esta situación cambia cuando observamos un objeto en presencia de ruido. El detector tiene que distinguir la energía desde el objeto y la energía de los ruidos. Cuando la potencia del ruido crece con respecto a la potencia de la señal del blanco, la probabilidad de detección disminuye, y la probabilidad de falsa detección crece.

Por lo tanto, un factor principal que afecta la probabilidad de detección es la relación de potencia desde el blanco entre potencia del ruido, y una técnica de comparación de eficiencia de los algoritmos de detección es trazar gráficas de probabilidad de detección contra razón de la potencia de la señal útil entre la potencia del ruido cuando la probabilidad de falsa detección es fija.

Simulando un paralelepípedo de tres dimensiones, las figuras 4.1 y 4.2 muestran la señal útil modeladas con los comandos *ones* y *rand* respectivamente, esto es, un cuadro del vídeo, de tamaño $N \times N = 2 \times 2$. Mientras que las figuras 4.3 y 4.4 muestran las mismas señales vistas “desde arriba”, es decir, se puede ver el comportamiento de las variaciones de la señal útil cuadro a cuadro el cual, para este caso, $K = 20$.

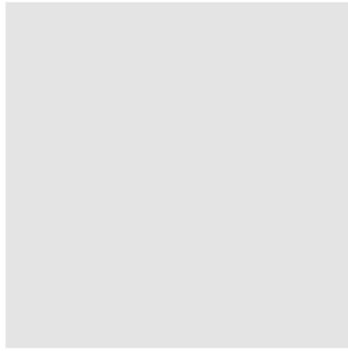


Figura 4.1: Señal útil modelada con el comando *ones* de MATLAB.

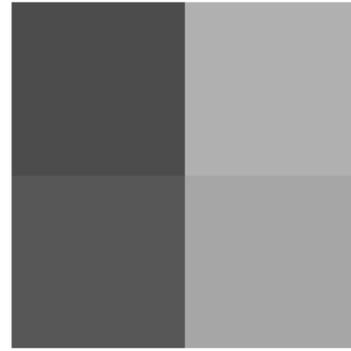


Figura 4.2: Señal útil modelada con el comando *rand* de MATLAB.



Figura 4.3: Señal útil modelada con el comando *ones* de MATLAB.



Figura 4.4: Señal útil modelada con el comando *rand* de MATLAB.

De la misma forma, se muestran las imágenes de la señal útil sumada con el fondo más el ruido del canal como cuadros de vídeo y de forma lateral en las figuras 4.5, 4.6, 4.7 y 4.8. Estas señales también tienen un $K = 20$.



Figura 4.5: Señal útil modelada con el comando *ones* de MATLAB más fondo decorrelacionado y ruido del canal.



Figura 4.6: Señal útil modelada con el comando *rand* de MATLAB más fondo decorrelacionado y ruido del canal.



Figura 4.7: Señal útil modelada con el comando *ones* de MATLAB más fondo decorrelacionado y ruido del canal, vista desde el plano en tres dimensiones.



Figura 4.8: Señal útil modelada con el comando *rand* de MATLAB más fondo decorrelacionado y ruido del canal, vista desde el plano en tres dimensiones.

En la figura 4.9 se pueden observar las gráficas de frecuencia del fondo con el que se trabajará en esta subsección.

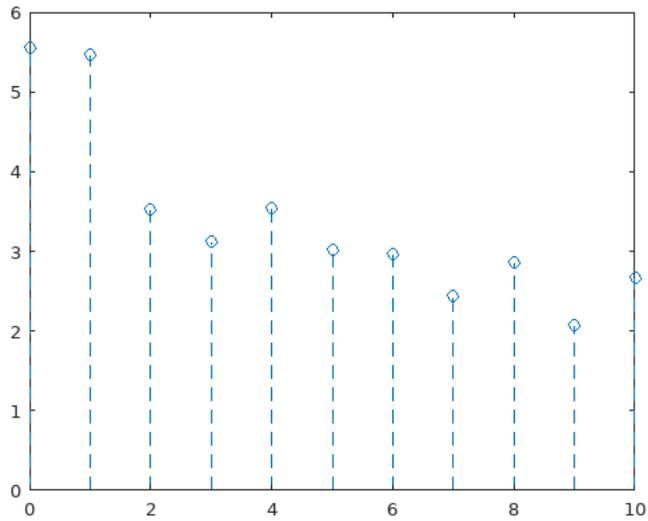


Figura 4.9: Espectro de fondo decorrelacionado

En las figuras 4.10 y 4.11 se puede apreciar el comportamiento de gráficas cuando $L = 20$, $N = 2$, $N_1 = 10$, $N_2 = 20$, $K = 20$, $w = 0.6$ son constantes, y se varía p .

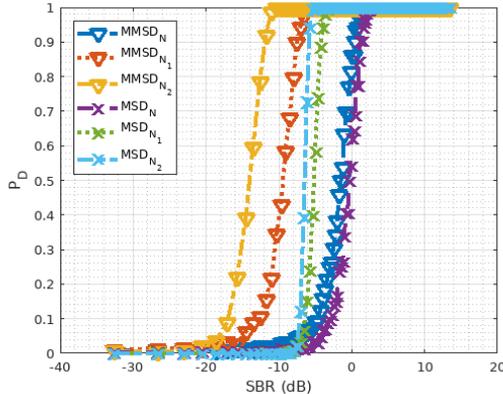


Figura 4.10: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01$, $p = 4$

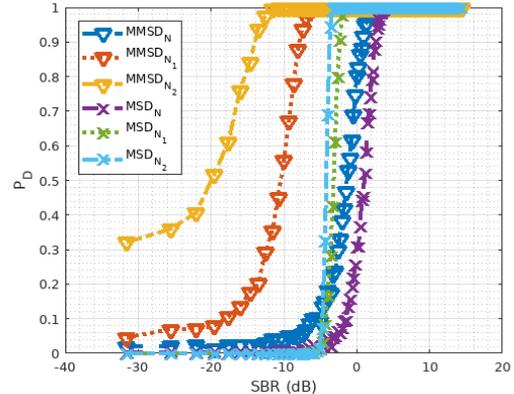


Figura 4.11: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01$, $p = 8$

Para comparar la eficiencia de los algoritmos, se elegirá un nivel de calidad $P_D = 0.9$ con $P_{FA} = 0.01$. En las gráficas de las figuras 4.10 y 4.11, se puede notar que los algoritmos MMSD y MSD son sensibles al número de armónicos de la señal útil y al número de píxeles ocupados dentro del paralelepípedo de análisis.

Para explicar este comportamiento hay que notar que el ruido del mar y del canal tiene 11

componentes espectrales. Cuando la señal útil tiene pocas armónicas ($p = 4$), la diferencia entre el comportamiento del mar y del blanco es suficientemente grande. Cuando p crece, la diferencia del comportamiento entre el mar y el objeto baja.

También se puede mencionar analizando las gráficas de estas figuras que la eficiencia de detección crece cuando el tamaño del objeto dentro del paralelepípedo crece.

El algoritmo MMSD alcanza un P_D de 0.9 a niveles de SBR menores que su contraparte MSD. También se puede observar que cuando $p = 4$, los algoritmos funcionan ligeramente mejor que con $p = 8$. Esto significa que a mayor cantidad de armónicos de la señal útil, el desempeño de los algoritmos decae.

El variar el parámetro w implica que la imagen se volverá más ruidosa. Si trasladamos este parámetro a un plano físico, un valor cercano a 0 indicaría un video del mar totalmente despejado, sin lluvia o niebla alguna. En cambio, mientras más se acerque a 1, esto indicaría que hay una tormenta en curso con lluvias torrenciales que impediría una visión clara del mar.

Ahora variaremos el parámetro w a un valor de 0.9. En las figuras 4.12 y 4.13 se observa un comportamiento similar a las figuras 4.10 y 4.11 respectivamente.

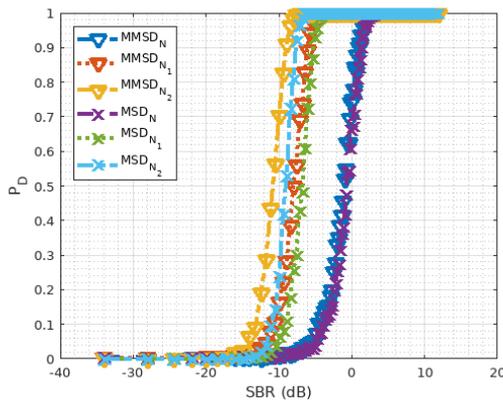


Figura 4.12: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. Parámetros $P_{FA} = 0.01$, $p = 4$

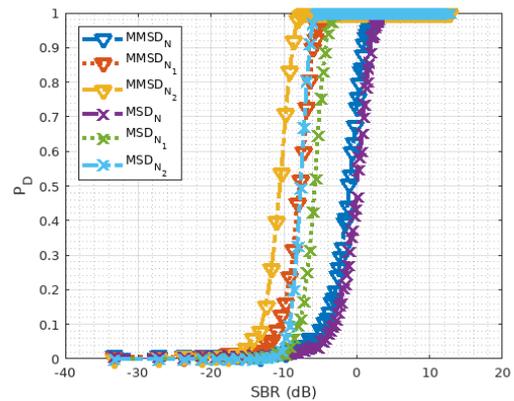


Figura 4.13: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01$, $p = 8$

Sin embargo, al observar estas figuras, se puede observar que las gráficas para tanto MMSD como para MSD muestran un desempeño pobre con respecto a las demás gráficas. Aun así, se observa que las gráficas para el algoritmo MMSD siguen teniendo una mejor respuesta que sus contrapartes MSD.

Ahora se cambiará la señal útil de **ones** a **rand**, el cual produce números reales aleatorios distribuidos uniformemente en un intervalo de 0 a 1. Físicamente podría representarse como un objeto que cuya energía no es totalmente uniforme en todos sus pixeles.

Se procede a repetir las figuras 4.10 y 4.11, es decir, con $w = 0.6$. Los resultados se muestran en las figuras 4.14 y 4.15.

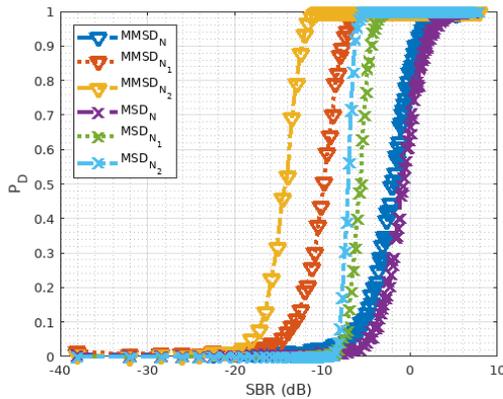


Figura 4.14: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01$, $p = 4$

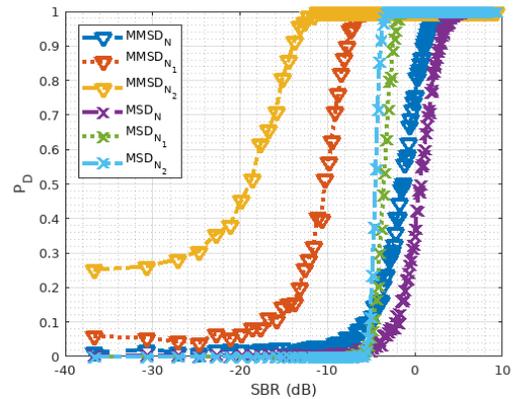


Figura 4.15: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01$, $p = 8$

Aunque a simple vista las figuras pueden parecer diferentes, se puede ver que en las figuras 4.14 y 4.15, el comportamiento de las gráficas en comparación con la señal útil modelada en **ones** es muy similar y no presentan diferencias significativas.

Y ahora con $w = 0.9$, se presentan las gráficas que aparecen en las figuras 4.16 y 4.17.

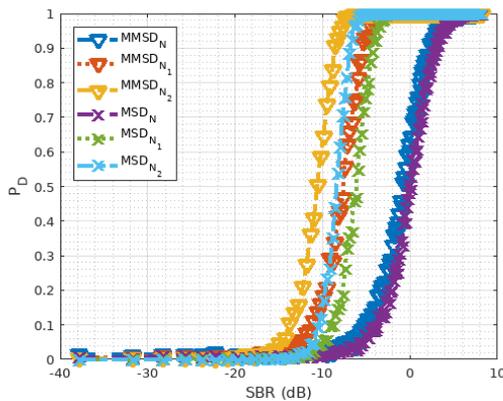


Figura 4.16: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01$, $p = 4$

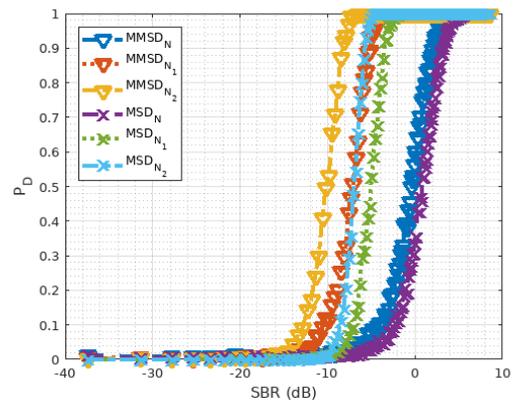


Figura 4.17: P_D contra SBR cuando modelo del mar es ruido blanco gaussiano. $P_{FA} = 0.01$, $p = 8$

En las figuras anteriores se mostró que bajo las condiciones dadas y un ruido blanco gaussiano, el algoritmo MMSD funciona mejor que su contraparte MSD. Cabe señalar que no se presentan las gráficas para valores de w más bajos que 0.6 debido a que las curvas de $MMSD_{N_1}$ y $MMSD_{N_2}$ siempre tendían a una línea recta de $P_D = 1$ para todo valor de SBR, y al encimarse una sobre la otra, la identificación de cada una de las gráficas era extremadamente difícil.

4.2. Simulación de una secuencia de imágenes con ruido correlacionado

Ahora se procederá a trabajar con un fondo con ruido correlacionado. A diferencia del ruido blanco, el ruido correlacionado tiene una distribución espectral de potencia no uniforme.

En este programa se repetirán las pruebas realizadas en la subsección anterior, con la diferencia de que incluiremos un parámetro más llamado **índice de correlación**.

En este caso, el modelo de reflexión de la luz desde la superficie marina es un proceso gaussiano aleatorio con valores correlacionados entre cuadros. MATLAB proporciona un proceso normal con valores decorrelacionados (ruido blanco), por lo que, para crear valores aleatorios correlacionados se utiliza la siguiente fórmula:

$$r = \exp [-(1 - corr) |j_1 - j_2|] \quad (4.1)$$

donde $corr$ es el índice de correlación.

Las señales mostradas en la subsección anterior (4.1, 4.2, 4.3 y 4.4) no cambian para este experimento, por lo que se muestran a continuación únicamente las figuras 4.18, 4.19, 4.20 y 4.21 correspondientes a la señal útil más fondo correlacionado y ruido del canal vistos como un cuadro de vídeo y como una cara de un paralelepípedo de 3 dimensiones con $K = 20$ cuadros.

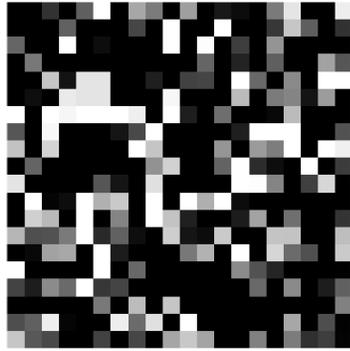


Figura 4.18: Señal útil modelada con el comando *ones* de MATLAB más fondo correlacionado y ruido del canal.

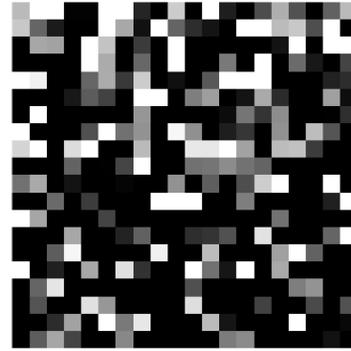


Figura 4.19: Señal útil modelada con el comando *rand* de MATLAB más fondo correlacionado y ruido del canal.



Figura 4.20: Señal útil modelada con el comando *ones* de MATLAB más fondo correlacionado y ruido del canal, visto en tercera dimensión.



Figura 4.21: Señal útil modelada con el comando *rand* de MATLAB más fondo correlacionado y ruido del canal, visto en tercera dimensión.

Para convertir el ruido decorrelacionado en correlacionado, se usan dos instrucciones: *levinson* y *filter*. La recursión Levinson-Durbin ayuda a encontrar los polos del filtro IIR (Respuesta al impulso infinito) con una secuencia de correlación determinada. El comando *filter* utiliza el vector de coeficientes del denominador de la función de transferencia, el numerador y los datos que se necesitan convertir.

Para las siguientes pruebas se pondrá este valor constante e igual a 0.9. El valor de este parámetro se puede variar entre 0 (sin correlación) y 1 (totalmente correlacionado).

En la figura 4.22 se pueden observar las gráficas de frecuencia del fondo con el que se trabajará en esta subsección.

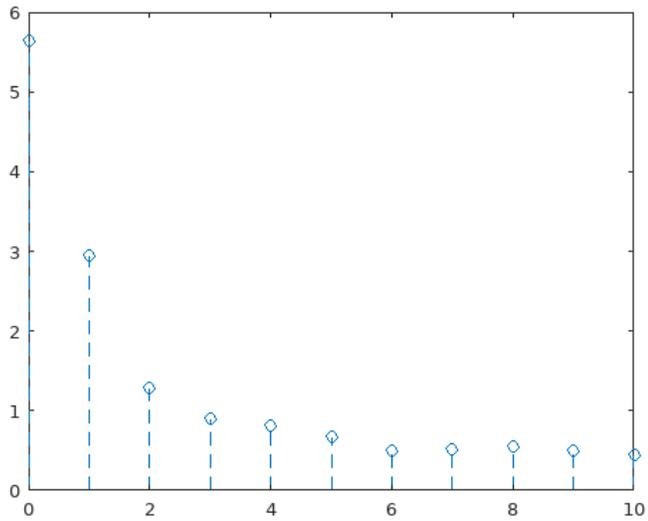


Figura 4.22: Espectro de fondo correlacionado

En las figuras 4.23 y 4.24 se muestran las gráficas para MMSD y para MSD cuando $L = 20$, $N = 2$, $N_1 = 10$, $N_2 = 20$, $K = 20$, $w = 0.6$ son constantes, y se varía p y la señal útil se modela con el comando *ones* de MATLAB.

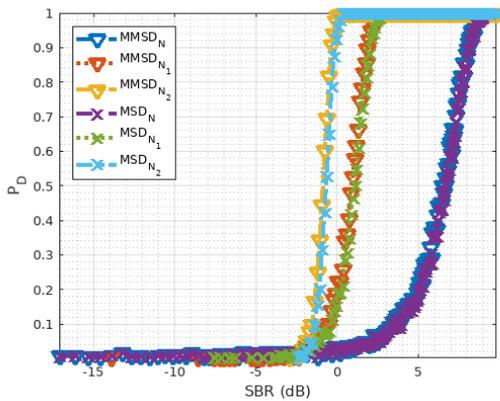


Figura 4.23: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 4$

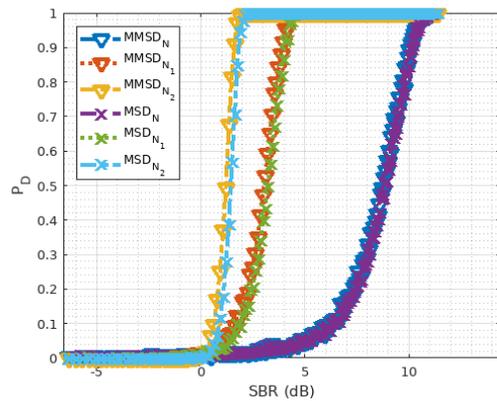


Figura 4.24: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 8$

Se puede notar que, aunque las diferencias son mínimas entre una figura y otra, el desempeño del algoritmo MMSD se comporta muy similar o ligeramente mejor al MSD.

Ahora se procede a modificar el valor w a 0.9. Las figuras 4.25 y 4.26 muestran el comportamiento de las curvas de probabilidad de detección.

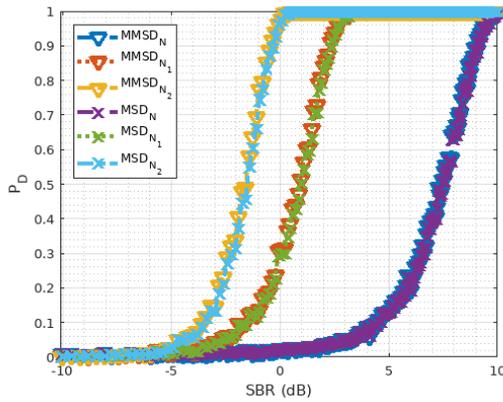


Figura 4.25: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 4$

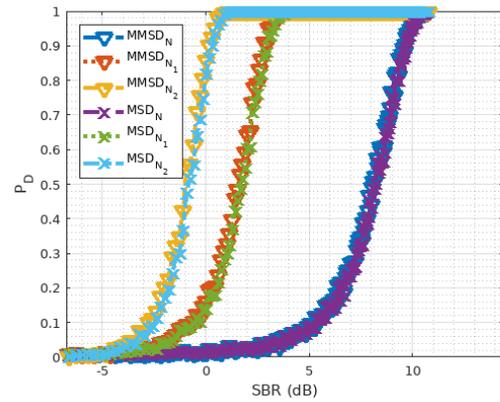


Figura 4.26: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 8$

Ahora se procede a modificar la señal útil a *rand*. Las figuras 4.27 y 4.28 muestran el comportamiento de las curvas de probabilidad de detección.

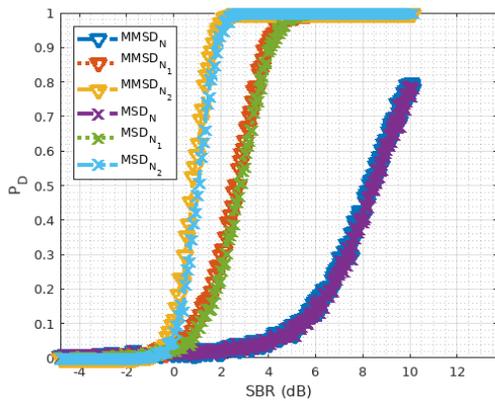


Figura 4.27: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 4$

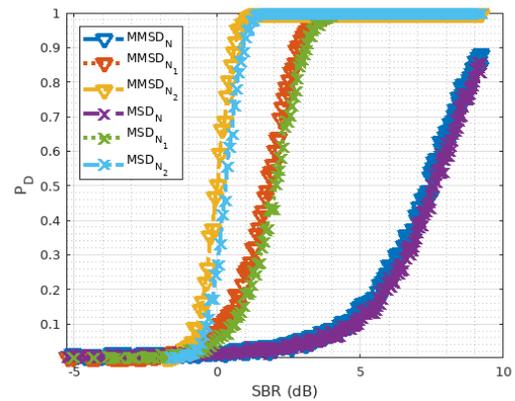


Figura 4.28: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 8$

Se puede notar en ambas figuras y en las anteriores que el algoritmo propuesto siempre dará un resultado igual o mejor que el algoritmo MSD clásico.

Ahora se procede a modificar el valor w a 0.9. Las figuras 4.29 y 4.30 muestran el comportamiento de las curvas de probabilidad de detección.

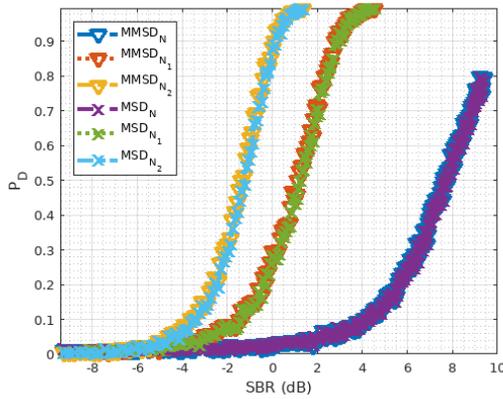


Figura 4.29: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 4$

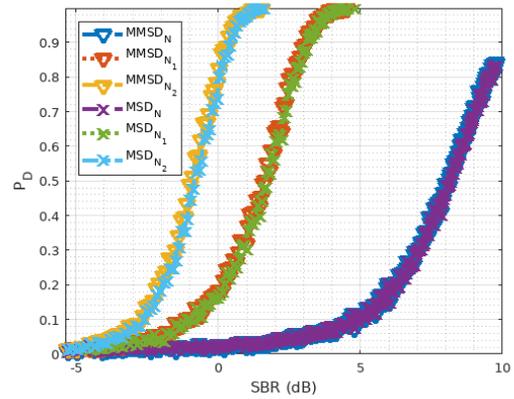


Figura 4.30: P_D contra SBR cuando modelo del mar es ruido correlacionado. $P_{FA} = 0.01$, $p = 8$

Las figuras anteriores nos muestran gráficamente la suposición matemática de que el algoritmo MMSD solo puede funcionar igual o mejor que el algoritmo clásico MSD.

4.3. Conclusiones

- Los algoritmos MMSD y MSD fueron probados bajo las mismas condiciones en un entorno controlado, por lo que es posible hacer una comparación.
- Conforme la frecuencia del objeto comienza a acercarse a la frecuencia del mar, ambos algoritmos comienzan a mermar su desempeño, sin embargo el comportamiento de MMSD nunca tiende a ser peor que MSD.
- En un fondo con ruido blanco, ambos algoritmos tienen un mejor desempeño que en un fondo con ruido correlacionado. Esto es de vital importancia en las pruebas experimentales ya que en un entorno real no se puede controlar el ruido. Aun así existen herramientas que permiten convertir el ruido correlacionado a blanco. No se ha estudiado en este trabajo el costo computacional de las mismas.

Capítulo 5

Pruebas experimentales

En el capítulo anterior se probaron los algoritmos anteriores en un entorno controlado, lo que implica que tanto el objeto como el mar, los niveles de ruido y la duración del video pueden ser controlados a voluntad. En un entorno real estos parámetros son imposibles de controlar y se está sujeto a las condiciones del ambiente, del objeto, del mar y de la cámara que se está utilizando.

En este capítulo se abordará la metodología para las pruebas con video real considerando las limitantes anteriores.

5.1. Muestras seleccionadas

Es necesario tener diferentes muestras de video con diferentes entornos de mar y objetos. De esa forma se tienen los siguientes escenarios:

- Video con un nadador sosteniendo dos bolas, una blanca y una negra, como se aprecia en la figura 5.1, al que se le llamará "video 1".
- Video con un nadador cruzando la superficie del mar, como se aprecia en la figura 5.2, al que se le llamará "video 2".
- Video de dos objetos desconocidos flotando en un mar agitado, como se aprecia en la figura 5.3, al que se le llamará "video 3".

Para cada video se realizan iteraciones de pruebas estadísticas de acuerdo a la cantidad de cuadros de video que lo conforman, manteniendo un patrón de parámetros a medir.

- Ventana de análisis de 5×5 , 10×10 y 20×20 píxeles, a excepción del video 3 cuyas ventanas son de 5×5 , 7×7 y 10×10 .
- 20 iteraciones conformadas por las ventanas anteriores y a 20 cuadros de video cada una.



Figura 5.1: Nadador sosteniendo dos bolas, una blanca y una negra Figura 5.2: Nadador con flotador cruzando el mar



Figura 5.3: Objetos flotando en el mar

Para las siguientes secciones de la tesis se definen dos ventanas de análisis:

- Una ventana de análisis donde no se encuentra el objeto (H_0).

- Una ventana de análisis donde se cree que podría estar el objeto (H_1).

Si se presenta un video como una matriz de tres dimensiones, se puede representar físicamente como se puede ver en la figura 5.4. El video a analizar se puede ver como el todo, el cual será dividido en un número n de iteraciones, donde cada iteración contará con una cantidad fija de cuadros de video.

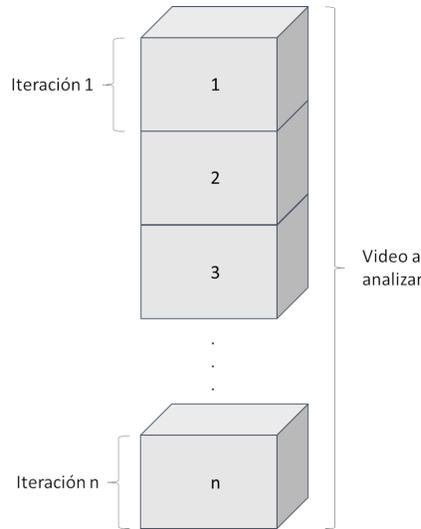


Figura 5.4: Representación gráfica del análisis de video

Para este tipo de pruebas se deben tomar en cuenta las siguientes consideraciones:

- Ambas ventanas deben ser capaces de contener la misma cantidad de píxeles y tener la misma cantidad de cuadros por iteración.
- Los parámetros estadísticos como media y varianza de las ventanas de análisis a utilizar deben normalizarse, es decir, obtener los parámetros estadísticos del mar sin objeto y sustraerse de ambas ventanas de análisis.
- Las matrices generadas por MATLAB deberán trabajarse en un rango de 0 (totalmente negro) a 255 (totalmente blanco). Normalizar a valores de 0 a 1 podría provocar pérdida de información y resultados erróneos al descartar cifras decimales.

5.2. Análisis de pruebas estadísticas de los videos en tercera dimensión

Para esta parte experimental se debe tener en cuenta que se convertirá una matriz de tres dimensiones (ancho \times alto \times cuadros a analizar) en una matriz de dos dimensiones (ancho por alto \times cuadros a analizar). Es de suma importancia hacer notar que en una aplicación en tiempo real este tipo de conversión es imposible de hacer.

Se procede a analizar el video 1. La ventana de análisis tiene un tamaño de $5 \times 5 \times 20$ pixeles, $10 \times 10 \times 20$ pixeles, o $20 \times 20 \times 20$ pixeles, y se realizan 20 iteraciones en los cuadros subsecuentes del video. Se selecciona la cabeza del nadador como se puede apreciar en la figura 5.1.

Las figuras 5.5 hasta la 5.10 muestran las escenas de video con diferentes tamaños de ventanas de análisis y las comparaciones entre los algoritmos MSD y MMSD. La figura 5.5 muestra dos pares de barras, y en cada par, del lado izquierdo la gráfica correspondiente a H_0 , y del lado derecho a H_1 . Mientras más alejada esté la media de la barra correspondiente a H_1 de la media de la barra de H_0 , significa que H_1 se cumple más veces.

En dichas figuras se puede ver que conforme va creciendo la ventana de análisis, el objeto ocupa menos cantidad de pixeles de la ventana de análisis, sin embargo las gráficas correspondientes a MMSD muestran un desempeño mucho mejor que el que se aprecia con MSD. También se debe tomar en cuenta de que el objeto se va moviendo a través de la ventana de análisis, por lo que es posible que en muchos cuadros de video parte o nada del objeto se encuentre dentro de la ventana.

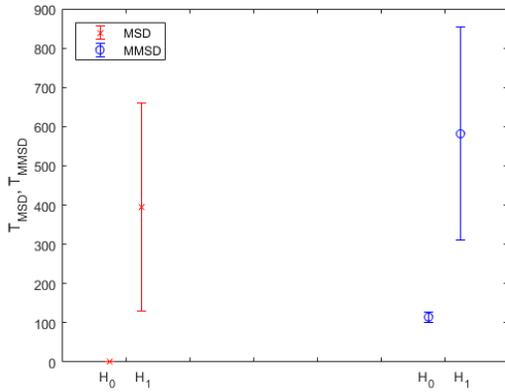


Figura 5.5: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.6

Figura 5.6: Nadador con dos bolas a su alrededor con una ventana de análisis de 5×5 píxeles

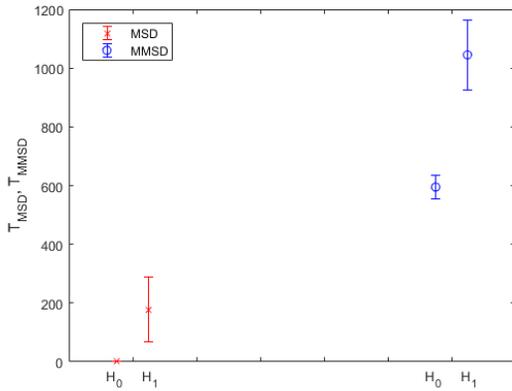


Figura 5.7: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.8

Figura 5.8: Nadador con dos bolas a su alrededor con una ventana de análisis de 10×10 píxeles

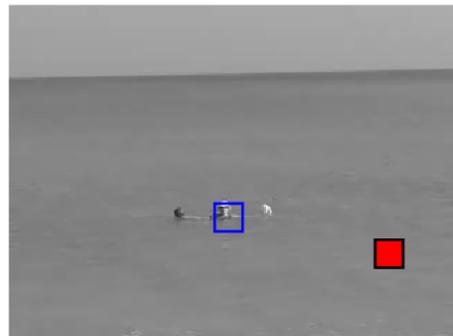
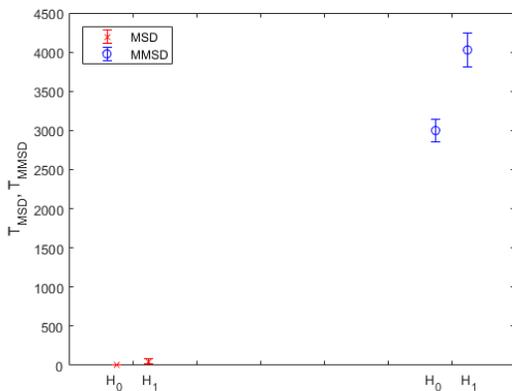


Figura 5.9: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.10

Figura 5.10: Nadador con dos bolas a su alrededor con una ventana de análisis de 20×20 píxeles

Ahora se procede a analizar el video 2. La particularidad de este video es que el objeto a analizar es un nadador que se desplaza de derecha a izquierda a lo largo del video, por lo que muchos de los cuadros no contienen al nadador, o solo contienen una parte de su cuerpo (ver figura 5.2 para comparación).

Las figuras 5.11 hasta la 5.16 muestran, de manera similar que con el video 1, los diferentes tamaños de ventanas de análisis y el comportamiento de los algoritmos.

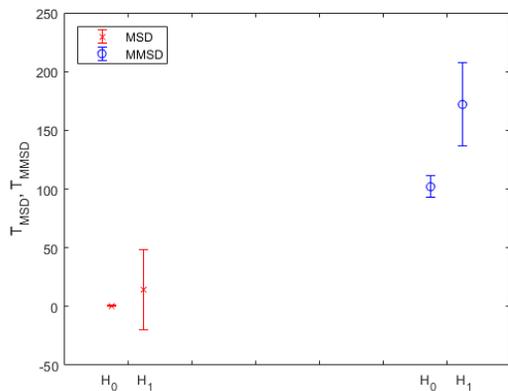


Figura 5.11: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.12

Figura 5.12: Nadador cruzando el mar con una ventana de análisis de 5×5 píxeles

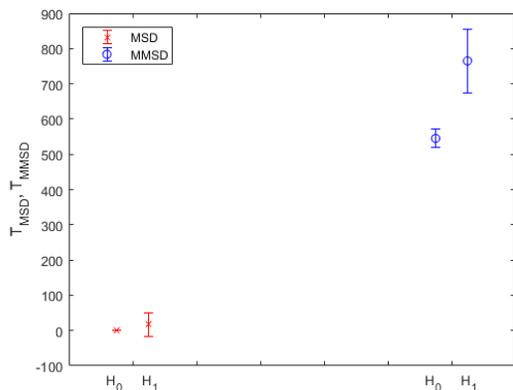


Figura 5.13: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.14

Figura 5.14: Nadador cruzando el mar con una ventana de análisis de 10×10 píxeles

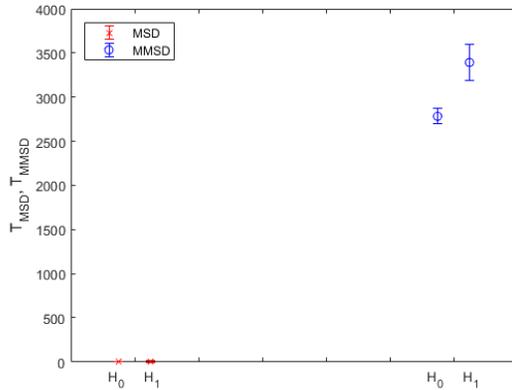


Figura 5.15: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.16

Figura 5.16: Nadador cruzando el mar con una ventana de análisis de 20×20 píxeles

Ahora se vuelve a utilizar el video 2, esta vez para tomar otra ventana, donde debería encontrarse un barco a la distancia. Esta prueba permite determinar si los algoritmos son capaces de detectar objetos pequeños dentro de las ventanas de análisis. Las figuras 5.17 a la 5.22 muestran el comportamiento con diferentes ventanas de análisis. En las figuras se puede observar que el comportamiento de MMSD sigue siendo superior al de MSD, sobre todo en los escenarios en que el objeto solo cubre una parte de la ventana de análisis.

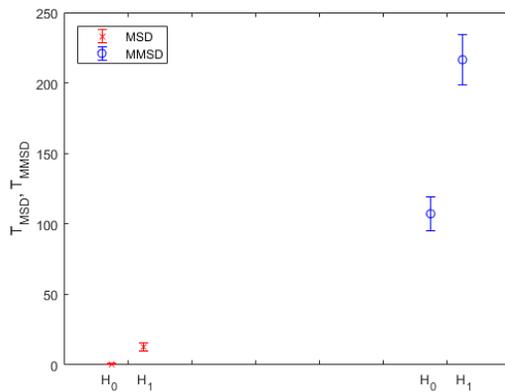


Figura 5.17: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.18

Figura 5.18: Barco en el mar con una ventana de análisis de 5×5 píxeles

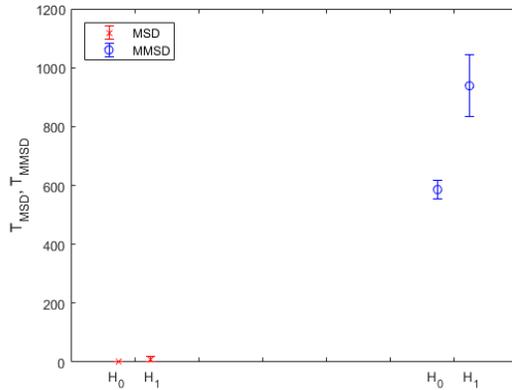


Figura 5.19: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.20

Figura 5.20: Barco en el mar con una ventana de análisis de 10×10 píxeles

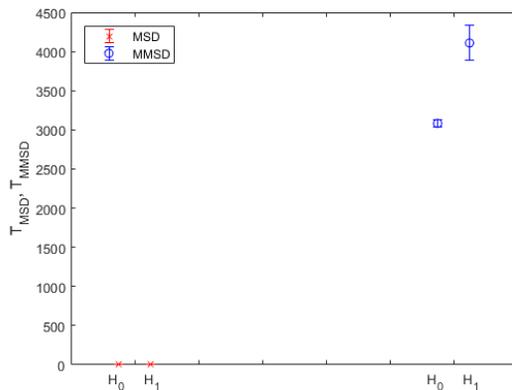


Figura 5.21: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.22

Figura 5.22: Barco en el mar con una ventana de análisis de 20×20 píxeles

Se repite la prueba con el video 3, objetos flotando en un mar agitado. Debido al tamaño de los objetos que se encuentran flotando, se decidió que las ventanas de análisis tendrían los siguientes tamaños:

- 5×5
- 7×7
- 10×10

Se puede ver desde la figura 5.23 hasta la 5.28 que el comportamiento de MMSD y MSD es similar a las pruebas anteriores, siendo mejor en todos los casos MMSD.

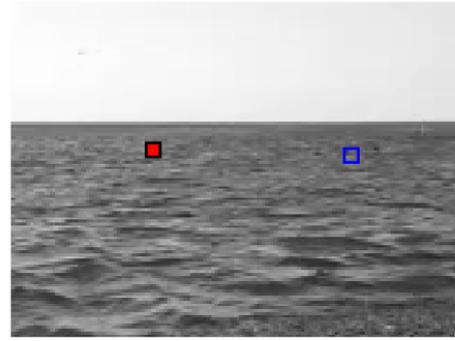
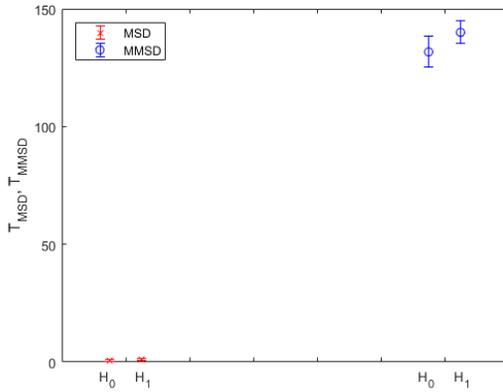


Figura 5.23: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.24

Figura 5.24: Objetos en el mar con una ventana de análisis de 5×5 píxeles

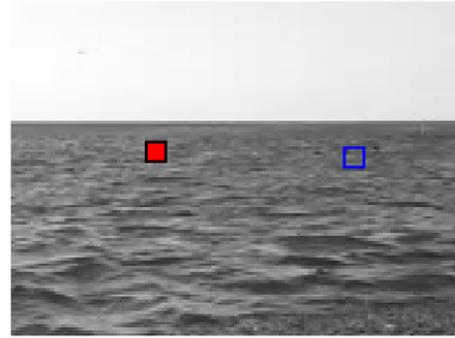
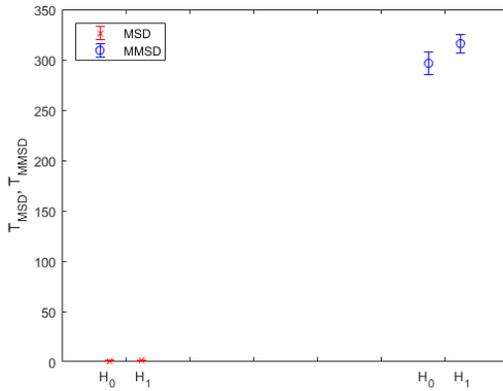


Figura 5.25: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.26

Figura 5.26: Objetos en el mar con una ventana de análisis de 7×7 píxeles

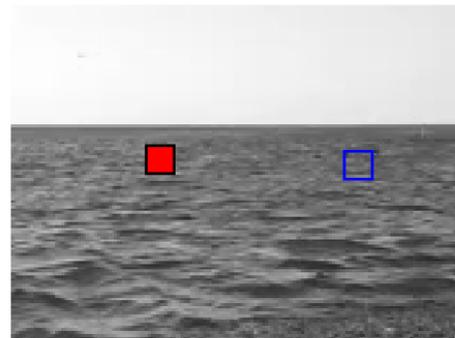
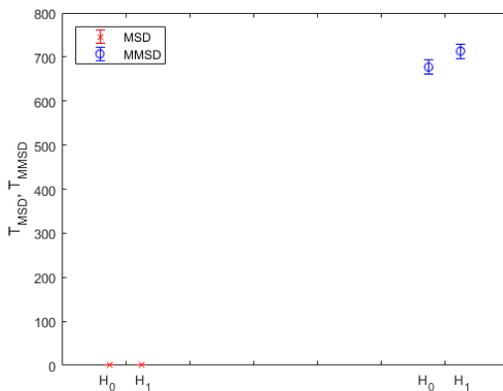


Figura 5.27: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.28

Figura 5.28: Objetos en el mar con una ventana de análisis de 10×10 píxeles

5.3. Análisis de pruebas estadísticas de los videos en primera y segunda dimensión

Este tipo de análisis suele ser el más ideal para el trabajo de detección en tiempo real debido a que habitualmente observamos las secuencias de imágenes conforme van ocurriendo y no se almacenan en memoria para su posterior análisis.

Su principal desventaja es que la matriz proyector H trabaja con la variación del reflejo del objeto y del mar, características que no pueden ser apreciadas en un solo cuadro de video.

Para esta prueba se vuelve a utilizar únicamente el video 3 con las mismas ventanas de análisis. En las figuras 5.29 a la 5.31 se observa el comportamiento de las gráficas cuando se cambia la ventana de análisis.

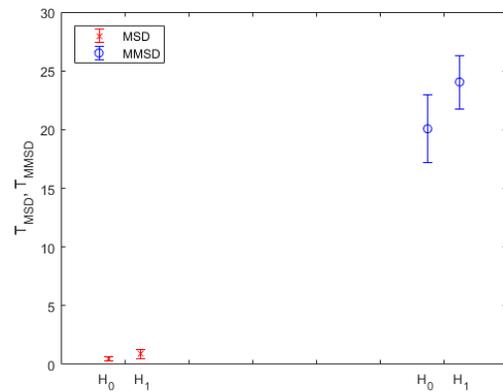
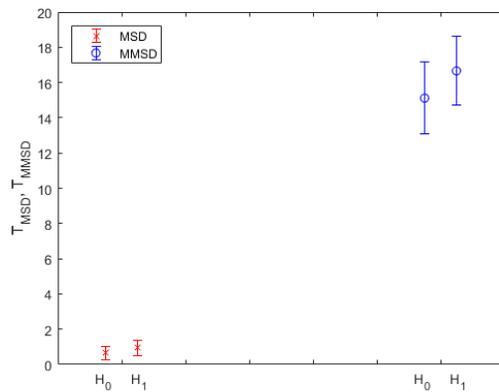


Figura 5.29: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.24

Figura 5.30: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.26

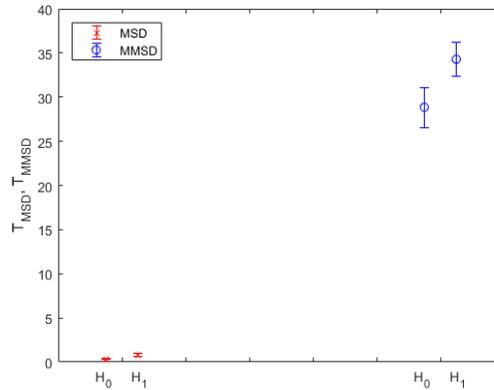


Figura 5.31: Comparación de los algoritmos MSD (izquierda) y MMSD (derecha) para la figura 5.28

5.4. Conclusiones

De las gráficas presentadas en las figuras anteriores, se pueden sacar varias conclusiones:

- El algoritmo MMSD en todo escenario resulta ser superior en desempeño que el algoritmo MSD clásico. Este desempeño se logra apreciar mejor en los escenarios donde el objeto apenas cubre una pequeña parte de la ventana de análisis.
- El algoritmo MSD puede detectar el objeto cuando este ocupa toda la ventana de análisis o la mayor parte de la misma, pero las mínimas diferencias entre su H_0 y su H_1 indican que le es difícil o casi imposible detectar el objeto cuando la ventana crece. Esto es, cuando el objeto es más pequeño que la ventana de análisis.
- Ambos algoritmos son más eficientes cuando el objeto tiende a cambiar de energía más lento que el mar.
- Más iteraciones y más cuadros de video por iteración dan una referencia más precisa de la calidad de detección de ambos algoritmos, sin embargo esta se limita a la cantidad de cuadros que contiene el video. Este fue el motivo principal por el cual se tomó como referencia una cantidad estandarizada de cuadros por estadístico y la cantidad de veces que se repetiría dicha prueba en los cuadros subsecuentes del video.

Capítulo 6

Conclusiones finales

- Este trabajo presenta una problemática que se presenta a menudo en un entorno como el de Ciudad del Carmen: la vigilancia de la superficie marítima y la detección de objetos sobre ella.
- Los métodos de detección basados en sustraer los objetos del fondo, a pesar de ser muy útiles en otros escenarios, en la superficie marítima estos suelen fallar, por lo que se propone cambiar a un algoritmo de detección basado en parámetros estadísticos.
- En este trabajo se analizan dos algoritmos diferentes de detección obtenidos por el método GLRT bajo diferentes condiciones de reflexiones de luz sobre una superficie marítima. Estos algoritmos fueron llamados MSD y MMSD.
- Estos algoritmos fueron puestos a prueba en un entorno simulado a través del software MATLAB. Esto con el objetivo de poder controlar los parámetros espectrales de un fondo y un objeto simulados y observar el comportamiento de dichos algoritmos bajo condiciones que no pueden ser obtenidas en una sola observación real.
- Los resultados de las simulaciones muestran que ambos algoritmos funcionan mejor bajo un fondo blanco que bajo uno correlacionado.
- Los resultados de las mediciones experimentales muestran que, al trabajar en el dominio de la frecuencia, se pueden separar las señales recibidas en dos partes: la primera es aquella donde existe el conjunto de armónicas de las señales que reflejan al objeto, y aquella de las que reflejan el mar. Ambas señales contienen al ruido del canal.
- El análisis de ambos algoritmos muestra que para una superficie marítima no agitada,

ambos algoritmos tienen un desempeño prácticamente igual. Este desempeño decae cuando el mar comienza a agitarse.

- El algoritmo MMSD acumula la energía de la señal útil mejor que la de la señal no útil (esto es, la energía del mar más la del canal). Esto permite al algoritmo MMSD tener un mejor desempeño cuando la energía del objeto es muy pequeña, además se requiere una menor cantidad de cuadros de video para poder obtener un resultado aceptable.
- Los resultados mostrados en las pruebas experimentales concuerdan con lo esperado en el desarrollo de este trabajo, así como lo esperado en las pruebas de simulación.

6.1. Trabajo a futuro

El trabajo a futuro plantea el reto de abandonar la plataforma MATLAB para portar el código realizado a lenguajes más universales como C++. Una de las sugerencias que se plantearon fue el uso de la biblioteca OpenCV, la cual permite trabajar el video como una matriz, muy similar al trabajo de MATLAB. Esto permitirá aplicar el desarrollo presentado anteriormente en plataformas con menos recursos computacionales y donde MATLAB no tiene presencia.

Otro de los retos es aprovechar las conclusiones y algoritmos presentados anteriormente para aplicarse en un proyecto real de monitoreo, ya sea tanto en tiempo real como fuera de línea. Esto va muy acorde con las líneas de investigación de la Maestría en Ingeniería Mecatrónica, ya que uno de los pasos siguientes en el diseño de sistemas autónomos es la toma de decisiones una vez se tiene la información presentada por los sensores.

Existe interés en la Facultad de Ciencias de la Información de la Universidad Autónoma del Carmen sobre aplicar un proyecto similar en colaboración con la Secretaría de Medio Ambiente y Recursos Naturales (SEMARNAT), ante lo cual se requeriría acercamiento con dicha facultad para conocer la problemática que se tiene y cómo se podría colaborar en conjunto.

Referencias

- [1] Félix Aguilar. «Investigación e implementación del algoritmo de detección sub-espacial de objetos en la superficie del mar usando procesamiento de vídeo digital». Universidad Autónoma del Carmen, 2014.
- [2] B. Aldalur y M. Santamaría. «Realce de imágenes: filtrado espacial». En: *Revista de Teledetección* 17 (2002). ISSN: 1133-0953.
- [3] F. Bandiera, D. Orlando y G. Ricci. «CFAR detection of extended and multiple point-like targets without assignment of secondary data». En: *IEEE Signal Processing Letters* 13.4 (abr. de 2006), págs. 240-243. DOI: 10.1109/lsp.2005.863653.
- [4] Andy Bateman. *Comunicaciones digitales : diseño para el mundo real*. Marcombo, S.A., 11 de sep. de 2003. ISBN: 8426713378.
- [5] Al Bovik, ed. *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Academic Press, 2000. ISBN: 0121197905.
- [6] Ajay Kumar Boyat y Brijendra Kumar Joshi. «A Review Paper: Noise Models in Digital Image Processing». En: *Signal & Image Processing : An International Journal* 6.2 (abr. de 2015), págs. 63-75. DOI: 10.5121/sipij.2015.6206.
- [7] Lucía Calderón. *La evolución de la cámara fotográfica en un gráfico*. 2014. URL: <http://www.clasesdeperiodismo.com/2014/06/09/la-evolucion-de-la-camara-fotografica-en-un-grafico/>.
- [8] G. Casal y J. Freire. «Síntesis de la evolución histórica de la teledetección en España (1889-2012)». En: *Revista de Teledetección* 38 (2012), págs. 109-120. ISSN: 1988-8740. URL: http://www.aet.org.es/revistas/revista38/Numero38_10.pdf.
- [9] Emilio Chuvieco. *Fundamentos de teledetección espacial (Colección Monografías y tratados GER) (Spanish Edition)*. Ediciones Rialp, 1990. ISBN: 8432126802.
- [10] Ukranio Coronilla. «Procesamiento digital de video en tiempo real y “video wall” con la PC». Tesis de mtría. Universidad Autónoma Metropolitana, 2005.

- [11] Elena Díaz, ed. *Apuntes Curso Doctorado: Análisis de Imágenes y Reconocimiento de formas*. 2007. URL: <http://informatica.uv.es/doctorado/AIRF/ParteAI/tema2.pdf>.
- [12] Reyna Duarte y Zacarías Malacara. «Evolución permanente». En: *Alternativas, la revista cultural* 45 (ago. de 2016), págs. 10-12.
- [13] Andrés Duque. «Sistema de atención visual para la detección de puntos topológicos de referencia». Tesis de mtría. Sep. de 2009. URL: https://www.researchgate.net/publication/40804360_Sistema_de_atencion_visual_para_la_deteccion_de_puntos_topologicos_de_referencia.
- [14] Iñaki Goiricelaya. «Introducción a la Visión Artificial». En: *DYNA INGENIERIA E INDUSTRIA* 61.9 (dic. de 1986), págs. 29-33. ISSN: 0012-7361. DOI: 10.6036/dynaii. URL: <https://www.revistadyna.com/busqueda/introduccion-a-vision-artificial>.
- [15] Victor Golikov, Olga Lebedeva y Marco Rodriguez Blanco. «GLRT subspace detection of multi-pixel targets with known and unknown spatial parameters in presence of signal-dependent background power». En: *Computers & Electrical Engineering* 52 (mayo de 2016), págs. 38-48. ISSN: 0045-7906. DOI: 10.1016/j.compeleceng.2016.04.016.
- [16] Victor Golikov, Marco Rodriguez-Blanco y Olga Lebedeva. «Robust multipixel matched subspace detection with signal-dependent background power». En: *Journal of Applied Remote Sensing* 10.1 (ene. de 2016), pág. 015006. DOI: 10.1117/1.jrs.10.015006.
- [17] Ana Gonzalez y col. *Técnicas y Algoritmos Básicos de Visión Artificial*. Universidad de la Rioja, Servicio de Publicaciones, 2006. ISBN: 84-689-9345-X.
- [18] William W. Hines. *Probabilidad y Estadística Para Ingeniería y Administración (Spanish Edition)*. Compañía Editorial Continental, 1995. ISBN: 9682612322.
- [19] Planetek Italia. *The History of Remote Sensing*. URL: https://www.planetek.it/eng/training_courses/online_manuals/on_line_course_of_remote_sensing/2_the_history_of_remote_sensing.
- [20] J. M. López. *Vídeo Digital. Tema 3 - Compresión de la Información de Vídeo*. Ed. por Escuela Universitaria Politécnica de Cuenca.
- [21] Maria Lyra, Agapi Ploussi y Antonios Georgantzoglou. «MATLAB as a Tool in Nuclear Medicine Image Processing». En: *MATLAB - A Ubiquitous Tool for the Practical Engineer*. InTech, oct. de 2011. ISBN: 978-953-307-907-3. DOI: 10.5772/19999.
- [22] A. Margalit, I.S. Reed y R.M. Gagliardi. «Adaptive Optical Target Detection Using Correlated Images». En: *IEEE Transactions on Aerospace and Electronic Systems* AES-21.3 (mayo de 1985), págs. 394-405. DOI: 10.1109/taes.1985.310570.

- [23] Julio Martínez. *Percepción Remota - Fundamentos de Teledetección Espacial*. Jun. de 2005. URL: <http://siga.cna.gob.mx/SIGA/Percepcion/Fundamentos%20de%20teledetecci%C3%B3n%20espacial.PDF>.
- [24] Minitab Inc. *¿Qué es un estadístico de prueba?* 2017. URL: <http://support.minitab.com/es-mx/minitab/17/topic-library/basic-statistics-and-graphs/hypothesis-tests/basics/what-is-a-test-statistic/>.
- [25] Michael Murdock. *Image Processing at Ancestry.com – Part 5: Auto-Normalization*. 2013. URL: <https://blogs.ancestry.com/ancestry/2013/05/28/image-processing-at-ancestry-com-part-5-auto-normalization/>.
- [26] Edilberto Nájera, Braly Peralta y Aroldo Pérez. «Lema de Neyman–Pearson para distribuciones de confianza basadas en estadísticas suficientes». En: *Lecturas Matemáticas* 34.2 (2013), págs. 205-223. ISSN: 0120-1980. URL: <http://studylib.es/doc/4901193/lema-de-neyman%E2%80%93pearson-para-distribuciones-de-confianza>.
- [27] Constantino Pérez. *La señal digital de video*. Ed. por Universidad de Cantabria. URL: <http://personales.unican.es/perezvr/pdf/Se%C3%B1al%20digital%20de%20video.pdf>.
- [28] Antonio Ramón Bartolomé Pina. «Vídeo digital - Digital video». En: *Revista Comunicar* 11.21 (2003), págs. 39-47. URL: <https://doi.org/>.
- [29] Carlos Platero. *Apuntes de Visión Artificial*. 2009. URL: <http://www.elai.upm.es/webantigua/spain/Asignaturas/Robotica/InfoRobotica.htm>.
- [30] Marco Rodríguez-Blanco y Victor Golikov. «Multiframe GLRT-Based Adaptive Detection of Multipixel Targets on a Sea Surface». En: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.12 (dic. de 2016), págs. 5506-5512. DOI: 10.1109/jstars.2016.2582383.
- [31] Rogelio Rojas, Ramón Silva y Aurora Molina. «La Visión Artificial en la Robótica». En: *Polibits* 35 (ene. de 2007), págs. 22-28. DOI: 10.17562/pb.
- [32] F. A. Sarría. *Teledetección*. 2003. URL: <http://www.um.es/geograf/sigmur/teledet/tema06.pdf>.
- [33] F. A. Sarría. *Sistemas de Información Geográfica*. 2006. URL: <http://www.um.es/geograf/sigmur/sigpdf/temario.pdf> (visitado 14-08-2018).
- [34] L. Scharf y B. Friedlander. «Matched subspace detectors». En: *IEEE Transactions on Signal Processing* 42.8 (1994), págs. 2146-2157. DOI: 10.1109/78.301849.

- [35] Gonzalo Silva y col. «Desarrollo de un Sistema de Detección de Movimiento basado en Flujo Óptico en Raspberry Pi». En: *La Mecatrónica en México* 4.2 (mayo de 2015), págs. 65-76. ISSN: 2448-7031. URL: <http://www.mecamex.net/revistas/LMEM/>.
- [36] Birgi Tamersoy. *Background Substraction*. 2009. URL: https://www.cs.utexas.edu/~grauman/courses/fall2009/slides/lecture9_background.pdf.
- [37] Ronald E. Walpole y col. *Probabilidad y estadística para ingeniería y ciencias*. 9th. Pearson Educación, 2012. ISBN: 6073214170.

Anexos

Análisis en tres dimensiones de video

```
1  %%Video de prueba 5
2  %%prueba1.avi
3  %%Tres dimensiones se transforma a dos
4
5  %%Limpiamos las variables
6  close all
7  clearvars
8  clc
9
10 %%Leemos el video
11 objeto = VideoReader('C:\Users\dkant\Documents\MATLAB\videoselegidos\MVI_0553_2
    .avi');
12 %%Obtenemos informacion del video
13 fps = objeto.FrameRate;
14 duracion = objeto.NumberOfFrames;
15 ancho = objeto.Width;
16 alto = objeto.Height;
17
18 %%Ventana de calculo de umbral
19 u2 = 51; %Coordenadas en y
20 u1 = 49; %Coordenadas en x
21 %%Ponemos el inicio y fin de los frames
22 inicio = 1;
23 final = 40;
24 repeticiones = 40;
25 ventana = 5;
```

```

26 %%Se obtiene el primer frame
27
28     frame = read(objeto,1);
29     temp = rgb2gray(frame);
30
31 %%Se crea la matriz proyector
32 p=3;
33 dur = final-inicio+1;
34 K=dur;
35 renglones=0:(K-1);
36 for columnas=1:p
37     h(:,columnas)=(1/sqrt(K))*exp((-1i*2*pi*renglones*(columnas-1)/K));
38 end
39 B=inv(h'*h);
40 PH=h*B*h';
41 PH0=eye(K)-PH;
42 %%Descomentar para elegir una ventana de analisis manual
43 % figure('Name','Measured Data');imshow(fondo(:,:,1),'InitialMagnification','
    fit');
44 % rectangle('Position',[u1,u2,ventana,ventana],'LineWidth',2,'EdgeColor','y');%
    Ventana de analisis
45 %%Elegir ventana de analisis manualmente
46 py=53;
47 px=120;
48 % py=int32(py);
49 % px=int32(px);
50 % close
51 f=[py,px];
52 figure('Name','Ventanas');imshow(temp,'InitialMagnification','fit');
53 a1=abs(f(1));
54 a2=abs(f(2));
55 rectangle('Position',[a2,a1,ventana,ventana],'LineWidth',2,'EdgeColor','b');%
    Ventana de analisis
56 rectangle('Position',[u1,u2,ventana,ventana],'LineWidth',2,'FaceColor','r');%
    ventana para calcular umbral
57
58 %%Media
59 for j=0:repeticiones-1

```

```

60     for i=inicio:final
61         frame = read(objeto,i+(dur*j));
62         temp = rgb2gray(frame);
63         fondo1(:,:,i) = double(temp(:,1:ancho));
64         Mf(:,:,i)=fondo1(u2:(u2+ventana-1),u1:(u1+ventana-1),i);
65         Mf0(i,:)=reshape(Mf(:,:,i),1,numel(Mf(:,:,i)));
66     end
67     mediamar(j+1) = mean(mean(Mf0,2),1);
68 end
69 mediamar2 = mean(mediamar);
70
71 %%Trace
72 for j=0:repeticiones-1
73     for i=inicio:final
74         frame = read(objeto,i+(dur*j));
75         temp = rgb2gray(frame);
76         fondo1(:,:,i-inicio+1) = double(temp(:,1:ancho))/1;
77         Mf(:,:,i-inicio+1)=fondo1(u2:(u2+ventana-1),u1:(u1+ventana-1),i-inicio
78             +1)-mediamar2; %%para h0
79         Mf0(i,:)=reshape(Mf(:,:,i),1,numel(Mf(:,:,i)));
80         M(:,:,i)=fondo1(py:(py+ventana-1),px:(px+ventana-1),i)-mediamar2; %%para
81             h1
82         M0(i,:)=reshape(M(:,:,i),1,numel(M(:,:,i)));
83     end
84     mediatrazaH0(j+1) = trace(real(Mf0'*Mf0));
85     mediatrazaH1(j+1) = trace(real(M0'*M0));
86 end
87 mediatraza2 = mean(mediatrazaH0);
88 mediatraza3 = mean(mediatrazaH1);
89 clear Mf, clear Mf0, clear M, clear M0
90
91 %%H0
92 for j=0:repeticiones-1
93     for i=inicio:final
94         frame = read(objeto,i+(dur*j));
95         temp = rgb2gray(frame);
96         fondo1(:,:,i) = double(temp(:,1:ancho));

```

```

96     Mf(:, :, i) = fondo1(u2:(u2+ventana-1), u1:(u1+ventana-1), i) - mediamar2;
97     Mf0(i, :) = reshape(Mf(:, :, i), 1, numel(Mf(:, :, i)));
98     end
99     Mf1(j+1) = trace(abs(Mf0' * PH * Mf0 / mediatraza2));
100    MMf1(j+1) = trace(abs(Mf0' * PH * Mf0 / mediatraza2)) - trace(log(abs(Mf0' * PH0 * Mf0 /
    mediatraza2)));
101    if j == 0
102        Mfprimero = Mf(:);
103    end
104
105 end
106 %% H1
107
108 for j = 0:repeticiones-1
109     for i = inicio:final
110         frame = read(objeto, i + (dur * j));
111         temp = rgb2gray(frame);
112         fondo2(:, :, i) = double(temp(:, 1:ancho));
113         M(:, :, i) = fondo2(py:(py+ventana-1), px:(px+ventana-1), i) - mediamar2;
114         M0(i, :) = reshape(M(:, :, i), 1, numel(M(:, :, i)));
115     end
116     M1(j+1) = trace(abs(M0' * PH * M0 / mediatraza2));
117     MM1(j+1) = trace(abs(M0' * PH * M0 / mediatraza2)) - trace(log(abs(M0' * PH0 * M0 /
    mediatraza3)));
118
119     if j == 0
120         Mprimero = M(:);
121     end
122 end
123
124 Y1 = [Mf1', M1'];
125 y = mean(Y1, 1);
126 e = std(Y1, 1, 1);
127 P = [0.95, 1.05];
128 figure;
129 errorbar(P, y, e, 'xr');
130 hold on;
131 Y2 = [MMf1', MM1'];

```

```

132 y = mean(Y2,1);
133 e = std(Y2,1,1);
134 P=[1.95,2.05];
135 errorbar(P,y,e,'ob');
136 hold on;
137 legend('MSD', 'MMSD', 'Location', 'northwest');
138 set(gca, 'XtickLabel', {' ' 'H_{0}' ' ' ' ' ' ' ' 'H_{0}' ' ' 'H_{1}' ' ' '});
139 ylabel('T_{MSD}, T_{MMSD}');

```

Análisis en primera y segunda dimensión en video

```

1  %%Video de prueba 5
2  %%prueba1.avi
3  %%Dos dimensiones nativo
4
5  %%Limpiamos las variables
6  close all
7  clearvars
8  clc
9
10 %%Leemos el video
11 objeto = VideoReader('C:\Users\dkant\Documents\MATLAB\videoselegidos\MVI_0553_2
    .avi');
12 %%Obtenemos informacion del video
13 fps = objeto.FrameRate;
14 duracion = objeto.NumberOfFrames;
15 ancho = objeto.Width;
16 alto = objeto.Height;
17
18 %%Ventana de calculo de umbral
19 u2 = 51; %Coordenadas en y
20 u1 = 49; %Coordenadas en x
21 %%Ponemos el inicio y fin de los frames
22 inicio = 1;
23 final = 20;
24 repeticiones = 20;
25 ventana = 10;

```

```

26 %%Se obtiene el primer frame
27     frame = read(objeto,1);
28     temp = rgb2gray(frame);
29 %%Se crea la matriz proyector
30 p=3;
31 K = ventana;
32 dur = final-inicio+1;
33 renglones=0:(K-1);
34 for columnas=1:p
35     h(:,columnas)=(1/sqrt(K))*exp((-1i*2*pi*renglones*(columnas-1)/K));
36 end
37 B=inv(h'*h);
38 PH=h*B*h';
39 PH0=eye(K)-PH;
40 %%Descomentar para elegir una ventana de analisis manual
41 % figure('Name','Measured Data');imshow(fondo(:,:,1),'InitialMagnification','
    fit');
42 % rectangle('Position',[u1,u2,ventana,ventana],'LineWidth',2,'EdgeColor','y');%
    Ventana de analisis
43 %%Elegir ventana de analisis manualmente
44 py=53;
45 px=120;
46 % [px py]=ginput(1);
47 % py=int32(py);
48 % px=int32(px);
49 % close
50 f=[py,px];
51 figure('Name','Ventanas');imshow(temp,'InitialMagnification','fit');
52 a1=abs(f(1));
53 a2=abs(f(2));
54 rectangle('Position',[a2,a1,ventana,ventana],'LineWidth',2,'EdgeColor','b');%
    Ventana de analisis
55 rectangle('Position',[u1,u2,ventana,ventana],'LineWidth',2,'FaceColor','r');%
    ventana para calcular umbral
56
57 %%Media
58 for j=0:repeticiones-1
59     for i=inicio:final

```

```

60     frame = read(objeto,i+(dur*j));
61     temp = rgb2gray(frame);
62     fondo1(:,:,i-inicio+1) = double(temp(:,1:ancho))/1;
63     Mf(:,:,i-inicio+1)=fondo1(u2:(u2+ventana-1),u1:(u1+ventana-1),i-inicio
        +1);
64     end
65     mediamar(j+1) = mean(mean(mean(Mf,3),2),1);
66 end
67 mediamar2 = mean(mediamar);
68
69 %%Trace
70 for j=0:repeticiones-1
71     for i=inicio:final
72         frame = read(objeto,i+(dur*j));
73         temp = rgb2gray(frame);
74         fondo1(:,:,i-inicio+1) = double(temp(:,1:ancho))/1;
75         Mf(:,:,i-inicio+1)=fondo1(u2:(u2+ventana-1),u1:(u1+ventana-1),i-inicio
            +1)-mediamar2; % Para H0
76         M(:,:,i-inicio+1)=fondo1(py:(py+ventana-1),px:(px+ventana-1),i-inicio
            +1)-mediamar2;%para h1
77         trazaH0(i) = trace(real(Mf(:,:,i))*Mf(:,:,i));
78         trazaH1(i) = trace(real(M(:,:,i))*M(:,:,i));
79     end
80     mediatrazaH0(j+1) = mean(trazaH0);
81     mediatrazaH1(j+1) = mean(trazaH1);
82 end
83 mediatraza2 = mean(mediatrazaH0);
84 mediatraza3 = mean(mediatrazaH1);
85 clear Mf, clear Mf0, clear M, clear M0
86
87 %%H0
88 for j=0:repeticiones-1
89     for i=inicio:final
90         frame = read(objeto,i+(dur*j));
91         temp = rgb2gray(frame);
92         fondo1(:,:,i-inicio+1) = double(temp(:,1:ancho))/1;
93         Mf(:,:,i-inicio+1)=fondo1(u2:(u2+ventana-1),u1:(u1+ventana-1),i-inicio
            +1)-mediamar2;

```

```

94     Mf1(i)=trace(abs(Mf(:,:,i))*PH*Mf(:,:,i)/mediatrazas2));
95     MMf1(i)=trace(abs(Mf(:,:,i))*PH*Mf(:,:,i)/mediatrazas2)-trace(log(abs(
        Mf(:,:,i))*PH0*Mf(:,:,i)/mediatrazas2)));
96     end
97     sumaMf1(j+1) = mean(Mf1);
98     sumaMMf1(j+1) = mean(MMf1);
99 end
100 %%H1
101
102 for j=0:repeticiones-1
103     for i=inicio:final
104         frame = read(objeto,i+(dur*j));
105         temp = rgb2gray(frame);
106         fondo2(:,:,i-inicio+1) = double(temp(:,1:ancho))/1;
107         M(:,:,i-inicio+1)=fondo2(py:(py+ventana-1),px:(px+ventana-1),i-inicio
            +1)-mediamar2;
108         M1(i)=trace(abs(M(:,:,i))*PH*M(:,:,i)/mediatrazas2));
109         MM1(i)=trace(abs(M(:,:,i))*PH*M(:,:,i)/mediatrazas2)-trace(log(abs(M
            (:,:,i))*PH0*M(:,:,i)/mediatrazas3)));
110     end
111     sumaM1(j+1) = mean(M1);
112     sumaMM1(j+1) = mean(MM1);
113
114 end
115 Y1 = [sumaMf1',sumaM1'];
116 y = mean(Y1,1);
117 e = std(Y1,1,1);
118 P=[0.95,1.05];
119 figure;
120 errorbar(P,y,e,'xr');
121 hold on;
122 Y2 = [sumaMMf1',sumaMM1'];
123 y = mean(Y2,1);
124 e = std(Y2,1,1);
125 P=[1.95,2.05];
126 errorbar(P,y,e,'ob');
127 hold on;
128 legend('MSD','MMSD','Location','northwest');

```

```
129 set(gca,'XtickLabel',{'H_{0} H_{1}' 'H_{0} H_{1}'});
130 ylabel('T_{MSD}, T_{MMSD}');
```

Robust Detector of Multi-Pixel Targets Using a Sequence of Images

Victor Golikov^{*}, David Kantun Marin, HussainAlazki

Universidad Autonoma del Carmen, calle 56 esq. av. Concordia, No.4, Facultad de Ingenieria, Ciudad del Carmen, Camp., Mexico. C.P. 24180
Corresponding Author: Victor Golikov

Abstract : We design and assess an algorithm to detect a multi-pixel target of an unknown spatial size, shape and position in a sequence of images in the presence of an additive Gaussian background clutter and a channel noise. The presence of the target decreases the background plus noise power that hence may be different under the null and alternative hypotheses. We use the generalized likelihood ratio (GLR) approach to derive a modified multi-pixel matched subspace detector (MMMSD) that is sensitive to both energy in the target subspace and reduced energy in the orthogonal subspace. The derived algorithm combines the multi-pixel matched subspace detector and multi-pixel background-plus-noise power change detector in a unique scheme. The crucial characteristic of the proposed detector is that prior knowledge of the target size, shape and position is not required. The designed detector is theoretically proved and numerically evaluated. Numerical simulations attest the validity of the theoretical analysis and show that the proposed MMMSD outperforms the known detector in the case of unknown spatial parameters of the target.

Keywords: *Multi-pixel subspace detection, sequence of optical digital images, signal-dependent background clutter.*

Date of Submission: 11-01-2018

Date of acceptance: 27-01-2018

I. INTRODUCTION

Target detection from optical/infrared images has been considered by many authors [1-25]. A number of approaches has been previously developed for the detection of targets in the presence of the dominant background clutter and noise. The GLR approach is a well known tool among the signal processing community and has been exploited in many detection problems [13], [14]. Reed and Yu [22] considered GLR target detection from a sequence of optical images, which are first preprocessed by removing local means so that the background clutter and noise will approximately have the Gaussian distribution. Distributed target detectors in Gaussian and Compound-Gaussian noise have been developed in [15-19]. In [21, 22, 25], the detector uses one pixel in an image sequence even though the target may occupy more than one pixel. It was shown [21] that the performance of the multi-pixel detector outperforms the performance of the detector using only one pixel. All quoted above detectors exhibit one drawback: they generally fail when the signal-to-background ratio (SBR) is low. The introduction of the paper should explain the nature of the problem, previous work, purpose, and the contribution of the paper. The contents of each section may be provided to understand easily about the paper.

In video/infrared systems, the target may completely cover the pixel cells on the fluctuating surface and, in this case, the received signal contains only target signal plus channel noise. Hence, the presence of the target removes the background clutter from the received signal. In this case, it is more appropriate to use the GLR approach with different background plus noise power under the two hypotheses. Specifically, each pixel contains the background-plus-noise power under the null hypothesis and the signal-plus-noise power under the alternative hypothesis only in the case of the presence of the target in this pixel. A modified GLR approach associated with the hypothesis dependent background clutter power has been recently proposed by us for subpixel optical/infrared objects [24, 25]. In our work, the detection problem of multi-pixel targets is being solved using the GLR approach that processes a certain set of pixels (subimage) in a sequence of images. We find the GLR test (GLRT) for a partially known deterministic multi-pixel target signal by using a set of K subimages of the same scene obtained from sequential observation in time. We extend the GLR approach to a general case when the possible object is contained within unknown N pixels located in the subimage of size L . The object may be partially or completely present within a received $K \times L$ subimage data matrix. Among the L pixels, N ($N \leq L$) pixels are a sum of the deterministic signal and Gaussian channel noise, while the remaining $L - N$ pixels are a sum of a Gaussian background and Gaussian channel noise. The received subimage data matrix is used to estimate the sample background power for each pixel only under the alternative hypothesis. The proposed detector (MMMSD) is sensitive to both the SBR and size of the area unoccupied by the target within

Trabajos enviados

- David Kantun, Victor Golikov, Hussain Alazki. *Multipixel Optical Detection of the Weak Floating Objects on an Agitated Sea Surface*. Trabajo enviado a IEEE Latin America Transactions, en espera de respuesta.



UNACAR

Universidad Autónoma del Carmen
"Por la Grandeza de México"

ACUERDO PARA USO DE OBRA

A quien corresponda

PRESENTE

Por medio del presente escrito, **David Emmanuel Kantún Marín** (en lo sucesivo EL AUTOR) hace constar que es titular intelectual de la obra denominada, "**Desarrollo del Sistema de Detección de los Objetos Flotantes sobre Superficie Marítima usando Visión Artificial**" (en lo sucesivo la OBRA), en virtud de lo cual autoriza a la Universidad Autónoma del Carmen (en lo sucesivo la UNACAR) para que efectúe resguardo físico y/o electrónico mediante copia digital o impresa para asegurar su disponibilidad, divulgación, comunicación pública, distribución, transmisión, reproducción, así como digitalización de la misma con fines académicos y sin fines de lucro como parte de Repositorio Institucional de la UNACAR (*Runacar*).

De igual manera, es deseo del AUTOR establecer que esta autorización es voluntaria y gratuita, y que de acuerdo a lo señalado en la Ley Federal del Derecho de Autor y la Ley de Propiedad Industrial, la UNACAR cuenta con mi autorización para la utilización de la información antes señalada, estableciendo que se utilizará única y exclusivamente para los fines antes señalados. EL AUTOR autoriza a la UNACAR a utilizar las obras en los términos y condiciones aquí expresados, sin que ello implique que se le conceda licencia o autorización alguna o algún tipo de derecho distinto al mencionado respecto a la "propiedad intelectual" de la misma OBRA; incluyendo todo tipo de derechos patrimoniales sobre obras y creaciones protegidas por derechos de autor y demás formas de propiedad intelectual reconocida o que lleguen a reconocer las leyes correspondientes. Al reutilizar, reproducir, transmitir y/o distribuir la OBRA se deberá reconocer y dar crédito de autoría de la obra intelectual en los términos especificados por el propio AUTOR, y el no hacerlo implica el término de uso de esta licencia para los fines estipulados. Nada de esta licencia menoscaba o restringe los derechos patrimoniales y morales del AUTOR.

De la misma manera, se hace manifiesto que el contenido académico, literario, la edición y en general de cualquier parte de la OBRA son responsabilidad del AUTOR, por lo que se deslinda a la UNACAR por cualquier violación a los derechos de autor y/o propiedad intelectual, así como cualquier responsabilidad relacionada con la misma frente a terceros. Finalmente, el AUTOR manifiesta que estará depositando la versión final de su Tesis de maestría, OBRA y cuenta con los derechos morales y patrimoniales correspondientes para otorgar la presente autorización de uso.

En la ciudad de Carmen, del estado de Campeche a los 11 días el mes de octubre de 2018.

Atentamente,


DAVID EMMANUEL KANTÚN MARÍN

Nombre y Firma Autógrafa de EL AUTOR

Escriba la Facultad, Escuela, Centro a la que está suscrita la obra: Facultad de Ingeniería, Universidad Autónoma del Carmen, Campus III.