



# APLICACIÓN DE TÉCNICAS DE LOCALIZACIÓN DE OBJETIVOS PARA SEGUIMIENTO VISUAL

**TRABAJO DE MASTER**

MASTER EN AUTOMÁTICA, ROBÓTICA Y TELEMÁTICA

E.T.S. de INGENIEROS de la UNIVERSIDAD DE SEVILLA

Autor: Eloy Sánchez de Gregorio

Titulación: Ingeniero Industrial

Sevilla, Noviembre de 2007

eloy.sanchez@tecnobit.net



## **INDICE DE CONTENIDOS**

<b>LISTA DE FIGURAS</b>	<b>V</b>
<b>RESUMEN</b>	<b>VII</b>
<b>1. ESTADO DEL ARTE</b>	<b>1-9</b>
<b>2. ALGORITMOS DE ESPECIAL INTERÉS</b>	<b>2-12</b>
<b>3. DETECCIÓN DE MOVIMIENTO CON CÁMARAS FIJAS</b>	<b>3-13</b>
<b>3.1 INTRODUCCIÓN.....</b>	<b>3-13</b>
<b>3.2 ALGORITMO DE DIFERENCIACIÓN .....</b>	<b>3-14</b>
3.2.1 Diferenciación Básica .....	3-14
3.2.2 Generación del Background.....	3-15
3.2.3 Umbral Automático de Diferenciación de Imágenes .....	3-15
3.2.4 Implementación de Algoritmos de Diferenciación.....	3-17
<b>3.3 RESULTADOS EXPERIMENTALES .....</b>	<b>3-25</b>
3.3.1 Vídeo 1 .....	3-25
3.3.2 Vídeo 2.....	3-26
3.3.3 Vídeo 3.....	3-27
3.3.4 Vídeo 4.....	3-28
3.3.5 Vídeo 5.....	3-29
<b>4. ODOMETRÍA INERCIAL</b>	<b>4-31</b>
<b>4.1 INTRODUCCIÓN.....</b>	<b>4-31</b>
<b>4.2 Fundamentos Teóricos.....</b>	<b>4-33</b>
4.2.1 Operador Matricial.....	4-33
4.2.2 Integración de Velocidad Inercial.....	4-34
4.2.3 Matriz Jacobiana .....	4-34
<b><u>4.3 Errores de Integración.....</u></b>	<b><u>4-37</u></b>

4.3.1 Errores de Redondeo .....	4-37
<b>4.4 DATOS EXPERIMENTALES DE INTEGRACIÓN INERCIAL.....</b>	<b>4-41</b>
<b>4.5 GENERADOR DE TRAYECTORIAS.....</b>	<b>4-45</b>
4.5.1 Fundamentos Teóricos.....	4-45
4.5.2 Pre-filtro de velocidad y aceleración.....	4-47
4.5.3 Simulaciones.....	4-48
<b>5. ESTABILIZACIÓN DE IMAGEN MEDIANTE FUSIÓN DE SENSORES</b>	<b>5-52</b>
<b>6. RESULTADOS EXPERIMENTALES DE IMÁGENES EN MOVIMIENTO</b>	<b>6-53</b>
6.1 Estabilización de la Imagen.....	6-53
6.2 Detección de objetos basada en diferenciación .....	6-55
6.3 Detección de objetos basados en algoritmo de background.....	6-56
<b>CONCLUSIONES</b>	<b>6-57</b>
<b>ANEXOS</b>	<b>6-58</b>
<b>BIBLIOGRAFÍA</b>	<b>59</b>

## **LISTA DE FIGURAS**

Figura 3-1: Algoritmo de Diferenciación básico (umbral=20).....	3-17
Figura 3-2: Algoritmo de Diferenciación básico (umbral=10).....	3-18
Figura 3-3: Paso 3: Dilatación o Crecimiento.....	3-19
Figura 3-4: Paso 4: Detección de contornos.....	3-19
Figura 3-5: Paso 6: Resultados del algoritmo de diferenciación .....	3-20
Figura 3-6: Ruido del Algoritmo de Diferenciación .....	3-21
Figura 3-7: Imagen inicial del background .....	3-22
Figura 3-8: Background Actualizado .....	3-23
Figura 3-9: Diferenciación con el fondo .....	3-24
Figura 3-10: Resultados vídeo 1.....	3-25
Figura 3-11: Resultados vídeo 2.....	3-26
Figura 3-12: Resultados vídeo 3.....	3-27
Figura 3-13: Resultados vídeo 4.....	3-28
Figura 3-14: Resultados vídeo 5.....	3-29
Figura 4-1: Triedro Inercial y sistema de referencia móvil.....	4-31
Figura 4-2: Ángulos de orientación del sistema de referencia móvil .....	4-32
Figura 4-3: Variación de la deriva frente a la frecuencia de muestreo .....	4-40
Figura 4-4: Velocidades Inerciales del giróscopo .....	4-41
Figura 4-5: Coordenadas articulares Vs Coordenadas articulares Integradas. ....	4-43
Figura 4-6: Errores del algoritmo de Integración. ....	4-44
Figura 4-7: Direct-Drive de actuación en ejes Y, Z.....	4-45
Figura 4-8: Representación eje de giro.....	4-46
Figura 4-9: Perfil Trapezoidal de velocidades .....	4-47

Figura 4-10: Movimiento Inercial $X_{REF\_U}^T = [0 \ 0 \ 1]$ con periodo de muestreo 0.56seg. ....	4-48
Figura 4-11: Componentes del Eje de Rotación .....	4-49
Figura 4-12: Velocidad Angular Proyectada en el plano del movimiento .....	4-50
Figura 6-1: Video Capturado: video_a .....	6-53
Figura 6-2: Velocidad Inercial de Orientación de la Cámara .....	6-53
Figura 6-3: Velocidad Inercial Integrada por Odometría.....	6-54
Figura 6-3: Velocidad Inercial sincronizada .....	6-54
Figura 6-3: Detección basada en diferenciación con imágenes en movimiento .....	6-55
Figura 6-6: Detección basada en el background con imágenes en movimiento .....	6-56
Figura 6-7: Background generado .....	6-56

### RESUMEN

El trabajo consiste en el desarrollo de algoritmos que permiten detectar objetos en una secuencia de imágenes en movimiento. El principal propósito es alcanzar dicha finalidad cumpliendo con algunas limitaciones de capacidad de procesado. Otra parte importante abordada durante el trabajo es la integración multitensorial, para conseguir los objetivos propuestos.

En el primer capítulo del trabajo se enumerarán algunos trabajos y artículos publicados y se comentarán las ventajas e inconvenientes de los métodos expuestos.

En el segundo capítulo se desarrollan los algoritmos de detección de movimiento, utilizando imágenes obtenidas con cámaras estáticas.

El tipo de imágenes utilizadas, son imágenes infrarrojas que poseen ciertas ventajas frente a las cámaras en el espectro visible. En esta franja de radiación infrarroja, los objetos de interés resaltan fácilmente sobre el fondo y no producen sombras.

Como aplicaciones en las que son útiles estos algoritmos de detección, podemos destacar vigilancia, alarmas, detección de vehículos, seguridad, etc...

En cuanto a la estructura de este capítulo, inicialmente se muestra una visión global de algunos algoritmos genéricos de detección. Seguidamente, se detallan algoritmos de diferenciación y su implementación en Matlab. Se muestran algunos resultados experimentales sobre vídeos reales grabados. Y para finalizar, se enumeran las conclusiones obtenidas sobre los algoritmos de detección basados en cámaras estáticas.

En el capítulo 4, se detallan las técnicas utilizadas en el ámbito de orientación espacial. La odometría se utilizan en diferentes ámbitos de la robótica, en este capítulo, se aplicarán dichas técnicas de integración, a las velocidades inerciales en un cuerpo rígido que puede rotar libremente en el espacio.

Un problema muy simple de odometría, como puede ser la integración de una variable en un solo grado de libertad, puede complicarse desde el punto de vista computacional y de implementación en tiempo real, cuando se aborda la integración en tres dimensiones angulares.

Para comenzar, se realizará una estimación de la orientación en el espacio, de un sistema móvil dotado de tres sensores de velocidad angular inercial, respecto de un triedro de referencia solidario con el universo.

La intención de este estudio de investigación, no es solamente ofrecer una solución al

problema planteado, sino alcanzar una solución con ventajas computacionales como para poder implementarla en un sistema de tiempo real con ciertas limitaciones de procesado (por ejemplo, un DSP industrial).

Otro aspecto importante estudiado en este trabajo, es el error en la orientación inercial o deriva. En esta línea, se analizará como varía dicha deriva con el método de integración utilizado, con la precisión de las variables de cálculo y con el periodo de muestreo (problema de optimización con restricciones para el cual ofrece una solución).

Además del problema de odometría en 3-D, la segunda línea desarrollada, es la estrategia de control para apuntar el sistema de referencia móvil a una dirección inercial de referencia. Se exigirá que el algoritmo optimice la trayectoria 3D, así como el tiempo en alcanzar la dirección objetivo. Se han incluido limitaciones físicas de velocidad y aceleración, para crear la necesidad de diseñar un filtro en la señal de referencia.

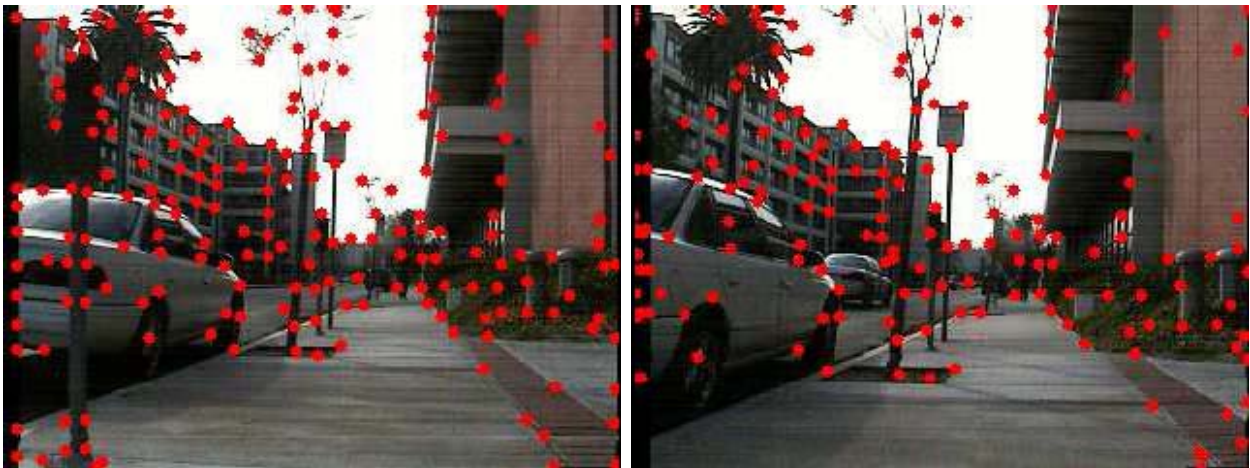
El capítulo 5, está dedicado a la estabilización de la imagen basada en la integración de la imagen. En este capítulo se explicarán los métodos utilizados para conseguir imágenes estables, con el objetivo de facilitar la aplicación de los algoritmos de localización de objetivos.

En el capítulo 6 se muestran algunas secuencias de video, como resultado de los algoritmos desarrollados y posteriormente se enumeran las conclusiones y la bibliografía utilizada.



### 1. ESTADO DEL ARTE

La mayoría de la literatura referida a la localización de objetivos utilizando secuencias de video en movimiento, basan sus algoritmos en reconocimiento de ciertas características de la imagen. Básicamente, dichos algoritmos trazan detalles de un frame, a las secuencias posteriores para obtener información del movimiento del fondo y del movimiento de los objetos incluidos en ella. Un ejemplo de este tipo de algoritmos fue publicado por Boyoon Jung con el título "Detecting Moving Objects using a Single Camera on a Mobile Robot in an Outdoor Environment"



Si analizamos el algoritmo desarrollado en el artículo, y con algo de experiencia en procesamiento de imágenes, se pueden obtener dos carencias importantes. La primera de ellas referente al tiempo de ejecución de dicho método y la cantidad de cálculos necesarios para desarrollar este tipo de algoritmos, restringe la utilización a procesamiento de tratamiento de video pero costoso si quisiéramos integrarlo en una aplicación de tiempo real. El segundo de los inconvenientes, es la fiabilidad a la hora de utilizar estos algoritmos en aplicaciones reales. Un algoritmo basado exclusivamente en la información de la imagen se puede confundir con facilidad.

La principal problemática para los algoritmos de detección de movimiento en

secuencias de imágenes consiste en discriminar entre el movimiento de los objetos filmados y el propio movimiento de la cámara.

Ante la problemática expuesta en el párrafo anterior, se extiende otra línea paralela que aborda la estabilización de la imagen. Hoy en día, el uso de estabilizadores de imagen está muy extendido en cámaras, videocámaras, telescopios y binoculares. Los más comunes son:

- Estabilizador de Imagen óptico: normalmente es un sistema mecánico aparte de la cámara que incorpora dos superficies ópticas flotantes paralelas al interior del lente que actúan como un tipo de prisma flexible. Cuando la cámara se mueve, el movimiento es electrónicamente detectado y se genera un voltaje que hace mover las lentes. Esto altera el ángulo de la luz que atraviesa el prisma y envía la imagen al sensor en la dirección opuesta al movimiento que realiza la cámara. Por tanto, estabiliza la imagen antes de ser procesada. Ya que la imagen completa del sensor es usada con la estabilización de imagen óptica, no se obtiene pérdida de calidad de la misma.

- Estabilizador de Imagen Digital: Es un sistema electrónico que actúa directamente sobre la imagen obtenida en el sensor CCD de la cámara. En este tipo de sistemas, la superficie de la imagen útil es ligeramente menor que la superficie de la imagen. Cuando la cámara se mueve, el encuadre menor se desplaza entre el área mayor del sensor CCD tratando de compensar el movimiento. El sistema electrónico se encarga de determinar el índice de la fila y la columna de la imagen capturada por el sensor que debe presentarse como primera fila y primera columna de la imagen útil. Si por ejemplo, la cámara se mueve ligeramente a un lado, el encuadre digital se moverá en la dirección opuesta, cancelando el efecto del movimiento en el sensor de la cámara. Aunque la técnica funciona efectivamente cancelando movimientos limitados de cámara, al reducir el área aprovechable del sensor, se sacrifican la resolución y claridad de la imagen.

Aunque los dispositivos de estabilización son de gran ayuda para reducir o eliminar movimientos de cámara no deseados, hay que tener en cuenta algunas consideraciones a la hora de realizar tomas en movimiento. Así, debido a los estabilizadores de imagen, cuando la cámara se desplaza intencionadamente de un lado a otro, existe normalmente una pequeña demora mientras la cámara trata inicialmente de compensar el movimiento. Una vez rebasado cierto punto, el estabilizador no puede compensar más el movimiento y la imagen comienza a moverse como se requería. De igual manera, al final del recorrido, la imagen continuará moviéndose durante un instante mientras el sistema vuelve a estabilizarse. Este hecho implica, que a la hora de realizar un

desplazamiento con la cámara, se deberá terminar un determinado instante antes y permitir que la cámara complete el movimiento.

## **2. ALGORITMOS DE ESPECIAL INTERÉS**

De los algoritmos más interesantes sobre la estabilización de imagen utilizando procesado de video, destaca un artículo del autor Zhigang Zhu titulado "Digital Video Sequence Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering"

Este algoritmo consigue una estabilización de la imagen frente a movimientos de la cámara de rotación y translación, utilizando un filtro de movimiento inercial.

### **ARTÍCULO: Digital Video Sequence Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering**

This paper presents a novel approach to stabilize digital video sequences. A 2.5D inter-frame motion model is proposed so that the stabilization system can perform in the situations where significant depth changes are present and the camera has both rotation and translation. Inertial motion filtering is proposed in order to eliminate the vibration of the video sequences with enhanced perceptual properties. The implementation of this new approach integrates four modules: pyramid-based motion detection, 2.5D motion parameter estimation, inertial motion filtering and affine-based motion compensation. The stabilization system can smooth unwanted vibrations or shakes of video sequences and achieves a real-time speed.

### 3. DETECCIÓN DE MOVIMIENTO CON CÁMARAS FIJAS

#### 3.1 INTRODUCCIÓN

Una de las formas más sencillas para detectar movimiento es utilizar un algoritmo de diferenciación. Este consiste en comparar dos imágenes consecutivas y detectar qué objetos se mueven. Este es un método útil y eficiente para cámaras estáticas, en cambio, cuando tenemos cámaras móviles, el algoritmo es incapaz de diferenciar entre el movimiento producido por la cámara y el movimiento de los objetos.

De la bibliografía sobre detección de movimiento, destacar principalmente dos artículos:

- El primero de ellos, referencia [1] de la bibliografía, está basado en la detección de movimiento de objetos, utilizando una cámara estática. Básicamente, el algoritmo se puede dividir en dos partes principales: La primera de ellas consiste en extraer el fondo de la imagen y separarla de los objetos en movimiento. El obtener los objetos en movimiento se reduce a una diferenciación entre el "background" estimado y cada una de las imágenes. En este artículo también se trata la problemática de las sombras, tema sin demasiado interés para las imágenes infrarrojas utilizadas.
- El segundo de los artículos, referencia [2] de la bibliografía, consiste en la detección robusta de objetos en movimiento, utilizada para la navegación segura de robots móviles. En este artículo, se abordan problemas como la separación entre el movimiento de la cámara y la de los objetos del entorno. El movimiento de la cámara es compensado, usando las correspondencias entre las características de las imágenes; y la posición de los objetos móviles, es estimada usando un filtro de partículas adaptativo y un algoritmo EM (Expectation-Maximization).

Se comenzará analizando en el capítulo siguiente el algoritmo de diferenciación, para la detección de movimiento.

## 3.2 ALGORITMO DE DIFERENCIACIÓN

La diferenciación de imágenes, es un método muy efectivo para detectar cambios en los niveles de gris de dos imágenes consecutivas. Es un algoritmo muy utilizado para detectar movimiento en la escena, manteniendo la cámara estática.

Los algoritmos de diferenciación de imágenes, pueden dividirse en tres partes: el primer paso es reconstruir el background, el segundo paso consiste en diferenciar las imágenes, finalmente se aplicará un paso de detección por umbral. Las imágenes de referencia, pueden conseguirse utilizando diferentes métodos: durante un periodo de relativa inactividad, o bien durante un periodo importante de secuencias dinámicas. Otro punto importante en la adaptación de las imágenes de referencia, son los cambios de luminosidad, las sombras, etc... Se utilizan variados métodos para mantener el background actualizado.

### 3.2.1 Diferenciación Básica

La forma más sencilla de implementar un algoritmo de diferenciación, utilizando cámaras estáticas, consiste en analizar las diferencias con la imagen del instante anterior. Este método tiene la ventaja de detectar pequeños cambios entre imágenes consecutivas; aunque también tiene algunas desventajas:

- Sólo se detecta el contorno de los objetos en movimiento. La parte central del objeto móvil no se detecta.
- El segundo de los inconvenientes de este tipo de algoritmos, pierde los objetos cuando éstos se detienen durante un corto periodo de tiempo.

La alternativa para resolver estos problemas, consiste en diferenciar la imagen  $I_{x,y}^t$  respecto del background de referencia. Si este background de referencia, fue calculado cuando la posición e intensidad de luz diferentes a las del instante  $t$ , este background sería obsoleto y no ofrecería buenos resultados.

Para evitar los problemas de la obsolescencia del fondo de la imagen, existen métodos dinámicos que generan y actualizan el background en tiempo real, incluso con objetos desplazándose por la imagen.

### 3.2.2 Generación del Background

Como se ha comentado en el apartado anterior, el background de la imagen  $B_{x,y}$  es generado utilizando la evolución de imágenes  $I_{x,y}^t$ , las cuales pueden contener objetos en movimiento.

Una aproximación para calcular el background, utilizando las imágenes anteriores al instante  $t$ , se puede formular utilizando un filtro de Kalman. Este tipo de algoritmos es utilizado en la referencia [4] de la bibliografía.

Otra alternativa para obtener el background, es analizar la evolución temporal de cada píxel, obteniendo así una sección estable. En la referencia [5], Long y Yang implementan un algoritmo que detecta cambios de intensidad durante cortos periodos de tiempo. La mayor desventaja de esta forma de calcular el background, es la necesidad del ajuste de varios parámetros.

De todas las opciones posibles para calcular el background, la mejor alternativa es utilizar filtros lineales. Estos tienen mucha robustez, pueden tolerar gran cantidad de ruido en la imagen y no requieren demasiados parámetros de ajuste.

### 3.2.3 Umbral Automático de Diferenciación de Imágenes

Una vez obtenido el background, restamos directamente cada imagen y aplicamos un criterio de umbral. En el artículo analizado, se validan diferentes alternativas.

Inicialmente se analiza la diferencia de imagen  $D_{x,y}^t = |I_{x,y}^t - B_{x,y}|$  para determinar la media  $MED^t = med_{x,y \in I} D_{x,y}^t$  y la desviación media absoluta  $MAD^t = med_{x,y \in I} |D_{x,y}^t - MED^t|$ . Suponiendo que menos de la mitad de la imagen está en movimiento, la mediana debe recoger el ruido de la imagen y un valor apropiado de umbral es  $T = MED + 3 \times f \times MAD$ , donde  $f$  es un factor normalizado para una distribución gaussiana.

Otros métodos expuestos en el artículo [1], son técnicas de umbral basadas en conectividad: se aplican diferentes umbrales, y se selecciona aquel que ofrece como resultado una solución más estable.

Algunos autores también utilizan algoritmos de umbral con histéresis, la principal ventaja de utilizar estas técnicas, es conseguir una gran inmunidad al ruido de la imagen.

En la referencia [6] de la bibliografía, Canny obtiene experimentalmente los umbrales superior

e inferior, con los cuales obtiene buenos resultados.

$$R = \sqrt{\frac{\ln 2}{\ln\left(\frac{1+2P}{1+P}\right)}}$$

Donde P indica la probabilidad de encontrar un borde. En este contexto, los umbrales de Canny fueron calculados utilizando P=0.23.



### 3.2.4 Implementación de Algoritmos de Diferenciación

#### 3.2.4.1 Algoritmo de Diferenciación Básico

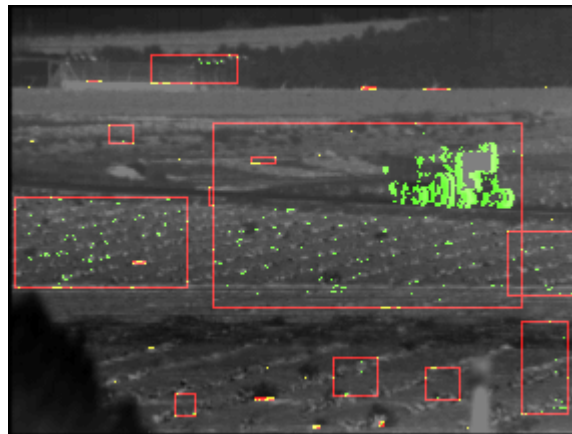
El primer algoritmo implementado en Matlab, consiste en un algoritmo de diferenciación directa entre imágenes consecutivas. Como se ha comentado anteriormente, este algoritmo es muy eficiente desde el punto de vista computacional pero tiene algunas carencias.



**Figura 3-1: Algoritmo de Diferenciación básico (umbral=20)**

En la figura anterior, se muestra una imagen de una secuencia de video. En color verde se representan los píxeles con un diferencial del nivel de gris superior a 20 unidades. En esta imagen también se observa que este tipo de algoritmos sólo detecta los bordes de los objetos. Además, si el vehículo se detuviera durante algunos frames dejaría de detectarse.

En la siguiente figura se muestra el resultado del algoritmo, reduciendo el umbral de detección a 10 niveles de gris. Aquí se observa, como las imágenes reales están muy afectadas por el ruido, y por lo tanto, se debe dotar este tipo de algoritmos de mayor robustez.



**Figura 3-2: Algoritmo de Diferenciación básico (umbral=10)**

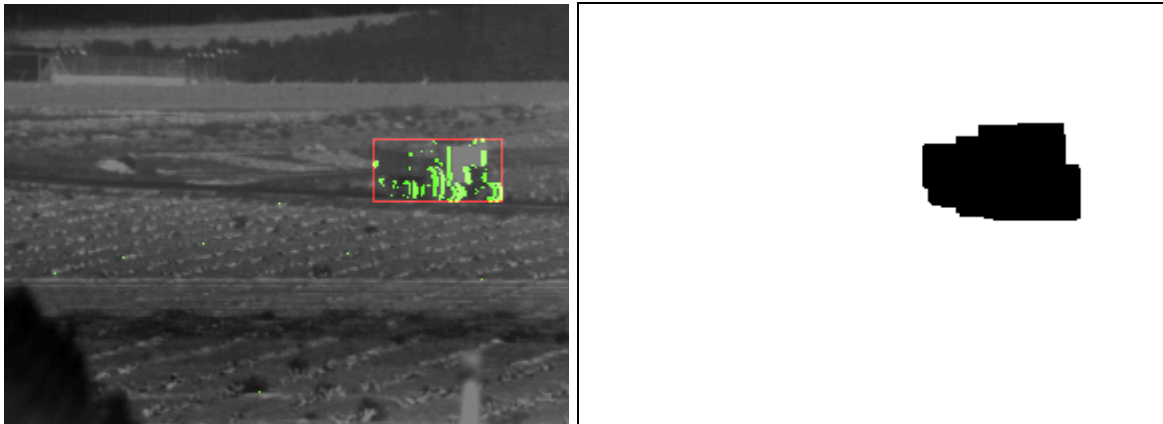
En este algoritmo de diferenciación, se han incluido las siguientes etapas de cálculo:

- El primer paso es calcular el valor absoluto de la diferencia con el frame anterior. Con un umbral estimado experimentalmente de  $\Delta I = 20$  niveles de gris, se obtiene un mapa binario de movimiento.
- El segundo paso, consiste en filtrar los núcleos aislados de ruido. Para ello se aplica una máscara matricial a cada uno de los píxeles con movimiento:

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

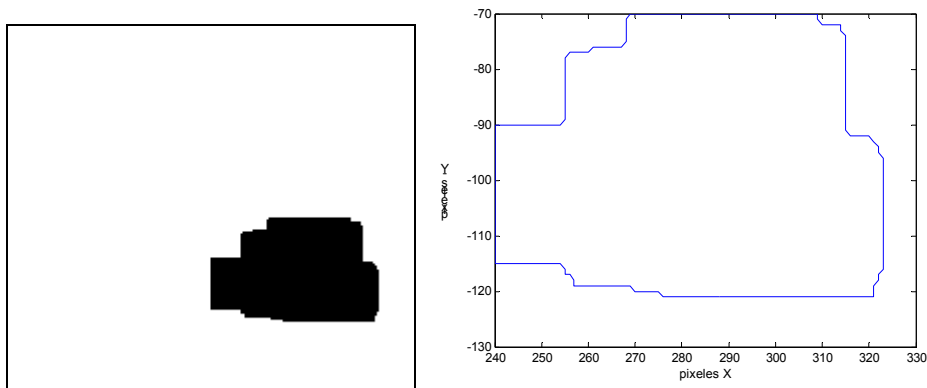
de esta forma, se elimina el problema del ruido en todos los píxeles aislados.

- En el tercer paso del algoritmo, se realiza una dilatación o crecimiento de los puntos detectados, con la finalidad de cerrar contornos. Experimentalmente, se ha aplicado una máscara de  $\pm 10$  píxeles en ambas direcciones, para realizar este crecimiento. En la Figura 3-3 se muestra el paso de dilatación para uno de los vídeos reales utilizados.
- Una vez cerrados los contornos de los objetos de interés, se procede a encontrar los contornos existentes en la imagen. Para ello se utilizan las funciones:
  - La función de Matlab Graythresh: Utiliza el método de Otsu para obtener un nivel que divida la imagen.



**Figura 3-3: Paso 3: Dilatación o Crecimiento**

- La función de Matlab `im2bw`: Convierte la imagen a binario, utilizando el nivel de la función anterior.
- Y utilizando las funciones `imfill` y `bwboundaries`, se obtienen los contornos de todos los objetos encontrados en la imagen.



**Figura 3-4: Paso 4: Detección de contornos**

- El quinto paso del algoritmo, consiste en filtrar y eliminar los contornos no válidos. Para ello se incluyen las siguientes reglas eurísticas en el algoritmo:
  - Las regiones encontradas deben tener una superficie mínima de 2x2 píxeles.

- Y mediante la siguiente desigualdad, se eliminan algunas regiones producidas por ruidos en la imagen. Para considerar una región como válida y estable, debe permanecer como mínimo durante dos frames consecutivos.

$$\min \left\{ \left| \bar{X}_k - \bar{X}_{k-1} \right| + \left| \bar{Y}_k - \bar{Y}_{k-1} \right| < d \right\}$$

Aplicando dicha expresión entre cada contorno detectado y los detectados en la imagen anterior, se consigue el efecto buscado.

- El último paso del algoritmo de diferenciación básico, consiste en remarcar en un rectángulo rojo los objetos móviles detectados. En la siguiente imagen se puede observar como parte del ruido (trazo verde) detectado inicialmente como movimiento, se elimina y deshecha con las etapas de filtrado descritas.



**Figura 3-5: Paso 6: Resultados del algoritmo de diferenciación**

Aunque el umbral se ha ajustado a un nivel y los filtros se han ajustado para eliminar la mayoría de los ruidos de la imagen, en la Figura 3-6 se observa la detección de falsos objetos cuando aparecen píxeles móviles en las inmediaciones del objeto real.



**Figura 3-6: Ruido del Algoritmo de Diferenciación**

Estos ruidos, junto con la detección incompleta del objeto y la pérdida del mismo, si no supera una velocidad mínima, ratifican las conclusiones del artículo y motiva el desarrollo de un método basado en el background de la imagen.

### 3.2.4.2 Generación del Background de la imagen

Con el fin de dotar de mayor robustez al algoritmo de detección de objetos, poder detectar el 100% de la superficie del objeto y permitir pequeñas paradas de los objetos móviles sin pérdida de detección, se incluye el cálculo del background de referencia para compararlo con cada una de las imágenes.

La primera de las dificultades encontrada para generar el background de la imagen, es la cantidad de objetos en movimiento durante su generación. Para solventar dicho problema, se han descartado las regiones en movimiento proporcionadas por el algoritmo de diferenciación básico.

En las regiones con movimiento, el algoritmo se inicializa con un nivel de gris intermedio y mediante un filtro lineal, se actualiza el background utilizando la secuencia de imágenes.



**Figura 3-7: Imagen inicial del background**

En la figura anterior se representa la imagen inicial del background. La región donde el algoritmo de diferenciación detecta movimiento, se inicializa con un nivel de gris intermedio. Cuando el vehículo detectado se desplaza, la zona gris, adquiere la textura del fondo como se puede apreciar en la siguiente figura.



**Figura 3-8: Background Actualizado**

El cálculo del background se realiza utilizando un filtro lineal de primer orden, que se corresponde con la expresión discreta:

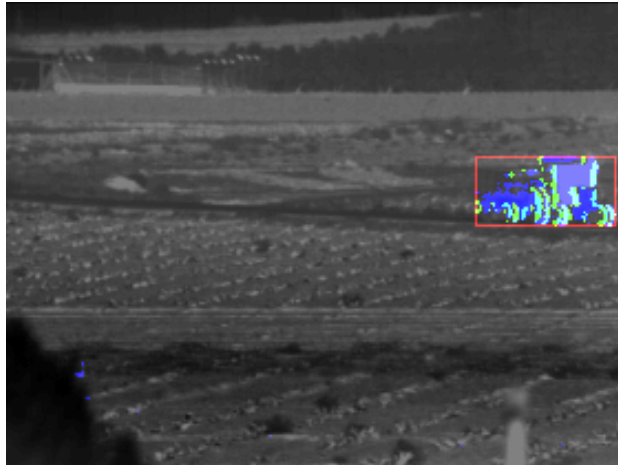
$$F(z) = \frac{0.0402z + 0.0402}{z - 0.9196}$$

Aplicado con un periodo de 67 ms (15fps), y con una frecuencia de corte de aproximadamente 0.2Hz.

Una secuencia de vídeo completa de la evolución del background, puede visualizarse en el archivo [tractor\\_background.avi](#).

### 3.2.4.3 Diferenciación con del Background de referencia

En esta última fase del algoritmo, se ha optado por establecer un umbral constante e independiente de los parámetros de la imagen. En este paso, se calcula la diferencia entre cada una de las imágenes y el background, y el umbral prefijado divide la imagen en zonas estáticas y objetos en movimiento.



**Figura 3-9: Diferenciación con el fondo**

En la Figura 3-9 se representan en un recuadro rojo los objetos móviles detectados, en verde las aristas detectadas por diferenciación básica y en azul los objetos móviles localizados utilizando la información del background. También puede observarse como se detectan los objetos móviles casi en su totalidad, en lugar de detectar solamente los bordes.

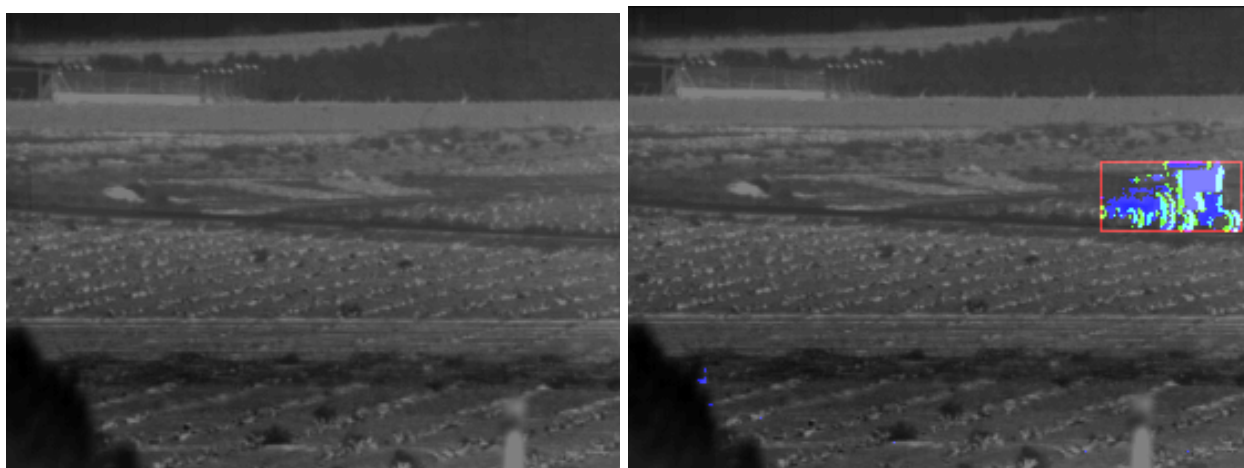


### 3.3 RESULTADOS EXPERIMENTALES

En esta sección, se exponen los resultados experimentales de los algoritmos de detección de movimiento. Para el desarrollo y depuración de dichos algoritmos, se utilizaron vídeos de cámaras infrarrojas grabados en un escenario real. Además de las capturas representadas en las gráficas, un enlace directo a los videos generados por el algoritmo pueden encontrarse anexo a cada ejemplo.

#### 3.3.1 Vídeo 1

En este ejemplo se realiza la detección de movimiento de un tractor que transita por la imagen. En la Figura 3-10 se observan una foto del background generado y una imagen representativa del móvil detectado.



**Figura 3-10: Resultados vídeo 1**

Se representan en un recuadro rojo los objetos móviles detectados, en verde las aristas detectadas por diferenciación básica y en azul los objetos móviles localizados utilizando la información del background.

Para observar la secuencia de imágenes completa, se ofrece la posibilidad de visualizar un vídeo de la generación del background: [tractor\\_background.avi](#) , como del seguimiento del objeto: [tractor\\_diff\\_bg.avi](#).

### 3.3.2 Vídeo 2

En este ejemplo se realiza la detección del movimiento de un camión que transita por la imagen. En la siguiente figura se observan una foto del background generado y una imagen representativa del móvil detectado.



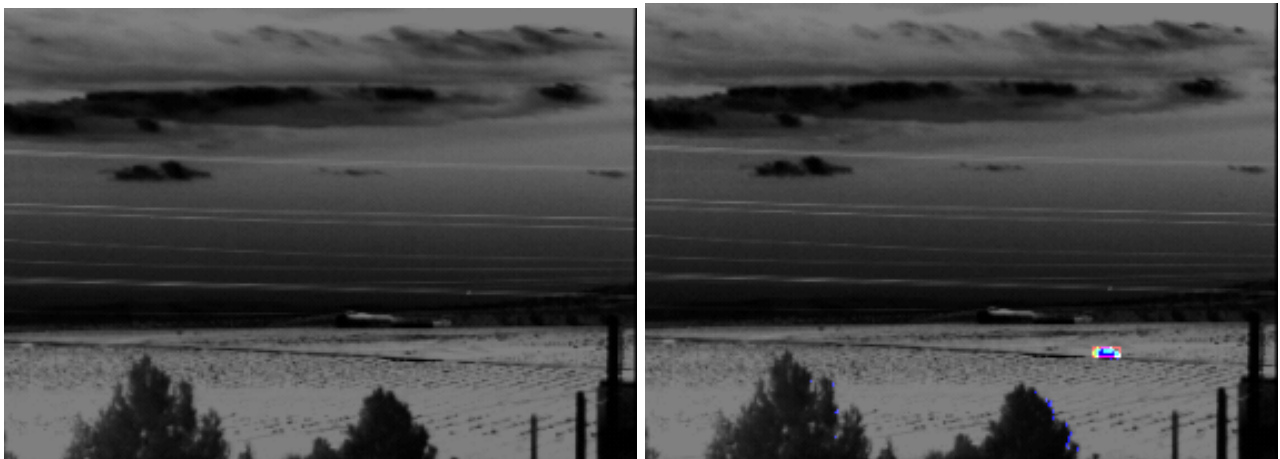
**Figura 3-11: Resultados vídeo 2**

Se representan en un recuadro rojo los objetos móviles detectados, en verde las aristas detectadas por diferenciación básica y en azul los objetos móviles localizados utilizando la información del background.

Para observar la secuencia de imágenes completa, se ofrece la posibilidad de visualizar un vídeo del seguimiento del objeto [camion\\_diff\\_bg.avi](#).

### 3.3.3 Vídeo 3

En este ejemplo se realiza la detección del movimiento de un coche que transita por la imagen. En la siguiente figura se observan una foto del background generado y una imagen representativa del móvil detectado.



**Figura 3-12: Resultados vídeo 3**

Se representan en un recuadro rojo los objetos móviles detectados, en verde las aristas detectadas por diferenciación básica y en azul los objetos móviles localizados utilizando la información del background.

Para observar la secuencia de imágenes completa, se ofrece la posibilidad de visualizar un vídeo del seguimiento del objeto [coche\\_diff\\_bg.avi](#). El background también puede visualizarse en el fichero anexo [coche\\_background.avi](#).

### 3.3.4 Vídeo 4

En este ejemplo se realiza la detección del movimiento de dos personas que transitan por la imagen. En la siguiente figura se observan una foto del background generado y una imagen representativa del móvil detectado.



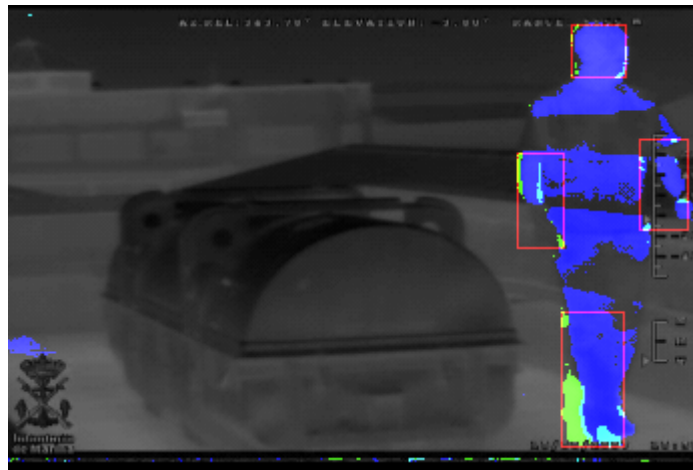
**Figura 3-13: Resultados vídeo 4**

Se representan en un recuadro rojo los objetos móviles detectados, en verde las aristas detectadas por diferenciación básica y en azul los objetos móviles localizados utilizando la información del background.

Para observar la secuencia de imágenes completa, se ofrece la posibilidad de visualizar un vídeo del seguimiento del objeto [personas\\_diff\\_bg.avi](#).

### 3.3.5 Vídeo 5

En este ejemplo se realiza la detección del movimiento de una persona situada cerca de la cámara. En la siguiente figura se observan una foto del background generado y una imagen representativa de las zonas detectadas.



**Figura 3-14: Resultados vídeo 5**

Se representan en un recuadro rojo los objetos móviles detectados, en verde las aristas detectadas por diferenciación básica y en azul los objetos móviles localizados utilizando la información del background.

Para observar la secuencia de imágenes completa, se ofrece la posibilidad de visualizar un vídeo del seguimiento del objeto [persona\\_wfov\\_diff\\_bg.avi](#).

Como conclusiones del presente trabajo, se ha realizado una búsqueda de información y artículos publicados para conocer el estado del arte en el ámbito de la detección de movimiento.

Se ha dejado pendiente para el trabajo de master los temas relacionados con detección de movimiento utilizando cámaras en movimiento, centrando la atención en algoritmos utilizando imágenes estáticas.

En cuanto a los artículos seleccionados, muestran información muy variada sobre los tipos y aplicaciones en el ámbito de la detección de movimiento. Principalmente, ofrecen pautas importantes para la implementación de algoritmos robustos que funcionen con imágenes reales.

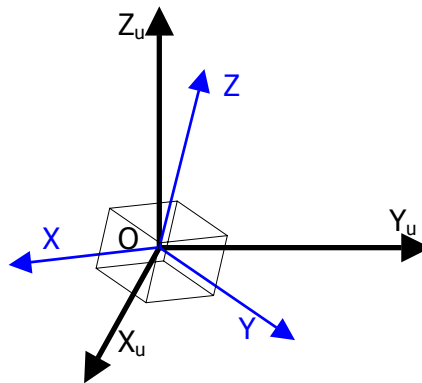
Sobre las rutinas implementadas, la diferenciación básica se ha dotado de bastante robustez al ruido de las imágenes y a otras características de las imágenes utilizadas. Además, dicho algoritmo es una excelente base de búsqueda para el resto de funciones desarrolladas.

El cálculo del background y la diferenciación por umbral, cubren las carencias de la detección incompleta de los objetos y del problema de detección para velocidades bajas. Se puede concluir que el algoritmo de diferenciación básica junto al algoritmo de umbral utilizando fondos, es una herramienta aceptable para muchas aplicaciones de detección de objetos móviles en cámaras fijas.

## 4. ODOMETRÍA INERCIAL

### 4.1 INTRODUCCIÓN

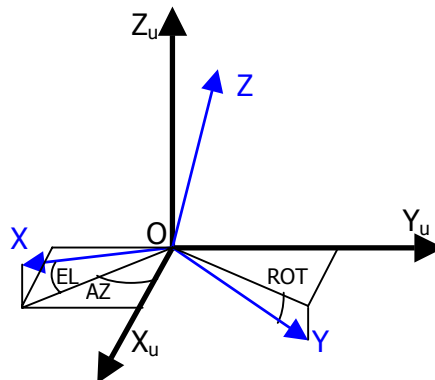
Para exponer el problema, partimos de un sistema de coordenadas solidario con el universo  $OX_uY_uZ_u$  representado en la Figura 4-1, y un triedro que representa la orientación del sistema móvil  $OXYZ$ . El primero de los problemas abordados en el trabajo, consistirá en desarrollar un método de integración, que nos permita calcular la orientación del sistema móvil respecto del sistema  $OX_uY_uZ_u$ . Para conseguir este objetivo, utilizaremos exclusivamente la velocidad angular inercial del sistema móvil, que viene dada por el vector  $[\omega_x \ \omega_y \ \omega_z]^T$ .



**Figura 4-1: Triedro Inercial y sistema de referencia móvil**

Para representar la orientación del triedro móvil con respecto al triedro de referencia inercial, utilizaremos una notación interesante desde el punto de vista de sistemas estabilizados. Estas aplicaciones prácticas podrían ir desde antenas de comunicaciones, cámaras, telémetros o similares donde la línea de mira ha de mantenerse estable, e independiente del movimiento de la base donde se asientan.

Bajo esta hipótesis, representaremos los tres ángulos que expresan la orientación de la línea de mira: elevación (EL), orientación o acimut (AZ) y rotación con respecto al plano  $Z_u=0$  como (ROT) (Ver Figura 4-2).



**Figura 4-2: Ángulos de orientación del sistema de referencia móvil**

A partir de la matriz de cambio de base representada por los vectores columna  $[X^T \ Y^T \ Z^T]$ , se obtienen las expresiones que relacionan la orientación del sistema móvil con respecto al triedro de referencia inercial  $OX_u Y_u Z_u$ :

$$ROT = \arctg\left(\frac{Y_3}{Z_3}\right)$$

$$EL = \arcsen(X_3)$$

$$AZ = \arctg\left(\frac{X_2}{X_1}\right)$$

**Ecuación 4.1: Ángulos de orientación del sistema de referencia móvil**

Para calcular las tres componentes angulares: orientación, elevación y rotación, se plantea resolver la integral desacoplada mostrada en la Ecuación 4.2.

$$\begin{bmatrix} ROT(t) \\ EL(t) \\ AZ(t) \end{bmatrix} = \int_0^t \begin{bmatrix} \dot{ROT}(\tau) \\ \dot{EL}(\tau) \\ \dot{AZ}(\tau) \end{bmatrix} \cdot \partial\tau$$

**Ecuación 4.2: Integral de las coordenadas angulares**

Este problema sería muy sencillo de resolver, si estuvieran accesibles las velocidades de las coordenadas angulares, en el sistema de referencia inercial  $OX_u Y_u Z_u$ . En cambio, al disponer de las tres velocidades giroscópicas en los ejes de referencia del sistema móvil, aparece un acoplamiento



entre las coordenadas angulares debido a la forma de la matriz Jacobiana.

$$\begin{bmatrix} ROT(t) \\ EL(t) \\ AZ(t) \end{bmatrix} = \int_0^t \bar{\mathbf{J}} \cdot \begin{bmatrix} \omega_x(\tau) \\ \omega_y(\tau) \\ \omega_z(\tau) \end{bmatrix} \cdot d\tau$$

**Ecuación 4.3: Integral de velocidades giroscópicas**

La Ecuación 4.3 y concretamente la forma de la matriz Jacobiana, complica el cálculo de la citada integral puesto que esta depende de las coordenadas ROT, EL y AZ; y obligaría a calcular y resolver un problema de autovalores en cada paso de integración.

**4.2 Fundamentos Teóricos**

**4.2.1 Operador Matricial**

En este capítulo, se intentará encontrar un método de resolución del primero de los problemas planteados. Tras evaluar diferentes opciones de posibles desarrollos matemáticos, y fundamentalmente para permitir un movimiento inercial combinado y obtener mayor exactitud en los resultados; se decide utilizar una transformación basada en el teorema de rotación de Euler.

Puesto que en el caso más genérico, el sistema móvil podría rotar respecto de los tres ejes simultáneamente, debemos encontrar una transformación que nos permita integrar las velocidades giroscópicas bajo esta hipótesis. Para obtener el operador matricial, podemos asumir que el movimiento de rotación se realiza alrededor de un eje que coincide con las componentes de las velocidades giroscópicas. Se puede obtener una orientación arbitraria en el espacio, girando un sistema de referencia un ángulo  $\theta$ , en torno a una dirección dada por un vector unitario  $\mathbf{K}$  (eje de giro).

Con un desarrollo matricial de Euler y cogiendo como eje un vector unitario cuyas componentes se denotan como  $\mathbf{K} = [K_x, K_y, K_z]^T$ , se obtiene el operador matricial de cambio de base representado en la Ecuación 4.4.

$$R(\theta, \bar{K})_K = \begin{bmatrix} K_X^2 v \theta + c \theta & K_X K_Y v \theta - K_Z s \theta & K_X K_Z v \theta + K_Y s \theta \\ K_X K_Y v \theta + K_Z s \theta & K_Y^2 v \theta + c \theta & K_Z K_Y v \theta - K_X s \theta \\ K_X K_Z v \theta - K_Y s \theta & K_Z K_Y v \theta + K_X s \theta & K_Z^2 v \theta + c \theta \end{bmatrix} \quad \text{donde} \quad \begin{aligned} s \theta &= \text{sen}(\theta) \\ c \theta &= \text{cos}(\theta) \\ v \theta &= 1 - \text{cos}(\theta) \end{aligned}$$

**Ecuación 4.4: Operador Matricial de Rotación**

Las principales ventajas de este desarrollo, son la exactitud del cálculo, y la simplificación computacional obtenida al utilizar una sola matriz de cambio de base. Otra ventaja importante, es que el operador obtenido no depende de los ángulos de Euler intermedios; ya que dichos ángulos no se corresponden con giros reales del sistema móvil.

#### 4.2.2 Integración de Velocidad Inercial

Con el operador desarrollado en la sección anterior, se puede integrar la velocidad inercial de los tres sensores giroscópicos, sin necesidad de resolver un problema de autovalores en cada periodo de muestreo. El procedimiento para calcular la posición inercial, consistirá en post-multiplicar la matriz que define el sistema de referencia móvil por el operador matricial. Dicho operador, se construye con las lecturas giroscópicas y el ángulo de rotación respecto del eje de referencia.

$$M_K = \begin{bmatrix} X_{1K} & Y_{1K} & Z_{1K} \\ X_{2K} & Y_{2K} & Z_{2K} \\ X_{3K} & Y_{3K} & Z_{3K} \end{bmatrix} = M_{K-1} \cdot R(\theta, \bar{K})_K$$

**Ecuación 4.5: Integración de Velocidad Inercial**

Desde el punto de vista de ahorro computacional, con el método definido en la Ecuación 4.5, se ha reducido a una multiplicación matricial la integración de velocidades del triedro móvil.

#### 4.2.3 Matriz Jacobiana

Aunque se ha desestimado en las secciones anteriores el cálculo de esta matriz para el problema de la integración, es interesante analizar el comportamiento del Jacobiano para

encontrar las relaciones entre las velocidades del triedro móvil y el sistema de referencia inercial.

Puesto que el valor numérico de cada uno de los elementos de la Jacobiana dependerá de los valores instantáneos de los ángulos ROT, EL y AZ, el valor de la matriz será diferente en cada una de las orientaciones del triedro móvil. En esta sección, se calcula la matriz Jacobiana para el caso particular  $M_{K-1}$  igual a la identidad y donde el operador matricial de rotación gira un ángulo que tiende a cero ( $\theta \rightarrow 0$ ). Bajo estas condiciones, en las siguientes ecuaciones se representa el cálculo de matriz Jacobiana.

$$ROT = \arctg\left(\frac{Y_3}{Z_3}\right) = \arctg\left(\frac{K_Z K_Y v \theta + K_X s \theta}{K_Z^2 v \theta + c \theta}\right)$$

$$EL = \arcsen(X_3) = \arcsen(K_X K_Z v \theta - K_Y s \theta)$$

$$AZ = \arctg\left(\frac{X_2}{X_1}\right) = \arctg\left(\frac{K_X K_Y v \theta + K_Z s \theta}{K_X^2 v \theta + c \theta}\right)$$

**Ecuación 4.6: Ángulos respecto del sistema de referencia inercial**

Utilizando las relaciones de ángulos de la Ecuación 4.6 y derivando respecto del tiempo tal y como se indica en la Ecuación 4.7, se obtienen las siguientes expresiones diferenciales.

$$\frac{\partial ROT}{\partial t} = \frac{\partial ROT}{\partial K_X} \cdot \dot{K}_X + \frac{\partial ROT}{\partial K_Y} \cdot \dot{K}_Y + \frac{\partial ROT}{\partial K_Z} \cdot \dot{K}_Z + \frac{\partial ROT}{\partial \theta} \cdot \dot{\theta}$$

$$\frac{\partial EL}{\partial t} = \frac{\partial EL}{\partial K_X} \cdot \dot{K}_X + \frac{\partial EL}{\partial K_Y} \cdot \dot{K}_Y + \frac{\partial EL}{\partial K_Z} \cdot \dot{K}_Z + \frac{\partial EL}{\partial \theta} \cdot \dot{\theta}$$

$$\frac{\partial AZ}{\partial t} = \frac{\partial AZ}{\partial K_X} \cdot \dot{K}_X + \frac{\partial AZ}{\partial K_Y} \cdot \dot{K}_Y + \frac{\partial AZ}{\partial K_Z} \cdot \dot{K}_Z + \frac{\partial AZ}{\partial \theta} \cdot \dot{\theta}$$

**Ecuación 4.7: Relaciones diferenciales**

Operando analíticamente la Ecuación 4.7, y expresando el resultado en forma matricial para el caso particular de  $M_{K-1}=I$  y ( $\theta \rightarrow 0$ ), la Matriz Jacobiana tiene la forma mostrada en la Ecuación 4.8.

$$\begin{bmatrix} \dot{ROT} \\ \dot{EL} \\ \dot{AZ} \end{bmatrix} = J \cdot \begin{bmatrix} \dot{\omega}_X \\ \dot{\omega}_Y \\ \dot{\omega}_Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{\omega}_X \\ \dot{\omega}_Y \\ \dot{\omega}_Z \end{bmatrix}$$

**Ecuación 4.8: Matriz Jacobiana para  $M_{k-1}=\mathbf{I}$  y  $(\theta \rightarrow 0)$**

Para el caso particular calculado, aparece un desacoplo entre los tres grados de libertad.

### 4.3 Errores de Integración

En cuanto a los errores de integración de las medidas angulares, tendremos tres fuentes principales de error: los errores de redondeo producidos por limitaciones numéricas, la precisión del sensor para velocidades nulas y la linealidad de las medidas inerciales para velocidades no nulas.

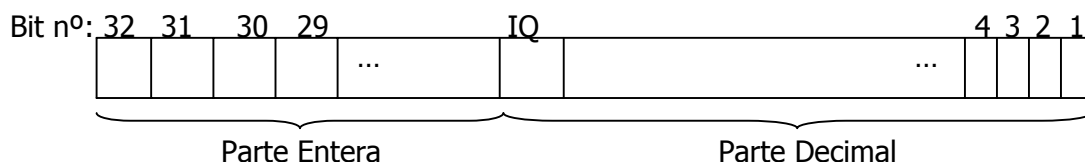
De las tres fuentes principales de deriva o errores de integración, se pueden realizar las siguientes acciones correctoras: sobre los errores numéricos, se puede buscar un compromiso entre precisión y tiempo de muestreo, de forma que se minimicen dichos errores. La deriva para velocidades nulas, está garantizada por el fabricante del giróscopo con una cota inferior a 0.25°/min. Y por último, respecto al error de velocidades inerciales no nulas, tendremos una gran incertidumbre y las no linealidades serán responsables de las mayores aportaciones a los errores de integración.

#### 4.3.1 Errores de Redondeo

Según la Ecuación 4.5, el método de integración se reduce a multiplicar una matriz 3x3 de cambio de base (con todos sus términos acotados en el intervalo [-1, 1]), por otra matriz 3x3 de rotación con todos sus términos acotados en el mismo intervalo [-1, 1].

Debido a la limitación del hardware (DSP industrial), donde se implementa dicho algoritmo, se decide utilizar una representación en coma fija.

La longitud de los datos será 32bits con una precisión definida por la cota  $\frac{1}{2^{IQ}}$ , donde el índice IQ representa la cantidad de bits destinados a la información decimal.



Se destinan IQ bits a la representación de la parte decimal, con una precisión de  $\frac{1}{2^{IQ}}$  y el resto:  $(2^{32-IQ} - 1)$  bits, a la parte entera.

Una vez definido el tipo de representación numérica, se pretende encontrar una cota para los errores angulares debidos a la precisión numérica de cada iteración del algoritmo. Con esta cota angular y el tiempo de muestreo, será inmediato calcular una cota para la deriva.

Para el cálculo del producto de dos matrices 3x3, los errores de cada elemento serán debidos a tres productos y dos sumas:

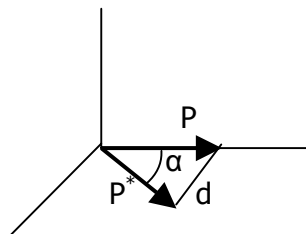
Una cota del error de cada producto puede encontrarse con la siguiente expresión:

$$d1 \times d2 = \frac{\text{mantisa}(d1) \cdot \text{mantisa}(d2)}{2^{2 \cdot IQ}} \Rightarrow |\text{error}(d1 \times d2)| < \frac{1}{2^{IQ}}$$

La cota del error para las sumas de cada término de la matriz, se puede calcular como la suma de las cotas de cada uno de los términos.

$$|\text{error}(d1 + d2 + d3)| \leq \frac{1}{2^{IQ}} + \frac{1}{2^{IQ}} + \frac{1}{2^{IQ}} = \frac{3}{2^{IQ}}$$

Utilizando la cota del error de cada uno de los términos del producto matricial  $\frac{3}{2^{IQ}}$ , se calcula una cota para el ángulo de los vectores de la matriz de cambio de base. Para calcular una estimación de la cota de error  $\alpha$  entre el vector director real P y el vector obtenido P\*.



Debido a la propiedad de vectores unitarios que tienen que cumplir estas matrices de

cambio de base, se puede estimar una cota angular para la dirección de los vectores de dirección.

$$\text{sen}(\alpha) \approx \frac{d}{1} \leq \sqrt{\left(\frac{3}{2^{IQ}}\right)^2 + \left(\frac{3}{2^{IQ}}\right)^2 + \left(\frac{3}{2^{IQ}}\right)^2} = \frac{3\sqrt{3}}{2^{IQ}} \quad \text{asumiendo que } \alpha \rightarrow 0 \Rightarrow |\alpha| \leq \frac{3\sqrt{3}}{2^{IQ}}$$

Si representamos la cota de los errores angulares en función de la precisión decimal, se obtienen los datos representados en la Tabla 4-1.

IQ [bits]	error(α)  [rad]
8	$2.03 \cdot 10^{-2}$
16	$7.92 \cdot 10^{-5}$
20	$4.96 \cdot 10^{-6}$
24	$3.09 \cdot 10^{-7}$
30	$4.84 \cdot 10^{-9}$

**Tabla 4-1: Errores angulares frente a la precisión numérica**

Utilizando los cálculos del apartado anterior, se puede obtener directamente una estimación de la cota para la deriva en función del tiempo de muestreo del algoritmo.

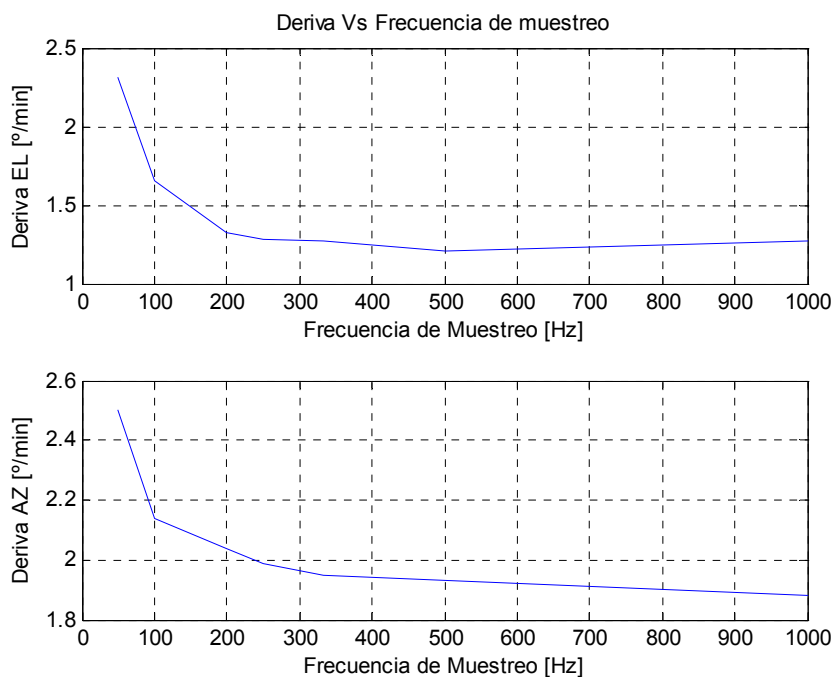
$$|\omega_{\text{deriva}}| \leq \frac{|\alpha|}{T_m} \leq \frac{3\sqrt{3}}{T_m 2^{IQ}}$$

En esta expresión, se observa la relación directa de la deriva inercial con la precisión numérica y el tiempo de muestreo.

Para decidir la precisión numérica a utilizar y el tiempo de muestreo, se plantea aplicar el algoritmo a los mismos datos experimentales con diferentes frecuencias de muestreo. De esta forma, se obtiene una relación óptima de dichos parámetros.

$f_m$ [Hz]	deriva EL  [°/min]	deriva AZ  [°/min]
50Hz	2.31	2.50
100Hz	1.65	2.14
200Hz	1.32	2.04
250Hz	1.28	1.99
333Hz	1.27	1.95
500Hz	1.21	1.93
1KHz	1.27	1.88

**Tabla 4-2: Variación de la deriva frente a la frecuencia de muestreo**



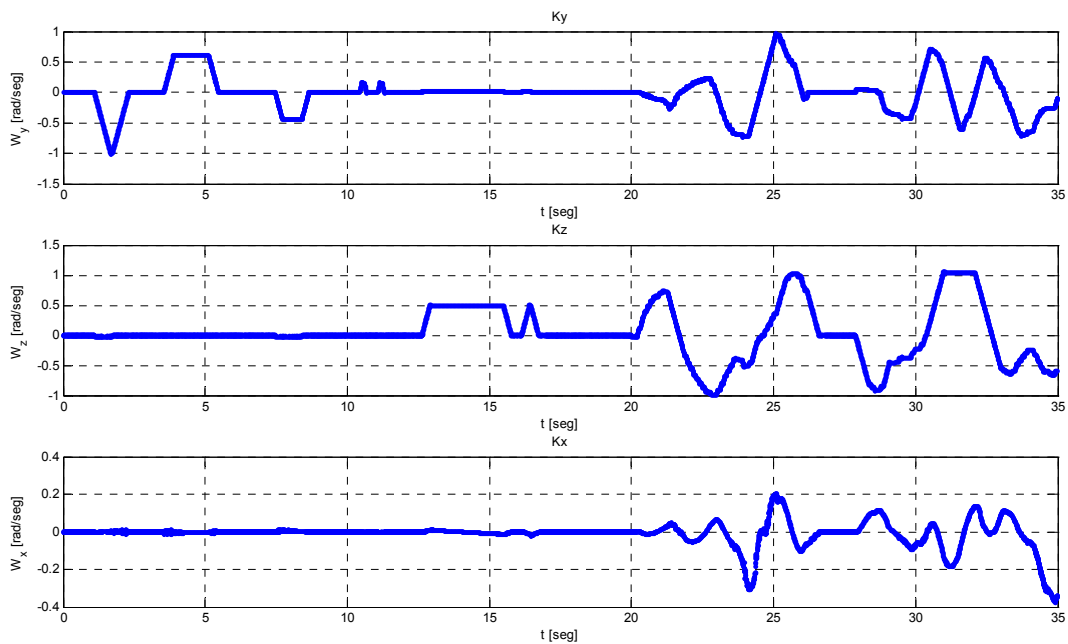
**Figura 4-3: Variación de la deriva frente a la frecuencia de muestreo**

De los datos experimentales obtenidos, se puede fijar una frecuencia de muestreo óptima en torno a 250Hz. Por encima de esta frecuencia, se multiplicarían los cálculos sin obtener mejoras importantes en los resultados. En cuanto a la precisión, se ha seleccionado la coma fija en 24 bits (IQ=24), que evita excesivos errores de redondeos y ofrece una precisión suficiente.



#### 4.4 DATOS EXPERIMENTALES DE INTEGRACIÓN INERCIAL

El algoritmo se aplicó con una frecuencia de muestreo de 250Hz, sobre las velocidades inerciales de un giróscopo de tres ejes. En la Figura 4-4 se representan las velocidades angulares medidas por los sensores inerciales.



**Figura 4-4: Velocidades Inerciales del giróscopo**

Con la finalidad de comparar los resultados de integración, se emplean resolvers de alta precisión (aprox.: 330000 pulsos/vuelta) utilizados para conocer los ángulos articulares y contrastar la posición integrada inercialmente. El sistema robótico, dispone de cuatro grados de libertad A, B, C y D relacionados por las siguientes expresiones angulares:

$$ROT_{Base} = \arctg\left(\frac{Y_3}{Z_3}\right) = \arctg\left(\frac{\text{sen}(B+C)\text{sen}(A)}{\cos(B+C)}\right)$$

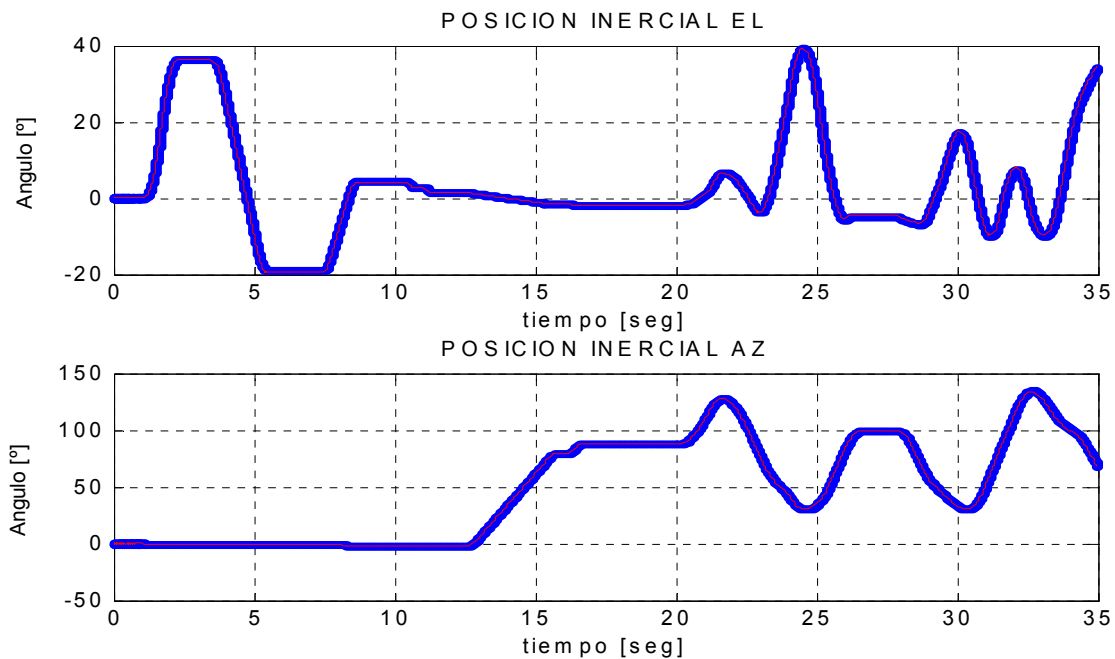
$$EL_{Base} = \arcsen(X_3) = -\arcsen(\text{sen}(B+C)\cos(A))$$

$$AZ_{Base} = \arctg\left(\frac{X_2}{X_1}\right) = \arctg\left(\frac{\text{sen}(D)\cos(B+C)\cos(A) + \cos(D)\text{sen}(A)}{\cos(D)\cos(B+C)\cos(A) - \text{sen}(D)\text{sen}(A)}\right)$$

**Ecuación 4.9: Ángulos Articulares Respecto de la Base**

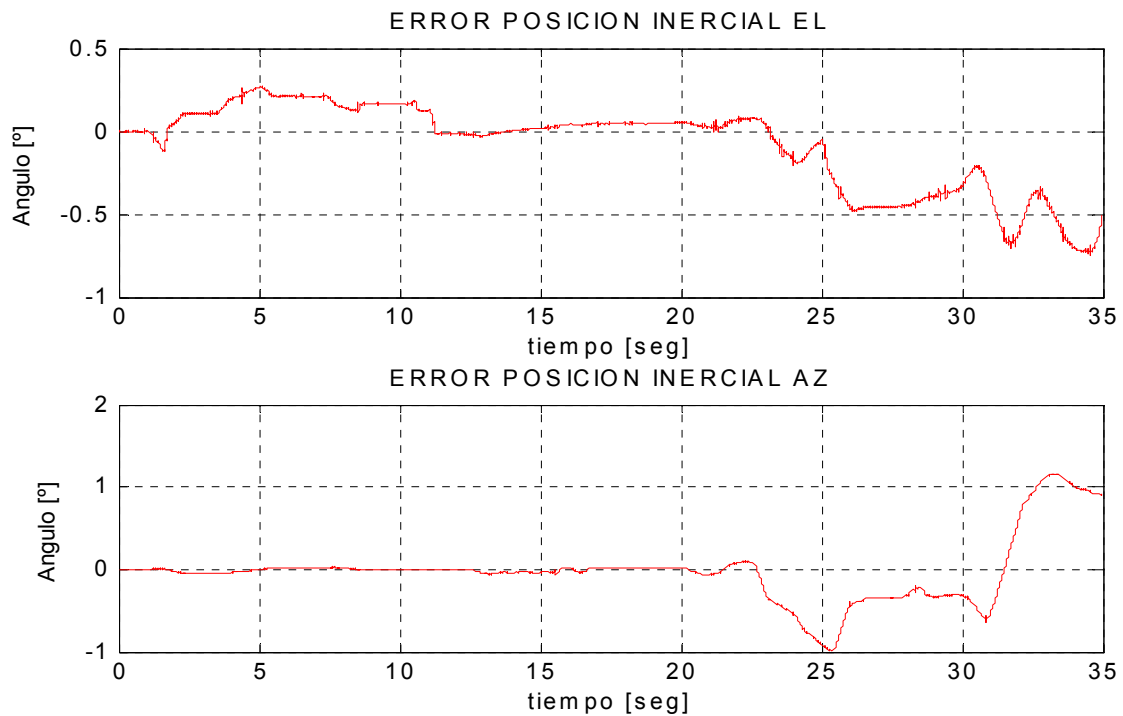
Para utilizar las medidas articulares (ángulo de los 4 resolvers), y comprobar la bondad del método de integración, se fijó la base del equipo al sistema de referencia inercial. De esta forma, el sistema inercial y sistema de la base fueron solidarios durante todo el experimento. Una vez verificada esta condición inicial, mediante un PC de control (RS-422), se comandaron movimientos inerciales y se aplicó el algoritmo de odometría inercial con una frecuencia de 250Hz.

Los resultados de posición inercial se representan en la Figura 4-4: Velocidades Inerciales del giróscopo. Se han integrado las medidas del giróscopo, con el operador matricial definido en capítulos anteriores; y se representan los ángulos de elevación (EL) y orientación (AZ) frente al tiempo en la siguiente figura.



**Figura 4-5: Coordenadas articulares Vs Coordenadas articulares Integradas.**

En la Figura 4-5, se han representado en trazo azul las medidas articulares calculadas con los resolvers (a partir de las Ecuación 4.9); y en trazo rojo, las medidas integradas con el algoritmo definido en este trabajo. Aunque en la citada gráfica, no se aprecia error alguno entre los valores integrados y las medidas de los resolvers, realmente como en cualquier método de integración, se han generado unos errores que podemos apreciar en la Figura 4-6. Además, una cota para la deriva puede calcularse a partir de dichos resultados experimentales.



**Figura 4-6: Errores del algoritmo de Integración.**

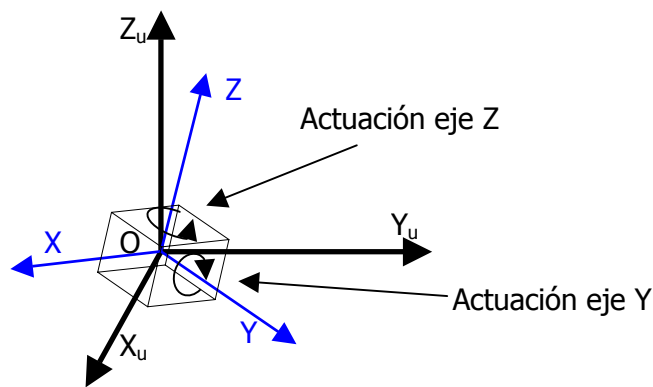
Una estimación de la deriva para las medidas de EL y AZ son  $1.28^{\circ}/\text{min}$  y  $1.99^{\circ}/\text{min}$  respectivamente. Además, como conclusión de un análisis de los errores del algoritmo de integración, se observa que la principal fuente de error, se produce por la no linealidad del giróscopo para velocidades inerciales elevadas. También puede destacarse, que errores numéricos y errores del sensor a velocidades bajas, introducen derivas despreciables en la integración.

## 4.5 GENERADOR DE TRAYECTORIAS

Tras solventar la integración 3-D desarrollada en los capítulos anteriores, la segunda línea desarrollada en este trabajo, es obtener la estrategia de control para apuntar el sistema de referencia móvil. Dicha estrategia de control, deberá minimizar el tiempo en realizar el movimiento, y deberá optimizar la trayectoria tridimensional. Se tendrán presentes limitaciones físicas de velocidad y aceleración que obligarán a diseñar un filtro para las velocidades de referencia.

### 4.5.1 Fundamentos Teóricos

Para plantear la formulación del problema abordado en este capítulo, partimos de una orientación arbitraria del sistema móvil con respecto al sistema inercial. Dicha orientación se define por la matriz de cambio de base  $M_k=[X^T Y^T Z^T]$ . Se pretende orientar el eje  $OX^T$  a una dirección  $OX_{REF\_U}^T$  del sistema inercial, optimizando la trayectoria y utilizando solamente dos actuadores angulares (Y, Z). La actuación de los ejes Y, Z tienen la característica principal que consisten en motores de accionamiento directo y permiten girar con precisión el sistema móvil respecto de los citados ejes.



**Figura 4-7: Direct-Drive de actuación en ejes Y, Z**

Para realizar el movimiento de orientación del sistema móvil, se utilizará el operador matricial de rotación definido en el apartado 4.2.1 de este trabajo. En cuanto a la definición del eje sobre el cual se pretende rotar la matriz de cambio de base  $M_k$ , se realizarán los siguientes pasos

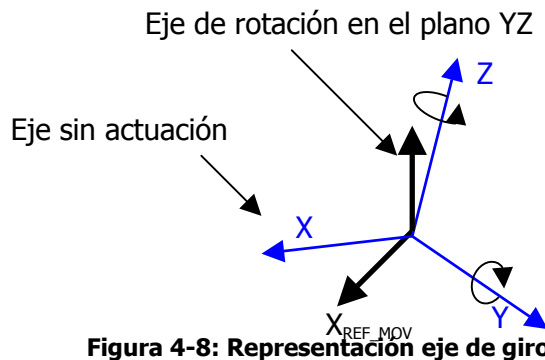
para obtener el eje de giro buscado:

1. Para comenzar, el vector donde se pretende orientar el sistema móvil  $X_{REF\_U}^T$  representado en coordenadas del sistema inercial, se debe proyectar sobre el sistema de referencia móvil utilizando la matriz de cambio de base:

$$X_{REF\_MOV}^T = M_k^T \cdot X_{REF\_U}^T$$

**Ecuación 4.10: Proyección de la dirección de referencia en el triedro móvil**

2. Una vez obtenido la dirección de apuntamiento con respecto a las coordenadas del propio sistema móvil, resulta lógico pensar que el eje de rotación óptimo, será perpendicular al eje de referencia y a la dirección X (puesto que carece de actuador rotativo). La Figura 4-8 representa el sistema de referencia móvil, el eje de rotación obtenido, y el vector de referencia  $OX_{REF\_MOV}$  (Ver Ecuación 4.11).



**Figura 4-8: Representación eje de giro**

$$EJE_{ROT} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ 1 & 0 & 0 \\ X_1^{REF\_MOV} & X_2^{REF\_MOV} & X_3^{REF\_MOV} \end{vmatrix} = -X_3^{REF\_MOV} \cdot \vec{j} + X_2^{REF\_MOV} \cdot \vec{k}$$

**Ecuación 4.11: Cálculo del eje de rotación**

3. Después de calcular las componentes del vector de rotación, mediante un producto escalar, se obtiene el ángulo de rotación para conseguir el apuntamiento deseado.

La Ecuación 4.12, representa la expresión para obtener el ángulo total, entre la posición del sistema móvil y la dirección de referencia.

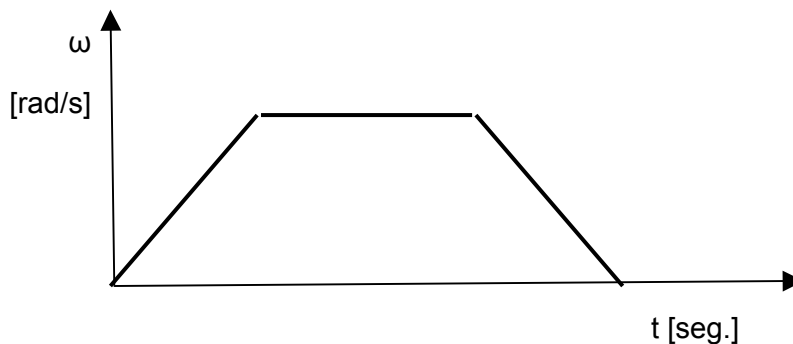
$$\theta_k = -\arccos\left(\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \cdot X_{REF\_MOV}^T\right) = -\arccos(X_1^{REF\_MOV})$$

**Ecuación 4.12: Ángulo de giro**

Evidentemente, debido a la dinámica del sistema, los movimientos angulares respecto al eje de rotación deberán de estar acotados en velocidad y aceleración. El siguiente apartado define una forma de prefiltrar este ángulo, para mantener acotadas la dinámica y cinemática en unos márgenes.

**4.5.2 Pre-filtro de velocidad y aceleración**

En este apartado se pretende calcular el ángulo que el operador matricial de rotación modificará el sistema en cada periodo de muestreo. Para dicho fin, una posibilidad es prefiltrar dicho ángulo con un perfil trapezoidal de velocidad. El movimiento se inicia con una aceleración constante hasta conseguir una velocidad máxima, se mantiene a velocidad máxima durante un tiempo, y finalmente decelerará con el máximo fijado.



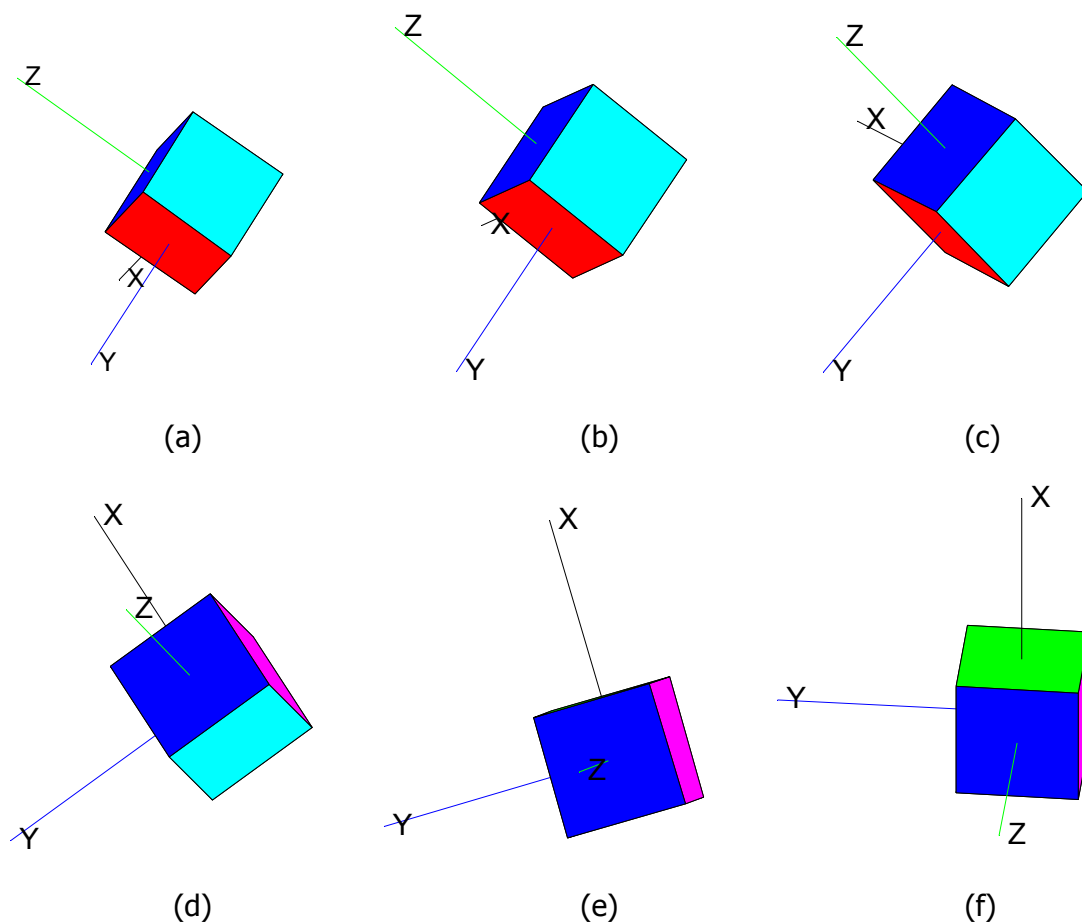
**Figura 4-9: Perfil Trapezoidal de velocidades**

En cuanto a los límites establecidos de velocidad y aceleración, se ha utilizado una velocidad inercial máxima de 60°/s y una aceleración máxima de 50°/s<sup>2</sup>.

### 4.5.3 Simulaciones

Durante la implementación de este algoritmo, se ha simulado el movimiento del sistema móvil partiendo desde una orientación arbitraria. El objetivo de las simulaciones, consiste en apuntar el eje X a una dirección referenciada al sistema inercial  $Ox_uY_uZ_u$ .

Se ha documentado uno de los múltiples movimientos, para una referencia inercial de apuntamiento  $X^T_{REF_U} = [0 \ 0 \ 1]$  (referencia de apuntamiento vertical para el eje X), obteniendo la siguiente secuencia de movimientos:

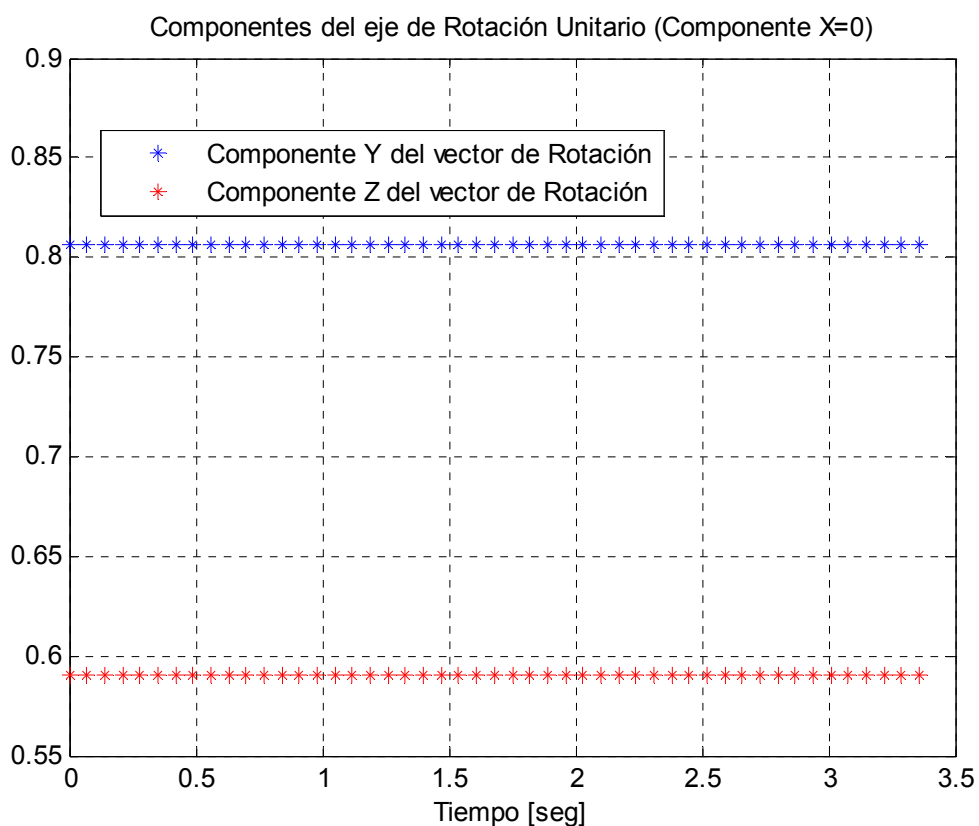


**Figura 4-10: Movimiento Inercial  $X^T_{REF_U} = [0 \ 0 \ 1]$  con periodo de muestreo 0.56seg.**

Para observar la secuencia completa, se ofrece la posibilidad de visualizar un video de dichos movimientos [movimiento9.avi](#).

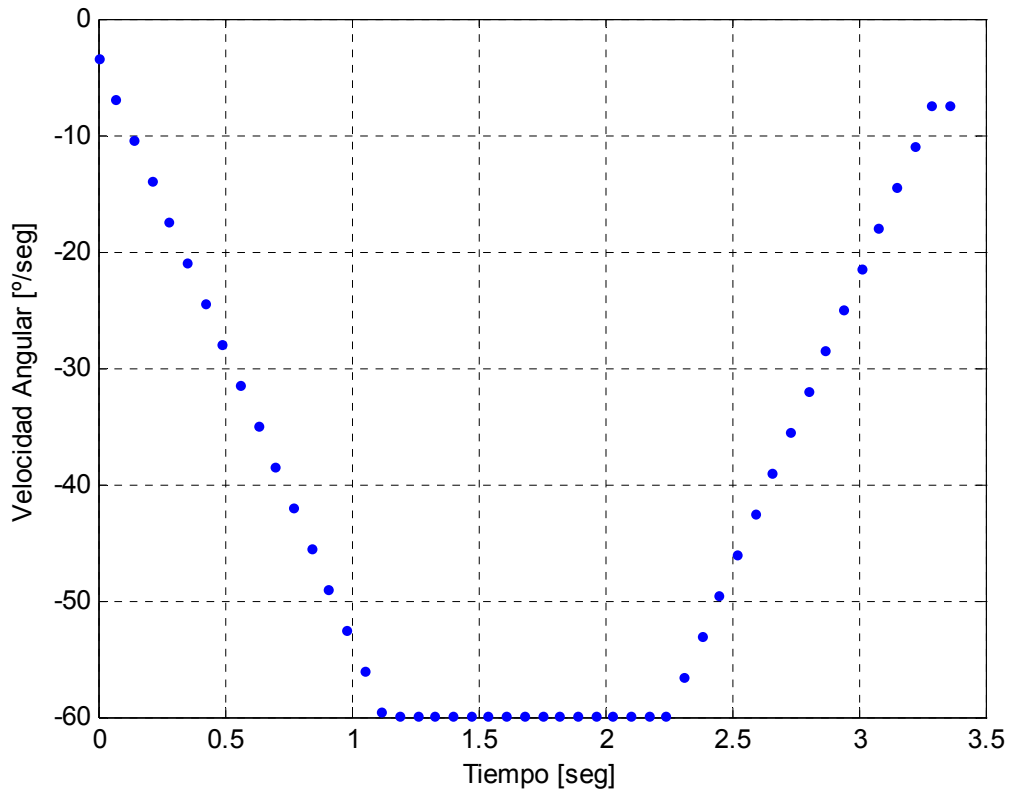


Una de las incógnitas que surgieron durante el desarrollo de este algoritmo, fue, si el vector de rotación sufre alguna modificación en alguno de sus componentes durante el transcurso del movimiento. Tras implementar y probar el algoritmo, se solventó dicha duda, concluyendo que el movimiento para conseguir la orientación objetivo, se realiza desplazando el eje X a través de un único plano perpendicular al eje de rotación. Es decir, el eje de rotación, no varía durante todo el movimiento. En la Figura 4-11 se representan las componentes del eje de rotación, cuya forma invariante verifica las conjeturas de este párrafo.



**Figura 4-11: Componentes del Eje de Rotación**

En cuanto a las velocidades angulares, en la Figura 4-12 se representa la proyección de la velocidad angular sobre el plano de movimiento. En esta figura, se observan los resultados del prefiltro de velocidad y aceleración definido en el Párrafo 4.5.2. Como se comentó previamente, se fijó un límite en velocidad máxima de  $60^{\circ}/s$  y un límite máximo de aceleración de  $50^{\circ}/s^2$ .



**Figura 4-12: Velocidad Angular Proyectada en el plano del movimiento**

Otras secuencias de movimientos capturados durante el desarrollo del trabajo, también pueden visualizarse: [movimiento5.avi](#); [movimiento6.avi](#); [movimiento7.avi](#); [movimiento8.avi](#).

Como conclusiones principales, se puede decir que se ha obtenido un algoritmo de integración inercial muy potente desde el punto de vista computacional. Se ha conseguido reducir la integración tridimensional, a una simple multiplicación de matrices en cada periodo de muestreo. Otra ventaja del método utilizado, es la generalidad del algoritmo, que ofrece la orientación del sistema móvil sin simplificación alguna. Durante el desarrollo, no se han descuidado problemas como el cálculo numérico, análisis de errores y derivas angulares. También se ha ofrecido una

solución óptima, al seleccionar precisiones numéricas y la frecuencia de aplicación del método.

En cuanto al algoritmo de posicionamiento, se ha desarrollado una estrategia para apuntar el sistema de referencia móvil, optimizando la trayectoria tridimensional y el tiempo en alcanzar la orientación de referencia. Se ha incluido un prefiltro en la señal de referencia, para acotar las velocidades y aceleraciones máximas.

## 5. ESTABILIZACIÓN DE IMAGEN MEDIANTE FUSIÓN DE SENSORES

Técnicas que integran información de diferentes sensores son muy utilizadas hoy día para mejorar las prestaciones de algunas aplicaciones. En este trabajo se pretende utilizar la información inercial de un giróscopo para estabilizar una secuencia de imágenes.

La estabilización de la imagen se consigue corrigiendo en el plano de la imagen la posición de cada uno de las imágenes de la secuencia de video. A continuación se enumeran los pasos seguidos durante el desarrollo del algoritmo para conseguir la estabilización de la imagen:

1. Sincronización de sensores: en este paso se sincronizan las lecturas del sensor inercial con la captura de imágenes.
2. Periodo de muestreo: se requiere una sincronización entre los sensores de movimiento y el periodo de adquisición de video.
3. Integración de velocidades inerciales: tal y como se expone en el capítulo 4 de odometría, se obtienen los ángulos de orientación de la cámara en cada instante (para cada frame).
4. Estimación del FOV (Field Of View): en este punto del algoritmo, se estima el campo de visión de la cámara.
5. Cálculo del desplazamiento: se calcula el desplazamiento de la imagen en píxeles dependiendo de los datos de posición, campo de visión de la cámara y resolución de la imagen.

Otros factores a tener en cuenta para el filtrado de la imagen utilizando información inercial son:

- Precisión de los elementos de medida.
- Deriva de los sensores inerciales.
- Calibración de la velocidad y aceleración del sensor inercial.
- Adaptación del campo de visión a las unidades angulares. Para utilizar este método de estabilización de imagen debe calibrarse la relación entre el campo de visión de la cámara y el movimiento detectado por los sensores angulares.

## 6. RESULTADOS EXPERIMENTALES DE IMÁGENES EN MOVIMIENTO

### 6.1 Estabilización de la Imagen

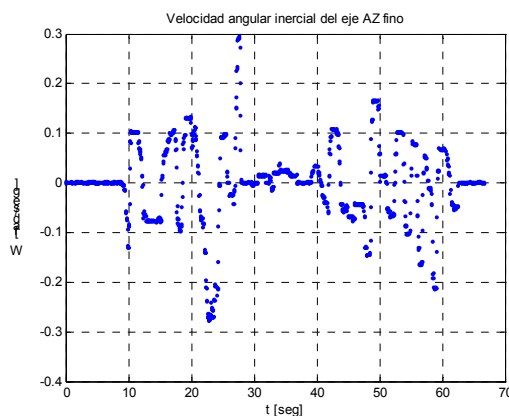
Para probar el algoritmo de estabilización, se ha capturado una secuencia de video en movimiento que incluyen a su vez objetos en movimiento.



**Figura 6-1: Video Capturado: video\_a**

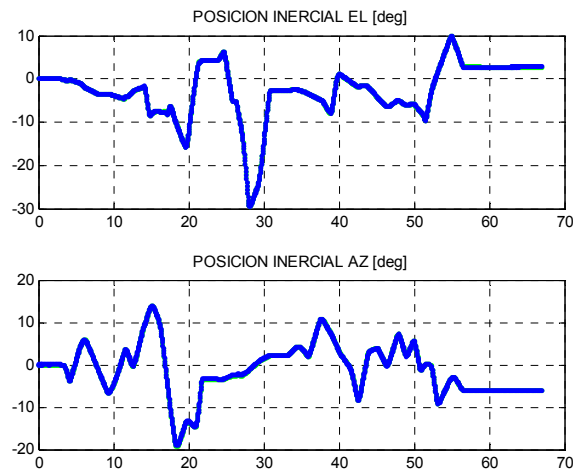
Dichas secuencia de imágenes se puede visualizar en el video: [video\\_a.mpg](#).

Durante la captura de la secuencia de imágenes, también se han capturado las velocidades inerciales de la cámara para realizar el cálculo de su orientación. En la siguiente figura se representa dicha velocidad en el eje de orientación.



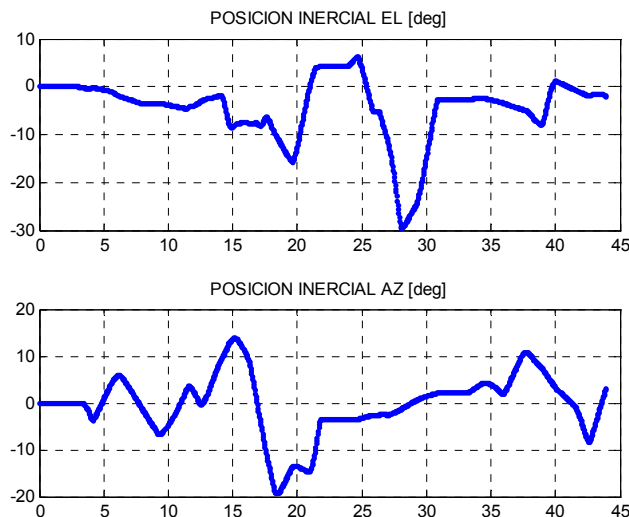
**Figura 6-2: Velocidad Inercial de Orientación de la Cámara**

El siguiente paso del algoritmo, está basado en el cálculo de la orientación de la cámara utilizando las técnicas de odometría definidas en el capítulo 4. En esta secuencia representada en grados, se aprecia como el movimiento de la cámara es aleatorio y con amplitudes considerables.



**Figura 6-3: Velocidad Inercial Integrada por Odometría**

Posteriormente, se procede a la sincronización y re-muestreo de las señales de posición, con el objetivo de preparar las secuencias de vídeo y señales para un procesamiento posterior.



**Figura 6-4: Velocidad Inercial sincronizada**

El resultado de aplicar el algoritmo de estabilización puede visualizarse en la siguiente secuencia de video: [out\\_video7.mpg](#).

### 6.2 Detección de objetos basada en diferenciación

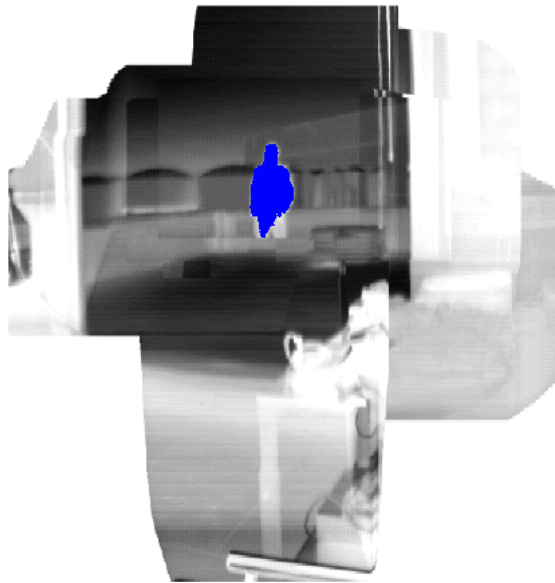
Una vez aplicado el algoritmo de estabilización, estamos en condiciones de aplicar las funciones de detección de movimiento. En la siguiente secuencia de imágenes se muestra el algoritmo aplicado al video inicial: [diff\\_video\\_local\\_3.mpg](#).



**Figura 6-5: Detección basada en diferenciación con imágenes en movimiento**

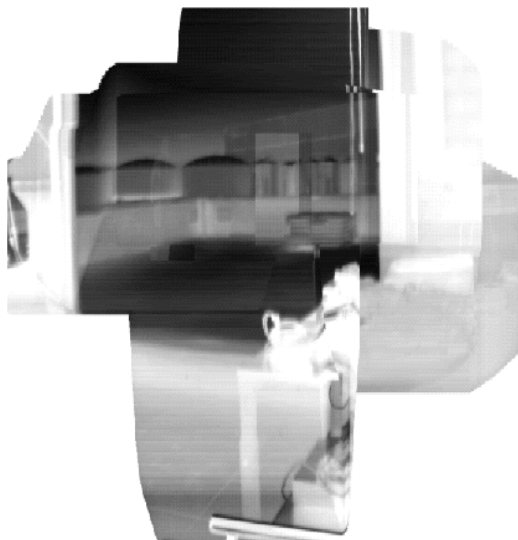
### 6.3 Detección de objetos basados en algoritmo de background

Establecida la base del algoritmo de diferenciación anterior, se puede aplicar un algoritmo de background para detección de los objetos en movimiento. En la siguiente secuencia de imágenes se muestra el algoritmo aplicado al video estabilizado: [background\\_video\\_c\\_5.mpg](#).



**Figura 6-6: Detección basada en el background con imágenes en movimiento**

También se muestra el background calculado después de 100 iteraciones del algoritmo:



**Figura 6-7: Background generado**



### **CONCLUSIONES**

Como conclusiones del presente trabajo, se ha realizado una búsqueda de información y artículos publicados para conocer el estado del arte en el ámbito de la detección de movimiento.

Se ha desarrollado algoritmos para la detección de objetos en movimiento en secuencias de imágenes en movimiento.

En cuanto a los artículos consultados, muestran información muy variada sobre los tipos y aplicaciones en el ámbito de la detección de movimiento. Principalmente, ofrecen pautas importantes para la implementación de algoritmos robustos que funcionen con imágenes reales.

Sobre las rutinas implementadas, la diferenciación básica se ha dotado de bastante robustez al ruido de las imágenes y a otras características de las imágenes utilizadas. Además, dicho algoritmo es una excelente base de búsqueda para el resto de funciones desarrolladas.

El cálculo del background y la diferenciación por umbral, cubren las carencias de la detección incompleta de los objetos y del problema de detección para velocidades bajas. Se puede concluir que el algoritmo de diferenciación básica junto al algoritmo de umbral utilizando fondos, es una herramienta aceptable para muchas aplicaciones de detección de objetos móviles en cámaras fijas.

En cuanto al algoritmo de estabilización desarrollado, es la parte que ofrece mayor valor añadido a los algoritmos de procesado de imagen.

## **ANEXOS**

En este apartado se enumera la documentación adjunta a la memoria:

- deteccion\_dinamica\_0B.m; Background\_0f.m; diferenciación\_0d: Código Matlab de los algoritmos de detección de movimiento implementados.
- Artículos publicados utilizados como base del presente estudio.
- Videos capturados con una cámara infrarroja, utilizados para el desarrollo de los algoritmos de estabilización y seguimiento.

## **BIBLIOGRAFÍA**

- [1] Image difference threshold strategies and shadow detection. Paul L. Rosin and Tim Ellis.
- [2] Detecting Moving Objects using a single Camera on a Mobile Robot in an Outdoor Environment. Boyoon Jung and Gaurav S. Sukhatme. Los Angeles 2004.
- [3] An a contrario Decision Framework for Region-Based Motion Detection. T. Veit, F. Cao and P. Bouthemy. 2005.
- [4] Moving object recognition using an adaptive background memory. K.P. Karmann and A. Von Brandt. 1990.
- [5] Stationary background generation: An alternative to the difference of two images. W. Long and Y.H Yang. 1990.
- [6] J. J. Craig (1989). Introduction to robotics, mechanics and control (2nd Ed.). Addison-Wesley, Reading, MA.
- [7] Richard L. Burden. Análisis Numérico. Grupo Editorial Iberoamérica.
- [8] D. H. Titterton. Strapdown Inertial navigation Technology (2nd Ed.). AIAA 2004.
- [9] Zhigang Zhu titulado. Digital Video Sequence Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering.

Fin del documento