



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

**Técnicas de procesamiento de imagen para el
seguimiento de objetos desde vehículos
aéreos no tripulados**

Autor:

Natalia Ibáñez Sáez

Tutores:

Pablo Casaseca de la Higuera

Santiago Aja Fernández

Valladolid, 8 de Septiembre de 2015

TÍTULO: **Técnicas de procesamiento de imagen para el seguimiento de objetos desde vehículos aéreos no tripulados**

AUTOR: **Natalia Ibáñez Sáez**

TUTOR: **D. Pablo Casaseca de la Higuera**
D. Santiago Aja Fernández

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **D. Marcos Martín Fernández**

VOCAL: **D. Rodrigo de Luis García**

SECRETARIO **D. Santiago Aja Fernández**

SUPLENTE **D. Federico Simmross Wattenberg**

SUPLENTE **D. Carlos Alberola López**

FECHA: **14 de Septiembre de 2015**

CALIFICACIÓN:

Resumen

Este proyecto ha sido realizado dentro del grupo "Laboratorio de Procesado de Imagen" (LPI) de la Universidad de Valladolid en colaboración con la Universidad del Oeste de Escocia (University of the West of Scotland) enmarcado en un proyecto de investigación sobre video vigilancia a través de vehículos aéreos no tripulados.

El objetivo principal de este proyecto ha consistido en realizar una herramienta de procesamiento de imagen capaz de realizar la detección y seguimiento de objetos en movimiento grabados desde vehículos aéreos no tripulados. El problema de realizar video vigilancia desde cámaras móviles, como sucede en el caso de vehículos aéreos no tripulados, supone retos adicionales a los existentes en situaciones en las que la cámara está fija. Por ello, varios de los métodos utilizados en el caso de cámara fija no son aplicables cuando la cámara está en movimiento. En el estado del arte se han analizado varios de los métodos utilizados en el problema de video vigilancia con cámara móvil y se ha seleccionado el método basado en alineamiento de la imagen mediante puntos característicos basado en flujo óptico por ser un método que proporciona buenos resultados y cuyo tiempo de ejecución es relativamente bajo.

El método seleccionado para realizar la herramienta deseada tiene como parte fundamental la comparación de dos flujos ópticos: un flujo óptico real, que describe el movimiento que se produce en los puntos seleccionados y un flujo óptico artificial, que describe el movimiento global de la imagen y que corresponde al movimiento del fondo de la imagen. Mediante la comparación de ambos flujos es posible compensar el movimiento producido por la cámara móvil y que de este modo, todo el movimiento que recoja la cámara corresponda a los objetos en movimiento. Así mismo, es fundamental realizar un posterior seguimiento de los objetos móviles que permite realizar un filtrado espacial y temporal de los mismos, eliminando posibles falsas detecciones de objetos móviles producidos por errores en la fase de detección. Este seguimiento proporciona también información sobre la trayectoria de los objetos y se realiza mediante el filtro de Kalman.

Por último, la herramienta realizada se ha probado con dos bancos de vídeos diferentes. El primero de ellos corresponde a una serie de vídeos grabados con una cámara de fotos en distintas situaciones de iluminación, movimiento de la cámara y en presencia de diferente número de objetos. Estos vídeos han sido utilizados para fijar los parámetros de funcionamiento de la herramienta y realizar un análisis cuantitativo que refleje la influencia de las características del vídeo en el funcionamiento de la herramienta así como el tiempo que tarda la herramienta en inicializarse. El segundo banco de vídeos consiste en vídeos grabados por un vehículo aéreo no tripulado controlado de forma teledirigida. Estos vídeos han sido utilizados para evaluar cualitativamente el buen funcionamiento de la herramienta a través de la comprobación visual de los vídeos procesados.

Palabras clave

Video vigilancia, vehículo aéreo no tripulado, alineamiento por características, flujo óptico, seguimiento, filtro de Kalman.

Abstract

This project has been performed within the "Laboratory of Image Processing" (LPI), research group at the University of Valladolid in collaboration with the University of the West of Scotland. The project has been developed in the context of a project of a research project focused on video surveillance carried out by unmanned aerial vehicles.

The aim of this project has been the development an image processing tool that performs the detection and tracking of moving objects in videos recorded from unmanned aerial vehicles. The problem of carrying out video surveillance from moving cameras, e.g., those attached to unmanned aerial vehicles, poses additional challenges to those related with video surveillance from non-moving cameras. Therefore, many of the methods used in the case of non-moving cameras are not applicable with moving ones. A review of the state of art analyses several methods that used to solve the video surveillance problem from moving cameras. Out of them, a method based on image feature alignment through optical flow has been selected because of its good results and its low running time.

The method that has been selected to implement the tool is based on the idea of the comparison of two optical flows: a real optical flow that describes the movement of the selected points and an artificial flow that describes the global movement of the image, which corresponds to the image background. The comparison of both flows allows to eliminate the movement caused by the moving camera and therefore, the remaining movement corresponds to moving objects. Another fundamental problem consist in tracking the detected object in order to filter them spatially and temporally to eliminate moving objects caused by false detections. This tracking process also gives information about the trajectory of the moving objects and is carried out with a Kalman filter.

Finally, the implemented tool has been tested with two different group of videos. The first of them corresponds to a group of videos filmed by a video camera in different illumination contexts, with various camera movements and with different number of objects present. These videos have been used to fix the parameters of the tool and to qualitatively test the performance of the tool with videos acquired with different illumination levels. The second group of videos consists of videos filmed from an unmanned aerial vehicle which is remote controlled. This has been used to test the operation of the tool by watching the results in the processed video.

Keywords

Video surveillance, unmanned aerial vehicle, feature alignment, optical flow, tracking, Kalman filter.

Agradecimientos

Este proyecto está dedicado a Pablo por haberme dejado formar parte de este proyecto y haberme ayudado a llevarlo a cabo; a Santi, por encontrarme un proyecto que me gustara tanto.

A mis amigos, Julia, Jorge, Belén, David, Javi, Marta, Nadia, Marina, Óscar y tantos otros, con los que estos años de universidad han sido super guays.

A Jose, ese teleco en la distancia y compañero de mus, que me dio la idea de este proyecto y me dio ánimos en la recta final.

Pero sobre todo a mamá, papá y Blanca que me han ayudado con tantas cosas del proyecto y sobre todo a darme ánimos. Y porque son tan buenos.

Índice general

Índice de figuras

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Metodología	2
1.4. Herramientas	3
1.5. Estructura	3
2. Estado del arte	5
2.1. Visión global de los sistemas de video vigilancia	5
2.2. Clasificación de los sistemas de video vigilancia	5
2.3. Aplicaciones de los sistemas de video	6
2.4. Técnicas de procesamiento de imagen para la detección y seguimiento de objetos desde cámaras en movimiento	9
2.4.1. Fase de extracción de puntos	9
2.4.2. Fase de determinación de correspondencia entre puntos	11
2.4.3. Fase de alineamiento de la imagen	12
2.4.4. Fase de segmentación de movimiento	14
2.4.5. Fase de definición de objetos	14
2.4.6. Fase de seguimiento de objetos	16
2.5. Técnicas de procesamiento de imagen para la detección y seguimiento de objetos desde cámaras en movimiento utilizadas en este proyecto	18
3. Diseño del método experimental	19
3.1. Introducción	19
3.2. Visión global del método experimental	20
3.3. Extracción de puntos: Algoritmo SURF.....	23
3.4. Correspondencia entre puntos: Algoritmo KLT	28
3.5. Alineamiento de la imagen: Cálculo de flujos ópticos real y artificial. Tipos de transformaciones. Algoritmo MSAC	32
3.6. Segmentación del movimiento: Comparación de flujos ópticos real y artificial ..	38
3.7. Agrupamiento de píxeles dinámicos: Algoritmo K-means	39
3.8. Seguimiento de objetos: Algoritmo de seguimiento basado en componentes conectados. Filtro de Kalman	42
4. Pruebas y resultados experimentales	52
4.1. Bases de datos utilizadas	52
4.2. Descripción de las pruebas experimentales	52
4.2.1. Análisis cualitativo del funcionamiento de la herramienta	53
4.2.2. Análisis cuantitativo de la mejora que supone el método de comparación de flujos ópticos	57

5. Conclusiones, trabajo futuro y presupuesto	62
5.1. Conclusiones	62
5.2. Trabajo futuro	64
5.3. Presupuesto económico	65
Referencias	66

Índice de figuras

Figura 3.1: Fases que componen el algoritmo diseñado para la detección y seguimiento de objetos en movimiento	22
Figura 3.2: Diagrama de bloques del proceso de detección y seguimiento de objetos	23
Figura 3.3: Esquema de la relación existente entre las octavas y la escala durante el proceso de detección de puntos de interés.....	24
Figura 3.4: Funciones ondulatorias de Haar.	26
Figura 3.5: Subregiones en las que se divide la vecindad del píxel para realizar el cálculo de su orientación en el algoritmo SURF.....	27
Figura 3.6: Ejemplo de agrupamiento divisivo mediante k-medias en dos agrupaciones.....	42
Figura 3.7. Esquema cíclico del funcionamiento del algoritmo del filtro de Kalman.....	50
Figura 3.8. Imagen completa de la operación del filtro de Kalman.....	51
Figura 4.1: Representación de cuatro cuadros consecutivos donde se ha producido el seguimiento de un objeto con cámara fija.....	54
Figura 4.2: Representación de cuatro cuadros consecutivos donde se ha producido el seguimiento de un objeto con cámara en movimiento.....	55
Figura 4.3: Representación de cuatro cuadros consecutivos. Los tres primeros representan la detección de un falso positivo y el cuarto su corrección mediante la etapa de filtrado.....	56
Figura 4.4: Representación de dos cuadros de un vídeo grabado por un UAV con objetos en movimiento a una distancia muy corta de éstos.	57
Figura 4.5: Gráfica comparativa del rendimiento de la herramienta para vídeos grabados con cámara en movimiento y objetos en movimiento.....	61
Figura 4.6: Gráfica comparativa del rendimiento de la herramienta para vídeos grabados con cámara en movimiento y objetos en movimiento: Resultados correspondientes a la tasa de detección y la tasa de falsas alarmas.....	61

Capítulo 1: Introducción

1.1. Motivación

El seguimiento de objetos ha sido desde hace tiempo un tema que ha suscitado gran interés debido las numerosas aplicaciones a las que puede ser destinado. Sus orígenes han sido en la mayoría de los casos militares puesto que la tarea de identificar objetos en movimiento suponía una forma de conocer con anterioridad posibles ataques y poderse prevenir de estos.

El hecho de poder hacerlo además sin intervención humana directa, trae consigo numerosas ventajas: En primer lugar se realiza de forma mecánica, de modo que se puede liberar a las personas del esfuerzo que supone tener que observar durante largo tiempo los lugares de interés. Además, de esta forma puede obtenerse la información de movimiento de forma estadística, lo que evita posibles errores humanos que podrían producirse si la supervisión tuviera que llevarla a cabo una persona. Y por último, también consigue alejar a las personas de peligros que puedan acarrear las labores de vigilancia así como seguir movimiento en lugares que no sean accesibles a las personas.

Para ello llevar a cabo el seguimiento de objetos se necesita instalar una cámara en el lugar de interés, y para que la tarea se pueda llevar a cabo de forma automática, basta con procesar los vídeos que se tomen, y mediante algoritmos matemáticos llevar a cabo el seguimiento de las personas.

El abanico de aplicaciones es muy amplio y está en constante desarrollo, y entre ellas se encuentran tanto aplicaciones militares como aplicaciones civiles. Dentro de las aplicaciones civiles la diversidad aumenta cada vez más y algunas de sus aplicaciones más importantes son la detección de intrusos en lugares privados, la supervisión del tráfico o las labores de búsqueda y rescate.

Existen dos formas de llevar a cabo el proceso de seguimiento de objetos mediante el procesamiento de vídeos: La primera de ella está basada en cámara estática y la segunda en cámara dinámica. Las ventajas que supone realizar el seguimiento con cámara dinámica son que el seguimiento se puede llevar a cabo en lugares de difícil acceso y que el objetivo puede ser seguido mediante la cámara para adaptarse a su movimiento y así poder seguir su trayectoria durante mayor espacio. Además se puede modificar la altura de la cámara para así poder o bien apreciar un determinado objeto con más detalle, o bien poder abarcar un espacio más amplio donde tengan cabida más objetos.

Sin embargo, esta técnica presenta el problema de tener que poner una cámara en movimiento. Una de las posibles soluciones, y con la que en la actualidad se está experimentando bastante, consiste en la utilización de vehículos aéreos no tripulados (Unmanned Aerial Vehicles, UAVs). Estos sistemas son aeronaves que no tienen tripulación y pueden ser controladas o bien de forma teledirigida o bien mediante mecanismos de inteligencia artificial. De esta forma, mediante UAVs se puede disponer de un sistema de cámara en movimiento que permita el seguimiento de objetos.

No obstante, son varios los retos que presentan las cámaras en movimiento instaladas en UAV. El movimiento, en ocasiones no controlado del vehículo, debido a agentes externos como el viento o vibraciones del propio UAV son novedades que presentan estos sistemas y hacen que las técnicas utilizadas en las situaciones de cámara estática ya no resulten válidas y haya que adaptar los algoritmos a esta situación concreta debido a que los vídeos tomados con cámaras en movimiento introducen movimiento debido al desplazamiento de la cámara que se mezcla con el movimiento propio del objeto que se desea detectar.

1.2. Objetivos

El principal objetivo de este trabajo consiste en realizar una herramienta capaz de procesar imágenes para el seguimiento de objetos desde vehículos aéreos no tripulados (UAVs). Al tratarse de un campo que presenta retos adicionales a los que presentaba el seguimiento de objetos desde cámaras fijas, la implementación de este tipo de sistemas involucra la resolución de diferentes problemas con requisitos computacionales importantes. Este proyecto pretende ser una primera aproximación a la creación de una herramienta, que sirva como base para realizar herramientas más potentes y versátiles. Por ello, el trabajo se ha enfocado al desarrollo de una herramienta orientado al objetivo mencionado sin una optimización específica con respecto al coste computacional. La implementación posterior de la herramienta para procesado a bordo requerirá dicha optimización, escapándose del alcance del presente trabajo.

Para comprobar el funcionamiento de la herramienta creada, se pretende también una comprobación principal que consiste en determinar de forma cualitativa, mediante su ejecución sobre diversos vídeos y su comprobación visual.

Se pretende además, obtener un método apropiado para realizar el seguimiento de objetos desde UAVs. Para ello se llevará a cabo un estudio de un método basado en comparación de flujos ópticos. Concretamente se realizará la comparación entre el flujo óptico real, que recoge el movimiento total de la imagen y el flujo óptico artificial, que recoge el movimiento producido por el desplazamiento de la cámara. El objetivo que se persigue en este sentido, consiste en realizar un análisis cuantitativo del rendimiento que confiere a la herramienta el método de comparación de flujos en comparación con una herramienta que no lo implemente.

1.3. Metodología

Para llevar a cabo el estudio del procesado de imagen para el seguimiento de objetos desde vehículos aéreos no tripulados se llevará a cabo la siguiente metodología:

En primer lugar, se realizará una revisión bibliográfica que permita dar una visión global sobre el estado del arte de los sistemas de video vigilancia. En concreto se pretende profundizar en aquellos métodos y sistemas que permitan realizar las tareas de seguimiento a través de cámaras móviles. Se estudiarán, por tanto, las técnicas que se han utilizado para llevar a cabo tal propósito y se seleccionará la que mejor se adapte al escenario deseado: el seguimiento mediante vehículos aéreos no tripulados controlados de forma teledirigida.

En segundo lugar, se procederá a la obtención de los vídeos que servirán como elemento a analizar mediante el programa de detección y seguimiento creado para tal efecto. Para ello se tomarán dos bancos de vídeos: en primer lugar videos simples que no serán grabados por el UAV, y en segundo lugar vídeos grabado por el UAV.

En tercer lugar, se llevará a cabo el desarrollo del programa de procesamiento de los vídeos. El programa desarrollado, se pretende que sea capaz de analizar los vídeos proporcionados y realizar en ellos el seguimiento de objetos en movimiento. Esta fase será la principal del proyecto y por tanto la más costosa y utilizará las técnicas de procesamiento de imagen seleccionadas anteriormente que incluye una fase de detección de los objetos en movimiento y una fase de seguimiento de los mimos. La fase de detección de objetos está basada en la técnica de comparación de flujos ópticos real y artificial y la fase de seguimiento se realiza con ayuda del filtro de Kalman.

Y por último, en cuarto lugar, se realizará un análisis de los resultados obtenidos y un estudio de las condiciones que afectan al UAV que hacen el programa desarrollado sea más o menos preciso en el seguimiento de objetos en movimiento. En concreto, se llevará a cabo un estudio de cómo afecta al funcionamiento de la herramienta el tipo de vídeo en función de su iluminación y según el movimiento de la cámara y otro estudio en el que se ilustre cómo afecta el tiempo de inicialización de la herramienta a su correcto funcionamiento.

1.4. Herramientas

Para llevar a cabo este proyecto se necesita por un lado el uso de la herramienta Matlab versión R2013b y su biblioteca de visión artificial para la implementación del programa de procesamiento de imágenes para detección de objetos en movimiento y por otro lado el uso de un vehículo aéreo no tripulado que se controle de forma teledirigida y que permita la obtención de los vídeos que se procesan.

También se hace uso de un cámara de vídeo para grabar los primeros vídeos con los que se realizan las pruebas experimentales. El UAV es de la marca Parrot modelo AR Drone 2 con dimensiones 77.7 x 38.3 x 12.5mm y con un peso de 31g. La cámara de vídeo es de marca Samsung y modelo Schneider Kreuznach.

1.5. Estructura

La presente memoria se estructura de la siguiente forma:

En el capítulo 2 se lleva a cabo una descripción del estado del arte de los sistemas que permiten el seguimiento de objetos, exponiendo en primer lugar los tipos de sistemas de video vigilancia; en segundo lugar las aplicaciones que proporcionan estos sistemas; y por último las técnicas de procesamiento de imagen para la detección y seguimiento de objetos desde cámaras en movimiento, de las que se especificará cuáles se han seguido en este proyecto.

En el capítulo 3 se explica el diseño del método experimental que se llevará a cabo y los diferentes fundamentos teóricos sobre los que se asienta. Se describen todas las fases de las

que consta el método de procesamiento, incluyendo los algoritmos utilizados y las decisiones tomadas en cuanto a umbralización.

En el capítulo 4 se presentan las pruebas realizadas para la caracterización de la herramienta y los resultados obtenidos mediante los experimentos realizados. Se realiza un análisis de dichos resultados para reflejar la mejora que supone la implementación del método de comparación de flujos ópticos real y artificial para el buen funcionamiento de la herramienta.

Por último, en el capítulo 5 se presentan las conclusiones obtenidas a partir del presente trabajo y se estudian posibles proyectos futuros que podrían ser de interés como extensión a este proyecto. Se presenta también un presupuesto del coste que ha supuesto realizar este proyecto.

Capítulo 2: Estado del arte

2.1. Visión global de los sistemas de video vigilancia

Los sistemas de video vigilancia comprenden todas las actividades que requieran la colocación de una o varias cámaras de grabación, fijas o móviles cuyo objetivo consiste en monitorizar el comportamiento o actividades de un espacio o personas [1-2].

Un caso particular de sistemas de video vigilancia muy importante debido a la gran cantidad de posibilidades que ofrece son los sistemas de video vigilancia inteligente. El objetivo de estos sistemas consiste en la monitorización en tiempo real de objetos en un escenario determinado y su correspondiente interpretación automática para la interpretación de los escenarios y la comprensión y predicción de acciones e interacciones en los objetos observados a partir de la información tomada por los sensores.

Las siguientes secciones presentan una descripción de los sistemas de video vigilancia, sus aplicaciones y las técnicas de procesamiento de imagen para video vigilancia a partir de cámaras en movimiento.

2.2. Clasificación de los sistemas de video vigilancia

Los sistemas de video vigilancia consisten en cámaras fijas o móviles que tienen como objetivo grabar el comportamiento de objetos o personas que se produce en la zona sobre la que tienen campo de visión. Estos sistemas pueden constar de una o varias cámaras conectados a pantallas desde las que se pueden observar las grabaciones. Los sistemas de video vigilancia se pueden clasificar atendiendo a criterios como los siguientes [1-2]:

- **Tipo de sensor:** Infrarrojo, audio o vídeo.
- **Multiplicidad del sensor:** Monocular o estéreo.
- **Emplazamiento del sensor:** Centralizado o distribuido.
- **Tecnología del sistema de video vigilancia:** La tecnología del sistema de video vigilancia está condicionada por la generación a la que pertenece el sistema. Existen las siguientes tres generaciones [1-2]:
 - La **primera generación** consiste en sistemas de circuito cerrado de televisión (CCTV) que consta de un conjunto de elementos enlazados entre sí. Todos los elementos son analógicos y suelen consistir en varias cámaras distribuidas que envían la información obtenida generalmente a una sala central desde donde se monitoriza a través de uno o varios equipos. Esta generación se caracteriza porque no incluye ningún dispositivo digital que ayude al estudio de las situaciones que se producen, de modo que se hace

necesaria la supervisión humana. Por tanto, se trata de un sistema que no puede enviar alertas por sí solo y no puede realizar tratamiento, distribución y almacenamiento de las imágenes de forma digital.

- La **segunda generación** supone una mejora a la generación anterior puesto que combinan CCTV analógicos con sistemas digitales que permiten vigilancia IP. La ventaja de esos sistemas frente a los anteriores consiste en que reduce la necesidad de presencia humana para la supervisión puesto que estos sistemas son capaces de interpretar automáticamente varios de los eventos que se producen. De esta forma se aumenta la eficiencia puesto que es posible analizar un mayor número de situaciones al mismo tiempo. Sin embargo, estos sistemas son menos robustos puesto que la tasa de error aumenta como resultado de falsas alarmas u objetos no detectados.
- La **tercera generación** se caracteriza por ser sistemas altamente distribuidos y cuyos sensores son más modernos, lo cual supone una mayor eficiencia haciendo posible que se pueda agilizar el cálculo en gran medida puesto que en este caso, el procesamiento se puede realizar en parte de forma local, lo cual evita la existencia de un nodo central con gran carga de procesamiento. De esta forma se consigue que se pueda realizar vigilancia en tiempo real. Además, estos sistemas son más robustos, puesto que al evitar la centralización del procesamiento, el fallo de un nodo no afecta al funcionamiento global. No obstante, la comunicación entre los distintos dispositivos resulta en ocasiones complicada debido a la heterogeneidad de los dispositivos.
- **Movilidad de la cámara:** Cámara fija o móvil.

2.3. Aplicaciones de los sistemas de video vigilancia

Las aplicaciones a las que se pueden destinar los sistemas de video vigilancia son muy diversas, pero todas ellas podrían clasificarse en tres grandes grupos: aplicaciones basadas en píxel, aplicaciones basadas en objeto y aplicaciones especializadas [3].

- Las aplicaciones **basadas en píxel**, se caracterizan por tomar sus decisiones en función de los cambios existentes entre los píxeles de cuadros de vídeo distintos. No se tiene en cuenta, por tanto el tipo de objeto que se esté moviendo, sino simplemente si existe movimiento o no. Las aplicaciones más frecuentes pertenecientes a este grupo son:
 - **Detección de movimiento**
Las aplicaciones basadas en detección de movimiento [4] tienen como objetivo la determinación del movimiento que existe en una secuencia de imágenes sin prestar atención a la identificación del objeto que se mueve. Sin embargo, también puede utilizarse la detección de movimiento como fase previa a la identificación de los objetos que se mueven. La detección de movimiento para cámaras fijas pasa por obtener y actualizar automáticamente las imágenes consideradas fondo o sin movimiento mediante una secuencia de imágenes.

- **Detección de manipulación de cámaras y mejora de imagen**
 Las aplicaciones de manipulación de cámaras [5] y mejora de imagen [6] realizan un procesamiento de las imágenes basado en píxel puesto que su objetivo no consiste en detectar o identificar objetos. En el caso de las aplicaciones de detección de manipulación de cámaras se pretende detectar rastros de que una imagen ha sido alterada. Para ello se lleva a cabo un estudio estadístico destinado a identificar los cambios asociados a manipulación de la secuencia de imágenes que un vídeo. Por su parte, las aplicaciones basadas en mejora de imagen que aplican procesamiento puntual por píxel tienen como objetivo mejorar el contraste tonal de la imagen mediante operaciones matemáticas que actúan en cada uno de los píxeles que se considera necesario retocar.

- Las aplicaciones **basadas en objeto** utilizan la caracterización e identificación de objetos. Estas aplicaciones suelen seguir tres fases: En primer lugar se comparan los píxeles del cuadro con un cuadro de referencia, después se extraen los posibles objetos de la imagen y se les asignan identificadores, y por último se clasifican en tipos de objetos. Las aplicaciones más destacables pertenecientes a este grupo son las que se presentan a continuación:
 - **Detección de intrusos:**
 Este tipo de video vigilancia consiste en detectar accesos no autorizados a un determinado escenario ya sea porque el escenario es peligroso, como sucede en las centrales nucleares o porque la persona está intentando acceder al recinto con fines maliciosos, como puede ser la entrada no autorizada a un banco. La detección de intrusos se inicia con la detección de anomalías en el vídeo que se procesa y el posterior análisis estadístico que de búsqueda de patrones que indiquen una actividad anormal en el recinto [1] [3].

 - **Contabilización de objetos o personas:**
 Este tipo de video vigilancia tiene el objetivo de contabilizar las personas u objetos que aparecen en algún momento en un determinado escenario. Las aplicaciones de este tipo son útiles para supervisión de tráfico, contabilización de personas en zonas video vigiladas y procesos de fabricación. El proceso de contabilización de objetos se realiza analizando el vídeo y buscando en éste objetos que coincidan con unos determinados patrones o criterios preestablecidos [1] [3].

 - **Seguimiento de objetos:**
 El seguimiento de objetos es un tipo de video vigilancia que consta de dos fases: en primer lugar hay que detectar objetos en el escenario vigilado y en segundo lugar hay que seguir la trayectoria de dicho objeto. Para ello se necesita predecir su ubicación en el futuro a partir de las observaciones que se hayan hecho del mismo en tiempos pasados y presente. Una vez se estime la trayectoria futura, se puede optar también, en caso de que la cámara sea móvil, por que ésta siga físicamente al objeto y acople su trayectoria y velocidad a la del objeto seguido. Existen diversas aplicaciones para el seguimiento de objetos, entre las que se encuentran la seguridad y vigilancia, la interacción persona-máquina o la realidad aumentada [1] [3].

- **Detección de objetos abandonos**

Ésta práctica consiste en localizar objetos que han sido abandonados en un escenario. Para ello se procesa el vídeo obtenido y se tratan de identificar objetos que permanezcan inmóviles en la escena pero que no pertenezcan al fondo. Para ello se utilizan técnicas de segmentación que eliminen la parte de la imagen correspondiente al fondo en combinación con un proceso de identificación que busque objetos con unas determinadas características que puedan ser los objetos abandonados [1].
- Las aplicaciones **especializadas** utilizan una combinación de píxel y objeto para tomar sus decisiones. Algunos ejemplos son:
 - **Reconocimiento de matrículas**

Un ejemplo de herramienta de reconocimiento de matrículas es la presentada por CP Ramos y DR Rodríguez [7], que consiste en un sistema que permita leer el número de matrícula de un vehículo a partir del procesamiento de vídeo. Este sistema se implementa mediante tres bloques básicos: localización de la matrícula basado en el cálculo de gradientes; umbralización y segmentación de caracteres a partir de información contenida en la imagen como es su histograma y reconocimiento de caracteres obtenidos a través de una red neuronal. Esta técnica se puede aplicar a la entrada de los aparcamientos para permitir o denegar el acceso.
 - **Reconocimiento facial**

Esta técnica consiste en reconocer un rostro, como lo haría una persona a partir de imágenes o vídeos. Para obtener un resultado muy detallado del análisis hay que llevar a cabo las siguientes cuatro fases: En primer lugar hay que detectar en la imagen un objeto que se ajuste a un patrón de tipo rostro; en segundo lugar hay que identificar la persona a la que pertenece el rostro analizado; en tercer lugar analizar las expresiones que exhibe el rostro; y por último llevar a cabo una clasificación basada en las características físicas de la cara. Este método proporciona un amplio abanico de aplicaciones entre las que se encuentran identificación la identificación criminal o la decisión de permitir o denegar el acceso de determinadas personas a un cierto lugar [8].
 - **Detección de fuego y humo**

La detección de humo y fuego [3] puede llevarse a cabo de la forma tradicional consistente en la utilización de sensores de humo que detectan la superación de un umbral por parte del humo. Sin embargo, existe una alternativa basada en el procesamiento de vídeo, que consiste en detectar la presencia de niveles altos de humo a través del análisis del color, la luz y el movimiento que presentan los cuadros del vídeo. Este sistema de detección es más rápido que los sistemas convencionales de detección de humo mediante sensor.

2.4. Técnicas de procesamiento de imagen para la detección y seguimiento de objetos desde cámaras en movimiento

El seguimiento de objetos mediante cámaras móviles presenta retos adicionales que surgen del hecho de que la cámara se encuentre en movimiento. Los métodos más habituales utilizados para el caso de detección y seguimiento de objetos mediante cámara fija no pueden ser utilizados si la cámara es móvil. De este modo, quedan descartados los métodos basados en técnica de substracción de fondo y los algoritmos de diferenciación temporal. Métodos alternativos han sido estudiados en los últimos años con el objetivo de la detección de objetos mediante cámara móvil.

Los métodos que se utilizan más habitualmente se basan en la clasificación siguiente [1]:

- Métodos que diferencian imágenes consecutivas alineadas:
 - Alineamiento global:
 - Identificación con plantilla
 - Flujo óptico
 - Modelo elástico
 - Alineamiento por características
- Métodos que detectan movimiento directamente en la imagen

Todos estos métodos pasan por unas fases comunes de segmentación de movimiento, cuyo objetivo principal es detectar los píxeles que han sufrido movimiento entre tramas consecutivas y dicho movimiento corresponde a objetos móviles y no al movimiento de la cámara, otra fase de definición de objetos en movimiento y una fase final de seguimiento de los objetos.

Algunos métodos tienen fases previas de alineamiento, cuyo objetivo consiste en eliminar el movimiento aparente de la cámara entre dos imágenes consecutivas de forma que las diferencias existentes entre ellas correspondan a los objetos móviles.

Algunos de los métodos que requieren alineamiento de imágenes requieren a su vez una fase previa de extracción de puntos característicos de forma que las operaciones de alineamiento se realicen sobre una selección de puntos de la imagen y no sobre todos los puntos de ésta.

A continuación se presentan con mayor detalle las fases que se llevan a cabo para la detección y seguimiento de los objetos en movimiento mediante cámara móvil y los diferentes algoritmos existentes para llevar a cabo cada una de estas fases.

2.4.1. Fase de extracción de puntos

Para llevar a cabo el alineamiento de las imágenes existen dos opciones. En primer lugar se puede decidir trabajar con todos los puntos de la imagen y realizar todos los posteriores cálculos sobre todos los píxeles de la imagen y en segundo lugar se puede decidir llevar a cabo una extracción de puntos que permita realizar los cálculos posteriores sobre un conjunto de puntos seleccionados y permita un alineamiento por características.

Los puntos que se extraen son puntos que se consideran relevantes en las imágenes que se pretende alinear. Estos puntos ser lo más invariantes posible frente a las variaciones entre imágenes para que la correspondencia entre puntos de imágenes consecutivas sea lo más eficaz posible.

Los algoritmos de extracción de puntos más importantes son los siguientes:

- **Harris**

El detector de Harris [9] es un detector de puntos relevantes muy utilizado puesto que es invariante ante la escala, rotación, cambios de iluminación y ruido en la imagen. Este detector se basa en el cálculo de gradientes horizontal y vertical en una ventana situada sobre cada punto de interés (x,y) de la imagen. Los puntos relevantes que proporciona como salida este algoritmo son aquellos con grandes variaciones en ambas direcciones. Los puntos que proporcionan valores elevados de los gradientes son los correspondientes a los puntos relevantes de la imagen [10].

- **Harris-Laplace**

Es un detector de puntos de interés invariantes en cuanto a rotación y escala. Los puntos se detectan mediante una función de Harris seleccionada en el espacio de escalas mediante el operador Laplaciano. La escala seleccionada determina el tamaño de la región de interés. Este detector es adecuado cuando las características son usadas para reconstrucción y localización y no sólo para reconocimiento puesto que este detector detecta características localizadas de forma más precisa [10].

- **SIFT (Scale Invariant Feature Transform)**

Es un detector, inicialmente presentado por Lowe [11] que extrae puntos distintivos de imágenes y además calcula el descriptor de los mismos. Se utiliza fundamentalmente para tareas de reconocimiento. Los puntos seleccionados son invariantes a escala, rotación y parcialmente invariantes a cambios de punto de vista e iluminación. El algoritmo de extracción se basa en la localización de los máximos y mínimos de una diferencia de gaussianas aplicada en el espacio de escalas. Esta extracción de características permite ejecutar el algoritmo de forma jerárquica de modo que las características se calculen a partir de una pirámide de imágenes con muestreo entre cada nivel [10].

- **SURF (Speeded Up Robust Features)**

Es un detector de puntos relevantes presentado por Herbert Bay [12] que calcula además sus descriptores. Es un método robusto invariante a rotación y escala. Permite también predecir la posición en la imagen consecutiva de los puntos detectados en la imagen. Este método está basado en SIFT; sin embargo es más rápido y más robusto frente a diferentes transformaciones de imágenes que SIFT. El algoritmo SURF está basado en sumas de aproximaciones de respuestas en 2D a la wavelet de Haar. La detección de puntos de interés se obtiene a partir de una aproximación simple de matriz hessiana [13].

- **SUSAN (Smallest Univalued Segment Assimilating Nucleus)**

El algoritmo SUSAN [14] permite la detección de esquinas en una imagen. La idea del algoritmo consiste en suponer que en torno al punto relevante existen dos áreas bien definidas. La primera de ellas corresponde a los píxeles que pertenecen a la esquina y la segunda corresponde a los píxeles que están fuera de la esquina. El algoritmo trabaja

mediante ventanas con forma circular que se sitúan en todos los puntos de la imagen. Los puntos relevantes se detectan cuando el área circular contiene un mínimo de puntos parecidos al píxel central [1].

- **FAST (Features from Accelerated Segment Test)**

El algoritmo FAST [15] es un detector de esquinas que se caracteriza por producir puntos relevantes muy estables. Este método pertenece a la categoría AST (Accelerated Segment Test), que es una versión modificada del criterio SUSAN para la detección de esquinas. Es el detector de esquinas computacionalmente más eficiente que existe hasta el momento. Este algoritmo trabaja mediante ventanas circulares centradas en cada uno de los puntos de la imagen. Para que un punto sea detectado como esquina deben existir un número mínimo de puntos dentro de su ventana que sean una cantidad umbral más claros que el centro u otra cantidad umbral más oscuros que el centro [13].

Frente a los métodos de extracción de puntos descritos anteriormente existe la alternativa de extraer los puntos de forma homogénea. Este es el modo más sencillo de proceder y que menos carga computacional supone. Sin embargo, no proporciona ningún control para la selección de las características que deben tener los puntos. Esto hace que se trate de un método muy ineficiente puesto que los puntos seleccionados cubren todo el espacio de la imagen de forma equilibrada, pero sin embargo pueden obviar regiones de la imagen que potencialmente resulten de interés.

2.4.2. Fase de determinación de correspondencia entre puntos

El alineamiento por características requiere una fase donde se determine la correspondencia entre los puntos característicos extraídos de dos imágenes consecutivas. Los métodos más importantes que resuelven este problema son los siguientes:

- **Correspondencia basada en flujo óptico:**

Este método consiste en determinar el patrón de movimiento existente entre dos imágenes consecutivas mediante la estimación de las velocidades instantáneas que existe entre ellas. A diferencia de la técnica de alineamiento global basada en flujo óptico, que será explicada en la sección siguiente, esta técnica realiza los cálculos de flujo óptico solamente para los puntos relevantes de la imagen, de modo que se obtiene una ecuación para cada punto característico. La resolución de estas ecuaciones se lleva a cabo mediante los métodos de Lucas-Kanade o de mínimos cuadrados [1].

- **Correspondencia basada en similitud de descriptores:**

Este método parte de la existencia de un descriptor asociado a cada punto relevante extraído de la imagen que contiene información sobre la escala y la orientación. Para llevar a cabo la identificación entre los puntos relevantes de las dos imágenes se calculan las distancias entre cada punto de una imagen y todos los puntos de la imagen siguiente y las correspondencias entre puntos se forman emparejando los puntos cuyos descriptores tengan la menor distancia. El proceso se repite para cada punto de la imagen origen [1].

2.4.3. Fase de alineamiento de la imagen

El alineamiento de una imagen pretende eliminar el movimiento existente entre imágenes consecutivas de tal forma que se consiga eliminar el movimiento aparente de la cámara. Este alineamiento se puede realizar de forma global, por características o por parámetros intrínsecos de la cámara.

- **Alineamiento global:**

Este alineamiento trabaja con todos los puntos de las imágenes que se pretenden alinear. De este modo, cada píxel de la imagen sufre una transformación que permite que exista una correspondencia entre ambas imágenes y por tanto no hay que realizar extracción de puntos como fase previa. Existen diversos métodos para llevar a cabo dicho alineamiento en función del modo de obtener dicha transformación [1]:

- **Identificación con plantilla:**

Este método consiste en utilizar dos imágenes consecutivas: la primera de ellas hará la función de plantilla y la segunda será la imagen objetivo. El alineamiento consiste en realizar una transformación de la imagen plantilla para adecuarla a la imagen objetivo. La transformación se realiza conforme a unas restricciones iniciales acerca del tipo de transformación que se desea hacer. Las transformaciones más habituales son transformaciones simples, que incluyen tres parámetros para traslaciones y rotaciones; transformaciones afines, que incluyen seis parámetros; y transformaciones proyectivas, de nueve parámetros.

Una vez que la imagen han sido transformada, se calcula la similitud existente entre ambas imágenes para determinar la calidad del alineamiento. Las medidas de similitud pueden ser la correlación, el error cuadrático medio o la información mutua.

El objetivo consiste en maximizar la similitud entre las imágenes mediante la elección de los parámetros de la transformación, lo cual se consigue mediante un algoritmo de búsqueda.

Este método presenta el inconveniente de ser muy costoso en tiempo debido a la búsqueda de los parámetros óptimos si la resolución de la imagen es alta [1].

- **Flujo óptico:**

Este método consiste en determinar el patrón de movimiento existente entre dos imágenes consecutivas mediante la estimación de las velocidades instantáneas que existe entre ellas.

Las velocidades de los puntos se calculan mediante el desarrollo de Taylor de forma que el resultado quede en función de las velocidades v_x y v_y . Para la ecuación que relaciona ambas velocidades se deben buscar los valores de los parámetros que la hacen mínima. Para minimizar dichas ecuaciones se suele utilizar el método de mínimos cuadrados o el método de Lucas-Kanade, que resuelve el sistema imponiendo unas restricciones considerando que el flujo situado en las proximidades del píxel sobre el que se trabaja es constante.

Una vez obtenidas las velocidades, se puede determinar el flujo dominante, que es el que corresponde al fondo de la imagen, y de este modo, se consigue alinear las dos imágenes eliminando así el movimiento de la cámara [1].

Este método es el más habitual; sin embargo sólo puede utilizarse si los cambios en la intensidad de los píxeles no es muy significativa.

- **Modelo elástico:**
 Los métodos anteriores llevan a cabo el alineamiento de las imágenes suponiendo que la imagen transformada y la segunda imagen diferirán en una serie de puntos, los cuales serán los puntos con movimiento.
 Este método sin embargo trabaja con el objetivo de hacer coincidir completamente la imagen transformada con la segunda imagen. Para ello, supone que la primera imagen se puede adaptar completamente a la segunda por medio de una transformación que considere la imagen como un material elástico de modo que las transformaciones de una zona no afecten a las transformaciones del resto de las zonas. Se considera por tanto que la imagen que pretendemos alinear puede adaptarse completamente a la imagen previa deformando, según convenga, unas zonas u otras con independencia de las deformaciones en el resto de la imagen [1].

- **Alineamiento por características:**
 Los métodos que utilizan alineamiento por características requieren en primer lugar una fase de extracción de puntos seguida de una fase de determinación de correspondencia entre puntos explicadas anteriormente. El objetivo del alineamiento por características consiste en, a partir de las correspondencias existentes entre los puntos relevantes de dos imágenes buscar la transformación que ponga en correspondencia unos puntos con otros. Las transformaciones que se pueden realizar pueden ser simple, afín o proyectiva. Mediante este método se consigue una reducción considerable de la carga computacional asociada al alineamiento entre imágenes con respecto a la técnica de alineamiento global puesto que en lugar de buscar correspondencias para todos los puntos de la imagen, sólo se buscan para unos puntos relevantes, que deben ser lo más robustos posibles frente a las variaciones entre imágenes [1].

- **Alineamiento por parámetros intrínsecos de la cámara:**
 Esta técnica de alineamiento utiliza los parámetros intrínsecos de la cámara que realiza el vídeo para estimar el movimiento que se produce en la propia cámara. Para obtener la relación que existe entre cada posición de los píxeles en las dos imágenes se utiliza la fórmula de Kanatani, cuyas ecuaciones se muestran a continuación:

$$x_{t-1} = f \frac{x_t + \alpha \sin \theta y_t + f \alpha \cos \theta}{-\alpha \cos \theta x_t + \gamma y_t + f} \quad (2.1)$$

$$y_{t-1} = f \frac{-\alpha \sin \theta x_t + y_t + f \gamma}{-\alpha \cos \theta x_t + \gamma y_t + f} \quad (2.2)$$

Siendo θ la inclinación inicial del sistema de la cámara, α el desplazamiento, γ la inclinación de la rotación y f la distancia focal, todos ellos parámetros conocidos.

La posición de un determinado píxel en la imagen n , determinada por (x_t, y_t) se convierte a través de estas ecuaciones a la posición (x_{t-1}, y_{t-1}) en la imagen siguiente.

El inconveniente de este método reside en que para que el método sea efectivo los desplazamientos de la lente central desde el centro de rotación de la cámara deben ser pequeños. Además esta técnica sólo es posible aplicarla si se conocen los parámetros intrínsecos de la cámara [1].

2.4.4. Fase de segmentación de movimiento

Esta fase tiene el objetivo de detectar los píxeles de la imagen que corresponden a los objetos móviles y separarlos del fondo de la imagen, que aunque tenga un movimiento producido por haber tomado los vídeos con cámara móvil en realidad, no se ha movido. Existen dos clases métodos para detectar los puntos donde existe movimiento real de objetos en función de si se realiza alineamiento entre imágenes consecutivas o no.

- **Métodos que diferencian imágenes consecutivas alineadas:**

Los métodos descritos en el apartado anterior se encargan de conseguir un alineamiento de la imagen; es decir tratan de relacionar las tramas consecutivas de tal forma que se consiga eliminar el movimiento aparente de la cámara. Esto se consigue tratando de distinguir qué parte de la imagen corresponde al fondo y cuál a los objetos móviles. Para que este método funcione correctamente la mayor parte del movimiento debe estar causada por el movimiento de la cámara, porque si no, la alineación no se estaría produciendo con respecto al fondo sino con respecto a los objetos que se mueven. Los píxeles donde existe movimiento son aquellos que una vez alineadas dos imágenes consecutivas, son diferentes en ambas imágenes [1].

- **Métodos de detección basada en distorsión de flujo:**

La alternativa al método anterior consiste en lo realizar alineamiento entre imágenes y tratar de descubrir los píxeles con movimiento directamente sobre las imágenes. Para ello, se calcula el patrón de movimiento general de la imagen y se determinan los píxeles que no encajan con dicho movimiento. Estos píxeles corresponden con los objetos que se mueven.

El flujo global se extrae ajustando un sistema de ecuaciones que relaciona los flujos de las dos imágenes. La parte de flujo no ajustado se considera correspondiente a los objetos móviles y el resto de flujo se considera que pertenece al fondo de la imagen [1].

2.4.5. Fase de definición de objetos

La definición de objetos tiene como objetivo la agrupación de los píxeles a los que se les ha detectado movimiento en las fases anteriores. De este modo se pretende que los píxeles con movimiento pasen a formar parte de objetos de forma que sólo sea necesario almacenar ciertas características de los objetos en lugar de cada uno de los puntos con movimiento, reduciendo así en gran medida la carga computacional de cara a las fases siguientes.

Para llevar a cabo la definición de objetos se utilizan algoritmos de agrupamiento, los cuales se basan en la idea de que dados n puntos en un espacio n -dimensional se pueden particionar los mismos en k grupos tales que los puntos dentro de un grupo son más similares que cada uno a los de los otros grupos.

Los algoritmos de agrupamiento pueden ser paramétricos si se tiene un conocimiento previo sobre la forma funcional de las distribuciones de probabilidad de cada clase sobre el espacio de representación y de este modo las fronteras de decisión estarán definidas por dichas

distribuciones de clases, y no paramétricos si no se tiene ningún conocimiento previo sobre las formas de las distribuciones y el único conocimiento que se tiene es la información inducida a partir de un conjunto de muestras, de modo que las fronteras de decisión estarán determinadas por las muestras del conjunto de entrenamiento.

Para el problema de video vigilancia sólo pueden ser usados los algoritmos no paramétricos puesto que el objetivo del problema consiste en detectar objetos de los que no se tiene información a priori. Existen tres clases distintas de agrupamiento no paramétrico: agrupamiento jerárquico, agrupamiento particional y agrupamiento basado en densidad [16].

- En los **algoritmos de agrupamiento jerárquico** el conjunto de datos se particiona por niveles de forma que en cada nivel generalmente se unen o se dividen dos grupos del nivel anterior dependiendo de si el algoritmo es aglomerativo o divisivo.
 - **Unión puntual** (Single Link, SL): En cada paso se unen los dos grupos cuyos elementos más cercanos tienen la mínima distancia.
 - **Unión de la media** (Average Link, AL): En cada paso se unen los dos grupos tal que tienen la mínima distancia promedio entre sus puntos.
 - **Unión completa** (Complete Link, CL): En cada paso se unen los dos grupos tal que su unión tiene el diámetro mínimo o los dos grupos con la menor distancia máxima entre sus elementos.

- Los **algoritmos de agrupamiento particional** trabajan realizando una división inicial de los datos en grupos y moviendo después los objetos de un grupo a otro tratando de optimizar una función determinada. Estos algoritmos requieren que se conozca a priori el número de agrupaciones entre las que se debe distribuir el conjunto de datos. Algunos de los algoritmos de agrupamiento particional son los siguientes:
 - **K-Means:**

Este algoritmo tiene como objetivo dividir los datos en un número prefijado de agrupamientos que contienen cada uno de ellos un centroide. Para ello se definen unos centroides iniciales alrededor y los datos se agrupan alrededor del centroide del que estén más cerca. A continuación, se recalcula el centroide de cada uno de los agrupamientos y se vuelven a distribuir todos los datos en función del centroide del que se encuentran más cerca. Este proceso se repite hasta que las agrupaciones se estabilicen y no haya cambios de datos entre una iteración y otra.

El inconveniente de este agrupamiento es que falla cuando los puntos de un grupo están muy cerca del centroide de otro grupo o cuando los agrupamientos son diferentes en cuanto a tamaño y forma.
 - **CURE:**

Este algoritmo es una mezcla entre los algoritmos de agrupamiento particional y agrupamiento jerárquico que trata de eliminar los inconvenientes de cada uno de ellos. Su principio de funcionamiento es el siguiente: en cada grupo se eligen varios puntos representativos en lugar de un solo punto. Estos puntos se calculan a partir de los más dispersos del grupo, los cuales se atraen hacia el centro con el mismo factor de

compresión y en cada paso se unen los dos grupos más cercanos y una vez unidos se vuelve a calcular para este nuevo grupo los puntos representativos.

La similitud entre los grupos se mide por la similitud que tenga del par de puntos más cercanos que sean uno de cada grupo.

- Los **algoritmos de agrupamiento basado en densidad** se basan en la idea de dividir los elementos de una base de datos en grupos teniendo en cuenta la distribución de densidad de los puntos, con el objetivo de que los grupos que se formen tengan una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad. Estos algoritmos utilizan diversas técnicas que pueden ser grafos, técnicas basadas en histogramas, núcleos y emplean los conceptos de punto central, borde o ruido.

2.4.6. Fase de seguimiento de objetos

El objetivo de esta fase consiste en detectar en varias tramas consecutivas la presencia de un objeto que se mueve en cada escena. Este seguimiento es útil para averiguar la trayectoria que sigue dicho objeto. Esta fase también es necesaria puesto que la fase de segmentación del movimiento genera errores en la detección de objetos debidos al ruido y mediante el seguimiento del objeto se consiguen reducir dichos errores. Algunos de los errores producidos pueden consistir en falsas alarmas correspondientes a objetos que se detectan como móviles cuando en realidad se trata de objetos que simplemente vibran en la escena pero que no se desplazan a lo largo de ésta. De esta forma se puede decir que mediante el seguimiento de objetos se consigue un filtrado temporal y espacial de los objetos.

Existen diversas formas de realizar seguimiento de objetos en imágenes de vídeo. En la literatura, la clasificación más habitual es la siguiente [4] [17]:

- **Seguimiento mediante modelado:**
El seguimiento mediante modelado parte de la base de la utilización de modelos tridimensionales que representan los objetos que se espera que aparezcan en el vídeo. Para la construcción de dichos modelos se parte de mediciones tomadas fuera de línea y su posterior proyección tridimensional generalmente mediante herramientas de diseño asistido por ordenador o mediante técnicas de visión artificial. La posición se estima ajustando el modelo sintético obtenido al aplicar transformaciones a los puntos característicos de la imagen.
Se trata de un método muy robusto pero, por contra requiere del conocimiento previo de las imágenes de la escena y la carga computacional es muy alta.
- **Seguimiento basado en el contornos activos:**
El seguimiento basado en contornos activos busca realizar el seguimiento de los objetos a través de la delimitación de los mismos. Este contorno se puede modelar mediante una curva parametrizada que defina la posición, el tamaño y la forma del mismo. La actualización de dichos contornos se realiza de forma automática entre las tramas sucesivas. Para ello se puede utilizar el filtro de Kalman de tal manera que se realice una

predicción de la curva parametrizada de la siguiente imagen, y se corrijan los errores en la predicción a partir de las mediciones realizadas en la trama siguiente.

Este método es probablemente el que menor carga computacional genera; sin embargo la gran dificultad reside en el modelado de la curva paramétrica y en el diseño del modelo dinámico que describa el comportamiento del contorno durante el proceso de seguimiento, necesario para el filtro de Kalman.

- **Seguimiento de elementos característicos:**

El seguimiento de elementos característicos parte de la detección de una serie de elementos relevantes de un determinado objeto como puede ser vértices de bordes angulares o líneas de campo de movimiento. Estos elementos característicos son seguidos a través de las sucesivas imágenes para determinar la trayectoria del objeto.

Este seguimiento soluciona la mayoría de los problemas de oclusión pero tiene como inconveniente que el coste computacional es alto.

- **Seguimiento de componentes conectados:**

El seguimiento de componentes conectados se puede considerar un caso particular del seguimiento de elementos característicos debido a que el término *elemento característico* es muy general. Para llevar a cabo el seguimiento de componentes conectados se parte de una extracción de *blobs* (componentes conectados) que se siguen a través de las sucesivas tramas. Para llevar a cabo este seguimiento se puede hacer uso del filtro de Kalman de forma que un objeto quede determinado por su centroide y su tamaño y la ecuación de estado se modele mediante las ecuaciones cinemáticas de posición y velocidad en el plano.

El problema principal de esta solución consiste en que si el número de objetos es alto, podría suceder que varios objetos quedaran fusionados en un sólo componente conectado como consecuencia de la oclusión o de otros factores. Sin embargo, si el número de objetos a detectar es bajo, esta solución es la más sencilla de implementar.

2.5. Técnicas de procesamiento de imagen para la detección y seguimiento de objetos desde cámaras en movimiento utilizadas en este proyecto

En este proyecto se utilizará un método basado en alineamiento de imágenes por características. Este método es el más apropiado para este proyecto puesto que los costes computacionales y de tiempo son relativamente bajos, lo cual permite que el procesamiento sea rápido, y eventualmente podría conseguirse un funcionamiento en tiempo real. Por otro lado, el alineamiento consigue dotar de robustez a la segmentación siempre y cuando la mayor parte de movimiento aparente entre las dos imágenes corresponda al movimiento de la cámara, como es caso de estudio, en el cual, las imágenes tomadas por el UAV se espera que correspondan a escenarios donde el número de objetos que se mueven y su tamaño son pequeños.

Respecto a las siguientes etapas, en la etapa de agrupamiento se utilizará un agrupamiento particional y la fase de seguimiento se basará en el método de componentes conectados puesto que se trata de un método muy sencillo de implementar y la carga computacional que genera es baja debido a que los objetos se representan como elementos conectados de los que se almacena información únicamente de la posición de su centro, su tamaño y las intensidades de sus colores RGB. Este seguimiento se realizará mediante la utilización de un filtro de Kalman cuya ecuación de estado se modele mediante las ecuaciones cinemáticas de posición y velocidad en el plano y permita un filtrado espacial y temporal de los objetos con el objetivo de eliminar posibles objetos que son en realidad ruido producido en la etapa de segmentación del movimiento.

Capítulo 3: Diseño del modelo experimental

3.1. Introducción

Este capítulo está destinado a explicar el conjunto de técnicas y algoritmos de procesamiento de imagen que constituyen las diferentes fases etapas que integran este método enfocado al seguimiento de objetos desde vehículos aéreos no tripulados.

El método elegido se basa en el paradigma PTAM, que por sus siglas en inglés significa Algoritmo de seguimiento y representación en paralelo (Parallel Tracking and Mapping). Este modelo, consiste en realizar, por dos caminos diferentes, el seguimiento y la transformación matemática de la secuencia de imágenes procesadas.

La parte de seguimiento hace referencia al estudio de la posición de la cámara del UAV con referencia a una determinada escena de la cual no se tiene conocimiento previo. Se pretende, por tanto estimar la trayectoria que siguen determinados puntos de la imagen previamente seleccionados. La base fundamental del seguimiento que se lleva a cabo en este experimento se basa en un estimador matemático conocido como filtro de Kalman.

La parte de transformación matemática, por su parte, consiste en la determinación de unos puntos relevantes extraídos de la imagen que son los que serán seguidos a través de las imágenes subsiguientes y encontrar sus correspondientes aplicaciones matemáticas en las imágenes posteriores.

Estos dos métodos se complementan de la siguiente forma: La parte de seguimiento proporciona la estimación de la trayectoria que siguen los puntos relevantes, lo cual le sirve de entrada al bloque de aplicación, que determina la posición que ocupan estos puntos en la nueva imagen. La forma en que se complementan ambos bloques, permite que se pueda llevar a cabo una optimización secuencial del algoritmo, lo que es la base para su posible traspaso a la implementación del algoritmo en tiempo real.

El problema de video vigilancia desde cámaras móviles presenta unos retos adicionales a los que presenta el problema de vigilancia para cámaras fijas. Por ello, la herramienta que se presenta en este proyecto, deberá incluir fases específicas que traten de compensar el movimiento de la cámara para poder considerar sólo el movimiento causado por los objetos móviles. Una fase de cálculo de flujo óptico artificial, que describa el movimiento causado por la cámara será la forma en que esta herramienta conseguirá su correcto funcionamiento desde cámaras en movimiento.

En las secciones posteriores se presentan las fases y algoritmos concretos que intervienen en el procesamiento de las imágenes para el seguimiento de objetos desde UAVs.

3.2. Visión global del método experimental

El problema de video vigilancia mediante cámaras en movimiento se basa en conseguir separar el movimiento causado por la cámara del movimiento causado por los objetos que se mueven en el vídeo. Además, los UAVs presentan la particularidad de que ocasionan vibraciones en la adquisición de vídeo, lo que hace más importante realizar una correcta caracterización del movimiento de la cámara.

Con el objetivo de conseguir distinguir entre los movimientos causados por la cámara y por los objetos en movimiento, se lleva a cabo una implementación basada en el cálculo de los flujos ópticos: un flujo óptico real, que describe todo el movimiento causado en la secuencia de imágenes y un flujo óptico artificial, que recoge solamente el movimiento causado por la cámara. La comparación de ambos flujos permite deshacer el movimiento de la cámara.

Para su correcto funcionamiento, la herramienta se ha diseñado de acuerdo a las siguientes fases (ver figura 3.1):

- **Extracción de puntos:** Esta fase tiene como objetivo seleccionar unos puntos de interés de la imagen, que será con los que se trabajará en el resto de fases para evitar tener que realizar el procesamiento de todos los puntos de la imagen. Estos puntos se seleccionan con el método de extracción de puntos, SURF, que permite extraer una serie de puntos relevantes en la imagen que suelen corresponder a bordes de objetos. Los puntos extraídos son los suficientemente característicos como para que no degradar el rendimiento de las fases posteriores del diseño. Esta selección de puntos permite reducir en gran medida la carga computacional de las fases posteriores.
- **Correspondencia entre puntos:** Esta fase tiene como objetivo realizar un seguimiento de los puntos de interés a través de cuadros consecutivos del vídeo. El algoritmo que se utiliza para llevar a cabo este propósito es el algoritmo de KLT, basado en correspondencia de flujo óptico, que permite obtener las correspondencias entre puntos de dos imágenes consecutivas, para así poder conocer la velocidad que tienen dichos puntos.
- **Alineamiento de la imagen:** Esta fase tiene como objetivo transformar la imagen anterior conforme al movimiento que ha tenido la cámara entre la imagen anterior y la imagen actual. De esta forma se consigue hacer coincidir globalmente la imagen anterior y la imagen actual, de tal forma que las únicas discrepancias que existan entre ambas, correspondan al movimiento de los objetos. Para llevar a cabo el alineamiento de las imágenes en primer lugar hay que calcular la transformación y en segundo lugar aplicar dicha transformación. Para calcular la transformación se hará uso del algoritmo MSAC que permite excluir los outliers de la transformación.

- **Segmentación del movimiento:** Esta fase tiene como objetivo determinar la eliminación del fondo de la imagen, para así dejar al descubierto los píxeles dinámicos. Esto se consigue realizando una comparación entre las imágenes que se han alineado previamente.

Esta herramienta realizará la segmentación del movimiento mediante la comparación de un flujo óptico real, que describe el movimiento de los píxeles de interés y un flujo óptico artificial, que refleja únicamente el movimiento de la cámara. Los puntos de interés cuyos flujos ópticos difieran en un determinado ángulo o módulo serán clasificados como píxeles dinámicos.
- **Agrupamiento:** Esta fase tiene como objetivo definir los objetos móviles mediante la agrupación de los píxeles dinámicos que han sido detectados en la fase anterior y pertenecen a un mismo objeto.

Esto permitirá reducir en gran medida la carga computacional puesto que cada objeto pasará a estar definido únicamente por la posición de su centro, su tamaño y los niveles de intensidad de sus colores.

Se realizará un agrupamiento divisivo basado en el método de las k-medias (k-means).
- **Seguimiento:** Esta fase tiene como objetivo realizar un filtrado espacial y temporal de los objetos detectados mediante un seguimiento de dichos objetos a través de los cuadros de vídeo.

Este seguimiento permitirá eliminar objetos que han sido detectados como consecuencia de ruido en la fase de segmentación, u objetos producidos por vibraciones, que no son realmente objetos en movimiento, como puede ser un arbusto que vibra ondeado por el viento.

El seguimiento se realizará con un filtro de Kalman.

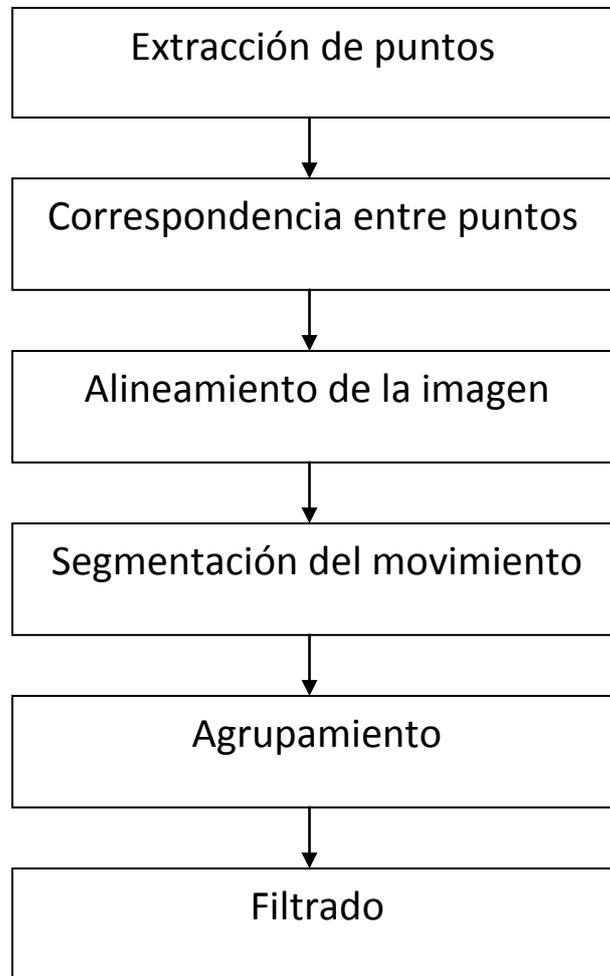


Figura 3.1: Fases que componen el algoritmo diseñado para la detección y seguimiento de objetos en movimiento.

En la figura 3.2 se presenta un diagrama de bloques donde se presentan todos los elementos que intervienen en la ejecución de la herramienta:

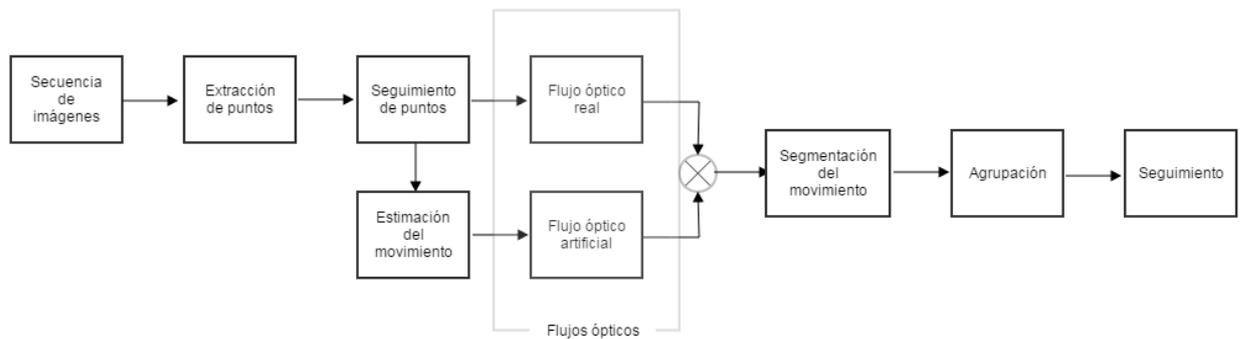


Figura 3.2: Diagrama de bloques del proceso de detección y seguimiento de objetos similar al presentad en [37].

3.3. Extracción de puntos: Algoritmo SURF

La extracción de puntos relevantes de las imágenes es un método que se utiliza con el objetivo de rebajar la carga computacional originada por las partes posteriores del procesamiento de las imágenes. De esta forma, el cálculo se vuelve más sencillo y rápido, aunque se vuelva también menos robusto. Sin embargo es un compromiso que hay que tener en cuenta a la hora del diseño del sistema. En el caso presentado en este trabajo, sin embargo es el método más adecuado, puesto que presenta buenos resultados y facilita que el algoritmo pueda ser extrapolable a un posible procesamiento en tiempo real.

Para que los puntos extraídos resulten robustos frente al ruido, debe buscarse un algoritmo que extraiga una serie de puntos que sean relevantes en la imagen. De este modo, sería una mala idea extraer los puntos de la imagen de forma homogénea, en forma de malla.

El algoritmo que se usará en este proyecto para llevar a cabo la extracción de puntos es el algoritmo SURF, debido a que los puntos extraídos son robustos frente a posibles transformaciones en la imagen.

SURF (Speeded Up Robust Features)

HISTORIA

SURF es un algoritmo de extracción de puntos de interés invariantes que fue presentado por Herbert Bay [12] en la Conferencia Europea de Visión Artificial (ECCV) de 2006 celebrada en Graz (Austria).

Este algoritmo está basado en el algoritmo SIFT (Scale-Invariant Feature Transform), presentado por David Lowe en 1999.

El algoritmo SURF presenta las siguientes mejoras respecto al algoritmo SIFT [18]:

- Velocidad de cálculo muy superior manteniendo el mismo rendimiento.
- Mayor robustez ante posibles transformaciones de la imagen.

DESCRIPCIÓN

SURF es un algoritmo de extracción de puntos de interés muy utilizado para la detección y seguimiento de objetos. Este algoritmo es invariante frente a rotaciones, cambios de escala y cambios en la iluminación de las mismas. Este método está basado en SIFT; sin embargo es más rápido y más robusto frente a diferentes transformaciones de imágenes que SIFT debido a que el algoritmo SURF almacena menos descriptores para los puntos de interés que SIFT.

ALGORITMO

El algoritmo SURF consta de las siguientes fases:

- Detección de puntos de interés.
- Asignación de orientación a los puntos de interés.
- Cálculo de descriptores asociados a los puntos de interés.

A continuación se describe cada una de las etapas. Las figuras presentadas en esta sección se basan en las presentadas en [12]:

Detección de puntos de interés

La primera de las etapas del algoritmo consiste en detectar los puntos de interés de una imagen. Para cada uno de los puntos se detecta su posición y su escala.

Este proceso se lleva a cabo de forma piramidal, definiendo el número de pirámides u octavas con las que se desea trabajar. Cada una de las octavas está compuesta por un número fijo de imágenes o capas que son el resultado de la convolución de la imagen analizada con una serie de filtros de dimensiones cada vez mayores.

Las dimensiones del filtro dentro de cada octava cumplen las siguientes características:

- El incremento de las dimensiones de los filtros dentro de cada octava debe ser el doble que el paso en la etapa anterior.
- Las dimensiones del primer filtro de cada octava son el doble que las dimensiones del segundo filtro de la octava anterior.

Estas octavas determinan la escala del punto de interés.

En la figura 3.3 se muestra la forma en que se relacionan las pirámides y la escala:

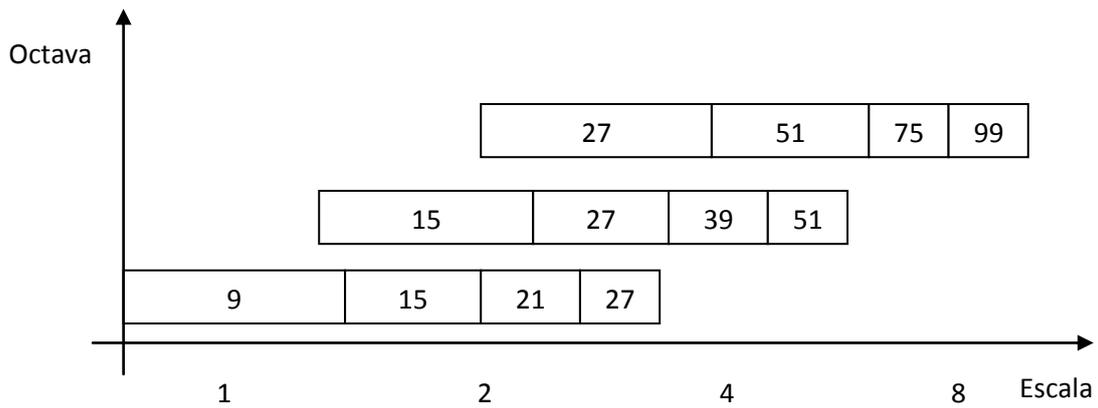


Figura 3.3: Esquema de la relación existente entre las octavas y la escala durante el proceso de detección de puntos de interés.

Las pirámides parten de imágenes integrales, que permiten acelerar los cálculos de los filtros de convolución. La imagen integral $I_{\Sigma}(\mathbf{x})$ en el punto $\mathbf{x} = (x, y)$ representa la suma de todos los píxeles de la imagen de entrada en una región rectangular formada por el origen y \mathbf{x} :

$$I_{\Sigma}(\mathbf{x}) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (3.1)$$

La utilización de imágenes integrales permite el aumento del tamaño del filtro a un coste constante de tiempo a diferencia de la reducción del tamaño de la imagen, que no proporciona un coste constante.

Para llevar a cabo la detección de los puntos de interés se hace uso de una aproximación determinante de la matriz Hessiana, lo que permite aumentar el rendimiento del algoritmo en términos de velocidad de cálculo y precisión. Además, el algoritmo SURF utiliza este valor para el cálculo de la posición y la escala de los puntos de interés en lugar de utilizar diferentes medidas para cada uno de estos valores.

Para un punto de interés $\mathbf{x} = (x, y)$ de una imagen I , la matriz Hessiana $\mathcal{H}(\mathbf{x}, \sigma)$ en \mathbf{x} a escala σ se define de la siguiente forma:

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathbf{x}, \sigma) & L_{xy}(\mathbf{x}, \sigma) \\ L_{xy}(\mathbf{x}, \sigma) & L_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \quad (3.2)$$

donde $L_{xx}(\mathbf{x}, \sigma)$ es la convolución de la derivada de segundo orden de una Gaussiana $\frac{\partial^2}{\partial x^2} g(\sigma)$ con la imagen I en el punto \mathbf{x} , y de forma similar para $L_{xy}(\mathbf{x}, \sigma)$ y $L_{yy}(\mathbf{x}, \sigma)$.

La aproximación del determinante de la matriz Hessiana se calcula del siguiente modo:

$$\det(\mathcal{H}_{aprox.}) = D_{xx}D_{yy} - (0,9D_{xy})^2 \quad (3.3)$$

donde D_{xx} , D_{xy} y D_{yy} son las aproximaciones de las derivadas parciales y el valor de 0,9 proviene de la aproximación del filtro Gaussiano.

La localización de los puntos de interés se consigue mediante la supresión de puntos que no cumplen la condición de máximo en una ventana de tamaño $3 \times 3 \times 3$. De esta forma se consigue que el valor máximo del determinante de la matriz Hessiana sea interpolado en posición y escala en la imagen. Esta interpolación es muy importante puesto que la diferencia entre capas dentro de cada octava es relativamente grande.

Asignación de la orientación

Esta etapa tiene como objetivo permitir que los puntos de interés sean invariantes frente a cambios de escala, rotaciones o cambios de iluminación en la imagen. Esto permite que los puntos de interés tengan una repetibilidad que hace que un punto considerado fiable sea detectado bajo distintos puntos de vista.

Para determinar la orientación de un punto de interés se calcula la respuesta a las funciones ondulatorias de Haar en las direcciones x e y en una región circular de radio $4s$, siendo s la escala del punto de interés.

Las funciones ondulatorias de Haar se representan en la figura 3.4:

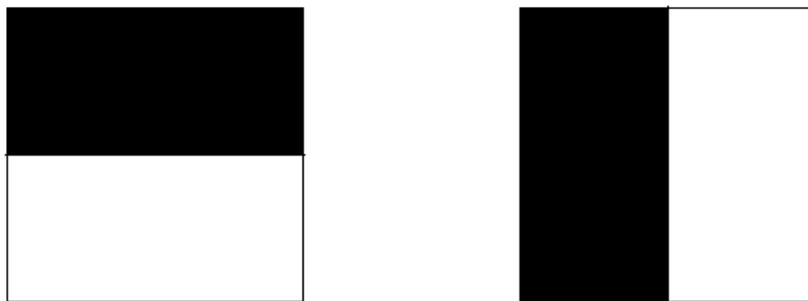


Figura 3.4: Funciones ondulatorias de Haar. La imagen de la izquierda corresponde a la respuesta en x , y la de la derecha a la respuesta en y . La parte oscura tiene valor -1 y la parte clara valor $+1$.

Una vez calculadas las respuestas a las funciones ondulatorias de Haar se ponderan éstas con una Gaussiana de parámetro $\sigma = 2s$ centrada en el punto de interés. Las respuestas obtenidas se representan en el plano complejo. La orientación dominante se estima calculando la suma de las respuestas en una ventana que barre todo el círculo cubriendo un ángulo de tamaño $\frac{\pi}{3}$.

Cálculo de descriptores

Para calcular los descriptores de los puntos de interés, el primer paso consiste en construir una región cuadrada centrada en el punto de interés y orientada según la orientación obtenida anteriormente. El tamaño de la región es de $20s$, siendo s la escala.

A continuación, se subdivide la región en 4 subregiones que a su vez se subdividen en 4 subregiones, y así sucesivamente hasta conseguir un total de 64 subregiones.

Para cada una de las subregiones se calcula la respuesta a la función ondulatoria de Haar. Las respuestas obtenidas en las direcciones horizontal y vertical se denotan como d_x y d_y , respectivamente, estando las direcciones horizontal y vertical definidas en relación a la orientación del punto de interés seleccionado. Las respuestas obtenidas se ponderan por una Gaussiana de parámetro $\sigma = 3,3s$ centrada en el punto de interés seleccionado.

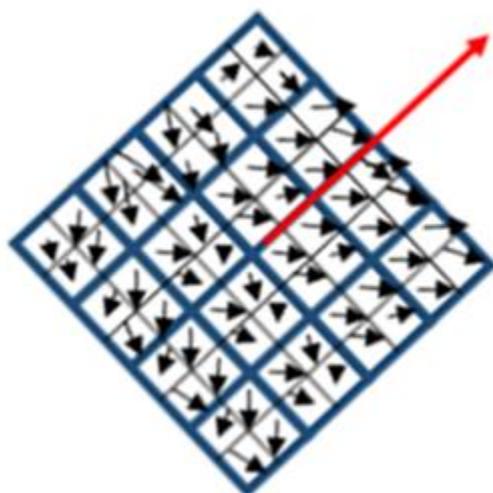


Figura 3.5: Subregiones en las que se divide la vecindad del píxel para realizar el cálculo de su orientación.

Estas respuestas se suman para cada subregión, formando así un primer conjunto de entradas para el vector de características. Para obtener información acerca de la polaridad de los cambios de intensidad se calculan los valores absolutos de las respuestas, $|d_x|$ y $|d_y|$. De este modo, cada subregión tiene asociado un vector de cuatro componentes $\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. El vector de características se obtiene concatenando el vector asociado a cada una de las 64 subregiones.

3.4. Correspondencia entre puntos: Algoritmo KLT

La determinación de correspondencia entre puntos de dos imágenes consecutivas tiene como objetivo emparejar los puntos de interés extraídos en la fase anterior correspondientes a dos imágenes consecutivas, de modo que se consiga seguir la trayectoria de los mismos en la secuencia de imágenes para así poder en fases posteriores deshacer el movimiento de la cámara.

El hecho de realizar el seguimiento para un conjunto de puntos de interés y no para todos los puntos de la imagen permite reducir considerablemente el tiempo de cálculo.

El algoritmo que se usará en este proyecto para el emparejamiento de puntos es algoritmo KLT, ya que se trata de un seguidor de puntos diseñado para secuencias de vídeo donde cada cuadro es muy parecido al anterior [19].

KLT (Kande-Lucas-Tomasi)

HISTORIA

KLT es un algoritmo de seguimiento de puntos que tiene como base el procedimiento propuesto por Lucas y Kanade para el encaje de dos imágenes en estereovisión [20] y que posteriormente fue mejorado por Tomasi y Kanade para permitir el seguimiento de puntos de interés que se repiten en cada imagen a lo largo de una secuencia [21].

Tomasi y Kanade plantean dos problemas fundamentales:

- Cómo seleccionar puntos característicos.
- Cómo realizar su seguimiento entre imagen y la siguiente.

El método que proponen tiene como objetivo solucionar el segundo problema, aunque a partir de éste consiguen determinar un método de extracción de puntos.

DESCRIPCIÓN

KLT es un algoritmo de seguimiento de puntos de interés diseñado para secuencias de vídeo, donde cada imagen es muy parecida a la anterior. Este algoritmo realiza seguimiento de puntos que tienen una cierta distancia con una tasa de acierto muy alta.

Las simplificaciones que este algoritmo asume son las siguientes [19]:

- Entre dos cuadros consecutivos se puede aproximar la transformación en regiones pequeñas por traslaciones.

- Entre dos cuadros consecutivos, las condiciones de iluminación y de adquisición son parecidas. lo que permite utilizar la suma de diferencias al cuadrado para medir las distancias.

ALGORITMO

Para establecer las correspondencias entre los puntos característicos de dos imágenes consecutivas el algoritmo KLT toma una imagen anterior y realiza una transformación de la imagen actual que permita adaptar esta imagen a la imagen anterior con el mínimo error de tal forma que las intensidades actuales puedan ser escritas como las anteriores más un término residual que depende casi linealmente del vector de traslación. Dicha transformación se calcula de forma iterativa y su convergencia es rápida.

Tomasi y Kanade realizan su estudio suponiendo que las transformaciones corresponden únicamente a traslaciones, lo cual no se cumple en varias situaciones como son las siguientes:

- Puntos que entran o salen de la imagen, y por consiguiente no tienen equivalencia.
- Puntos en fronteras de oclusión de un objeto por otro.
- Cambios en el punto de vista, lo que genera cambios en la intensidad de la imagen debidos a la reflectividad de las superficies.

Por consiguiente, los puntos pertenecientes a superficies con textura que estén lejos de bordes o fronteras de oclusión son buenos puntos para seguir en las situaciones en que las transformaciones consisten en traslaciones, de modo que una fase previa de extracción de estos puntos es de vital importancia.

Las ecuación que describe las transformaciones afines es la siguiente:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \begin{bmatrix} (1 + p_1) \cdot x + p_3 \cdot y + p_5 \\ p_2 \cdot x + (1 + p_4) \cdot y + p_6 \end{bmatrix} = \begin{bmatrix} 1 + p_1 & p_3 & p_5 \\ p_2 & 1 + p_4 & p_6 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.4)$$

o en el caso de que sólo haya traslaciones, la siguiente:

$$\mathbf{W}(\mathbf{x}; \mathbf{p}) = \begin{bmatrix} x + p_1 \\ y + p_2 \end{bmatrix} \quad (3.5)$$

Donde \mathbf{p} son los parámetros de la transformación que deben calcularse de tal forma que minimicen la suma del error cuadrático medio entre los niveles de intensidad de la imagen anterior o imagen plantilla T y la transformación de la imagen actual o imagen objetivo I , tal y como se muestra en la siguiente ecuación:

$$\sum_x [I(\mathbf{W}(x; \mathbf{p})) - T(x)]^2 \quad (3.6)$$

El cálculo de $I(\mathbf{W}(x; \mathbf{p}))$ implica que la posición de la imagen I no tiene por qué ser un número entero, o lo que es lo mismo, una posición de un píxel real, de modo que es necesario realizar una interpolación para calcular el valor de intensidad en dicha posición.

A continuación, se debe llevar a cabo la minimización de la ecuación anterior, que es una ecuación no lineal. Para poder linealizarla se procede de forma iterativa, suponiendo que se tiene una estimación de la transformación \mathbf{p} y se necesita minimizar las variaciones $\Delta\mathbf{p}$ que reduzcan el error de forma que en la ecuación

$$\sum_x [I(\mathbf{W}(x; \mathbf{p} + \Delta\mathbf{p})) - T(x)]^2 \quad (3.7)$$

hay que minimizar $\Delta\mathbf{p}$ y seguidamente actualizar el valor de la estimación de \mathbf{p}

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p} \quad (3.8)$$

La convergencia se alcanza cuando $\Delta\mathbf{p}$ se sitúa por debajo de un umbral prefijado.

El siguiente paso consiste en minimizar la ecuación (3.7). Esta minimización se lleva a cabo mediante el algoritmo de Gauss-Newton de gradiente descendente del desarrollo en serie de Taylor del término $I(\mathbf{W}(x; \mathbf{p} + \Delta\mathbf{p}))$ de dicha ecuación respecto a $\Delta\mathbf{p}$, en torno a \mathbf{p} y hasta el término lineal. La linealización produce la siguiente ecuación:

$$\sum_x \left[I(\mathbf{W}(x; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(x) \right]^2 \quad (3.9)$$

siendo $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$ el gradiente de la imagen I evaluado en $\mathbf{W}(x; \mathbf{p})$ y $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ el jacobiano de la transformación.

Si $\mathbf{W}(x; \mathbf{p}) = (W_x(x; \mathbf{p}), W_y(x; \mathbf{p}))^T$ entonces

$$\mathbf{W}(x; \mathbf{p}) = \begin{bmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \dots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \dots & \frac{\partial W_y}{\partial p_n} \end{bmatrix} \quad (3.10)$$

La solución al problema se obtiene minimizando la ecuación (3.9). Para ello se deriva respecto de $\Delta \mathbf{p}$ y se obtiene lo siguiente:

$$2 \sum_x \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[I(\mathbf{W}(\mathbf{x}; \mathbf{p})) + \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x}) \right] \quad (3.11)$$

Si se iguala a cero y se resuelve para $\Delta \mathbf{p}$ se obtiene la siguiente solución:

$$\Delta \mathbf{p} = H^{-1} \sum_x \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))] \quad (3.12)$$

siendo \mathbf{H} una matriz $n \times n$ que es la aproximación de Gauss-Newton a la matriz Hessiana:

$$H = \sum_x \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] \quad (3.13)$$

Para obtener la estimación final hay que resolver iterativamente la ecuación (3.12) para obtener $\Delta \mathbf{p}$ y poder actualizar la estimación de la transformación \mathbf{p} .

El algoritmo está diseñado específicamente para que las transformaciones sean únicamente traslaciones. En este caso el jacobiano $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ es la identidad, de forma que el sistema de ecuaciones que se debe resolver es el siguiente:

$$\left(\sum_x \nabla I^T \nabla I \right) \Delta \mathbf{p} = \sum_x \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))] \quad (3.14)$$

De la ecuación anterior se define

$$\mathbf{G} = \sum_x \nabla I^T \nabla I \quad (3.15)$$

$$\mathbf{e} = \sum_x \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^T [T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x}; \mathbf{p}))] \quad (3.16)$$

La matriz **G** es la matriz de los coeficientes y es de tamaño 2 x 2 y simétrica y el vector **e** es un vector columna de dimensión 2.

Para que el seguimiento sea fiable, el sistema (3.14) debe representar medidas fiables y se debe poder resolver de forma robusta, lo cual implica que la matriz G tenga valores mucho mayores que el ruido en la imagen y que esté bien acondicionada.

3.5. Alineamiento de la imagen: Cálculo de flujos ópticos real y artificial. Tipos de transformaciones. Algoritmo MSAC

El alineamiento de dos imágenes consecutivas tiene como objetivo transformar una imagen para hacerla coincidir globalmente con la otra. De este modo se consigue que la mayoría de los píxeles de ambas imágenes coincidan y unos pocos sean diferentes de tal forma que estos últimos corresponderán a los objetos móviles y aquéllos al movimiento de la cámara. En esta fase se conseguirá, por tanto, determinar los píxeles correspondientes a objetos móviles y deshacer el movimiento de la cámara para reducir el problema a un problema de cámara fija.

Para realizar el alineamiento entre imágenes se hará uso de la técnica de cálculo de dos flujos ópticos:

- **Flujo óptico real**, que calcula las velocidades a las que se mueven todos los puntos seguidos en la fase de seguimiento
- **Flujo óptico artificial**, que representa una estimación de la velocidad con la que se mueve la imagen global y corresponde al movimiento producido por la cámara.

Estos dos flujos serán comparados en la fase de segmentación de movimiento para determinar cuáles son los píxeles dinámicos.

A continuación se describe la implementación que se ha llevado a cabo de ambos flujos:

Flujo óptico real

El flujo óptico real representa las velocidades de cada uno de los píxeles seguidos y su cálculo se basa en gran medida en los resultados obtenidos en la fase de seguimiento de puntos. Una vez se ha realizado su seguimiento, para calcular las velocidades de los puntos de interés basta con calcular el desplazamiento que se ha producido en cada uno de los puntos entre dos imágenes consecutivas y la tasa de adquisición del vídeo.

Cabe destacar que aunque el diseño del método experimental está basado en el paradigma PTAM, los puntos seguidos para este algoritmo son estáticos. El objetivo de este proyecto consiste en detectar objetos dinámicos de modo que hace falta modificar ese detalle y realizar extracción de puntos relevantes en cada imagen.

Flujo óptico artificial

El flujo óptico artificial tiene como objetivo conocer el movimiento producido por la cámara sin tener en cuenta el movimiento particular de cada objeto. El flujo óptico representa, por tanto, una proyección matemática de los píxeles de interés en la siguiente imagen considerando el movimiento de la cámara entre dos imágenes consecutivas. Este movimiento se representa mediante una rotación y traslación (R, T) obtenida mediante el algoritmo PTAM.

Para calcular el flujo óptico artificial se estimará la transformación que se ha producido entre los puntos de interés de dos imágenes consecutivas y excluirá los puntos atípicos (outliers), que son aquellos que se desvían una cierta cantidad de la transformación estimada mediante el algoritmo MSAC, que tiene como objetivo maximizar el número de puntos correctos (inliers).

La transformación estimada entre imágenes, servirá para realizar una proyección de los puntos de interés de una imagen a la siguiente de acuerdo al movimiento de la imagen, lo que permitirá calcular el flujo óptico artificial a partir del desplazamiento calculado en la transformación y la tasa de adquisición del vídeo.

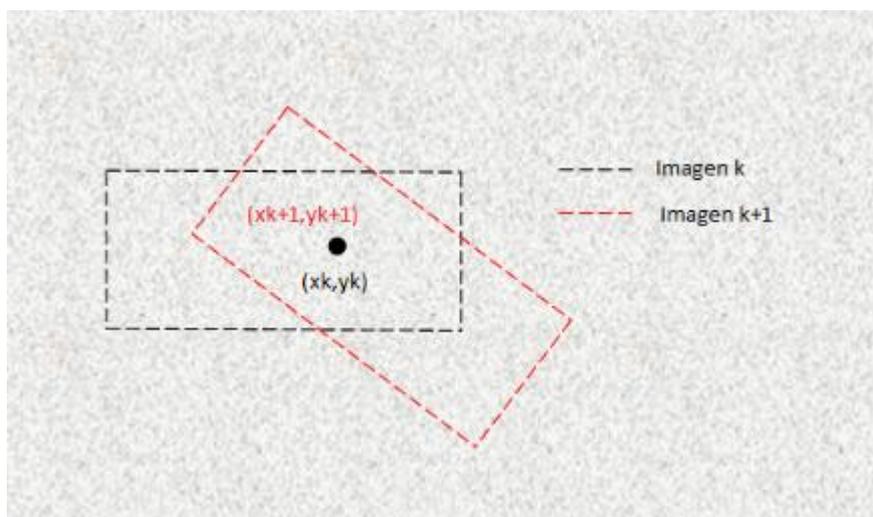


Figura 3.5: Ejemplo que muestra dos observaciones consecutivas sobre un escenario. Figura basada a la presentada en [27].

A continuación se describen los distintos tipos de transformaciones, y seguidamente se presenta el algoritmo MSAC.

Tipos de transformaciones

Existen distintos tipos de transformaciones, que se pueden clasificar en lineales y no lineales. Las transformaciones lineales son más simples y son las que más se suelen utilizar para realizar registrado de imagen. Las transformaciones suelen corresponder a traslaciones, rotaciones, simetrías y homotecias [1].

Las transformaciones lineales se dividen en los siguientes grupos:

- Transformación euclídea.
- Transformación afín.
- Transformación proyectiva.

Cada una de estas transformaciones se describe a continuación:

Transformación euclídea

Las transformaciones euclídeas definen los cambios producidos por traslaciones, rotaciones y reflexiones de objetos. Sin embargo, no es habitual que en los problemas de seguimiento se produzcan reflexiones de objetos pero sí traslaciones y rotaciones. Las transformaciones euclídeas se ajustan a los casos en los que la cámara del UAV se desplaza a una altura considerable del suelo y no existen grandes cambios en el terreno debidos a inclinaciones. En este caso, las transformaciones se consideran como bidimensionales puesto que las imágenes se recogen sobre fondos planos.

Para describir este comportamiento se utilizan tres parámetros: t_1 , que define las traslaciones horizontales, t_2 que define las traslaciones verticales y θ que define el ángulo de rotación.

Aunque no exista coordenada z , se puede considerar esta magnitud como valor constante e igual a 1 para poder escribir las ecuaciones en forma matricial.

Las traslaciones se definen del siguiente modo: Sea (x', y') el punto de la imagen actual y (x, y) el punto de la imagen actual, ambos puntos se pueden relacionar mediante la siguiente ecuación:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_1 \\ 0 & 1 & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.17)$$

La matriz que relaciona ambos puntos recibe el nombre de matriz de traslación.

Por su parte, las rotaciones relacionan ambos puntos de acuerdo a la siguiente ecuación:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.18)$$

La matriz que relaciona ambos puntos recibe el nombre de matriz de rotación.

La matriz de transformación recoge las traslaciones y rotaciones y relaciona los puntos del siguiente modo:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & t_1 \\ \sin \theta & \cos \theta & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.19)$$

Estas transformaciones tienen 3 parámetros: θ , t_1 y t_2 .

Transformación afín

Las transformaciones afines son una generalización de las transformaciones euclídeas que se diferencian de las anteriores en que en este caso las longitudes y los ángulos dejan de conservarse. Estas transformaciones definen, además de transformaciones, rotaciones y reflexiones, escalados y cizalladuras o deformaciones.

Estas transformaciones son las más utilizadas para registro de imágenes y describen las situaciones en las que el UAV vuela más cerca del suelo que en el caso de transformación simple pero más lejos que la transformación proyectiva.

La ecuación que relaciona los puntos de dos imágenes consecutivas a través de una transformación afín es la siguiente:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.20)$$

La matriz de transformación afín se puede definir como A del siguiente modo:

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \quad (3.21)$$

Esta matriz se obtiene a través del producto de dos matrices de rotación con ángulos distintos y una matriz de escalado D , como se muestra a continuación:

$$A = R(\theta)R(-\phi)DR(\phi) \quad (3.22)$$

siendo

$$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \text{ y } D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \quad (3.23)$$

donde λ_1 y λ_2 son los parámetros de escalado, θ el parámetro del ángulo de rotación y ϕ el parámetro del ángulo de deformación. Estas transformaciones tienen 6 parámetros.

Transformación proyectiva

Las transformaciones proyectivas son las más complejas de las tres clases de transformaciones y por consiguiente, las que más tiempo tardan en calcularse y más precisas resultan.

Estas transformaciones son una generalización de las transformaciones afines que definen cambios de perspectiva.

La matriz de homografía, H , que define este tipo de transformación es la siguiente:

$$x' = \mathbf{H}x = \begin{pmatrix} \mathbf{A} & t \\ \mathbf{v}^T & v \end{pmatrix} x \quad (3.24)$$

donde $\mathbf{v} = (v_1, v_2)^T$, es el vector que define los efectos en la perspectiva.

Las transformaciones proyectivas pueden expresarse del siguiente modo:

$$H = H_S H_A H_P = \begin{pmatrix} s\mathbf{R} & \mathbf{t} \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{K} & \mathbf{t} \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{v}^T & v \end{pmatrix} = \begin{pmatrix} \mathbf{A} & t \\ \mathbf{v}^T & v \end{pmatrix} \quad (3.25)$$

donde A es una matriz no singular que se define como $\mathbf{A} = s\mathbf{R}\mathbf{K} + \mathbf{t}\mathbf{v}^T$ y K es una matriz triangular superior con determinante igual a uno.

Este tipo de transformación tiene 8 parámetros.

MSAC (M-estimator Sample and Consensus)

HISTORIA

MSAC es un algoritmo de estimación creado por Torr y Zisserman en 1997 [22] como mejora del algoritmo RANSAC, un algoritmo de estimación muy usado en el ámbito de visión artificial, creado por Fischler y Bolles en 1981 [23].

Ambos algoritmos tienen como objetivo calcular una estimación de la transformación geométrica entre dos imágenes y calcular los puntos correctos (inliers).

Estos algoritmos presentan una extraordinaria capacidad para proporcionar un buen ajuste a partir de datos contaminados con grandes proporciones de datos atípicos (outliers), superiores incluso al 50% que es el límite insalvable para otras técnicas de estimación robusta como LMedianaS [24].

La diferencia entre ambos es que el algoritmo RANSAC trata los datos correctos (inliers) de forma uniforme, o lo que es lo mismo, en la función de coste todos los inliers tienen una penalización con coste cero y los outliers penalizan con coste constante, mientras que el algoritmo MSAC asigna una penalización distinta a los inliers en función de cómo de bien satisfagan la relación funcional requerida y los outliers penalicen con coste constante.

MSAC presenta la ventaja de obtener mejores resultados que RANSAC sin requerir mayor tiempo para realizar su tarea [25].

DESCRIPCIÓN

MSAC es un modelo de estimación robusta que permite hallar un modelo matemático a partir de datos contaminados con numerosos valores que no se ajustan al modelo matemático (datos atípicos) [24].

En concreto, permite en este caso hallar la transformación para la que los datos consistentes en parejas de puntos de dos imágenes consecutivas se ajustan mejor.

ALGORITMO

En primer lugar, el algoritmo MSAC parte de un conjunto de datos emparejados entre dos imágenes consecutivas de los cuales, elige una cantidad de s parejas al azar, de las cuales calcula su transformación a partir de los modelos matemáticos de transformaciones anteriormente mencionados. El número de parejas s son los puntos necesarios para establecer el modelo de transformación elegido.

En segundo lugar, una vez estimada la transformación, el algoritmo trata de buscar posibles correspondencias que apoyan la transformación calculada en el paso anterior. Para ello, se le

aplica la transformación calculada a cada punto de la imagen anterior y se calcula el error que la relaciona con su correspondiente punto de la imagen actual [24].

El coste que producen los puntos viene definido por la siguiente ecuación:

$$C(D; \theta) = \sum_{i=1}^N \rho(\mathbf{p}_i, \theta) \quad (3.26)$$

donde $D = \{\mathbf{p}_1, \dots, \mathbf{p}_N\} \subset \mathbb{R}^2$ es el conjunto de puntos con $\mathbf{p}_i = (x_i, y_i)$, θ es el parámetro buscado y ρ son los pesos de penalización asociado a cada punto y definidos de la siguiente forma:

$$\rho(\mathbf{p}_i, \theta) = \begin{cases} e(\mathbf{p}_i, \theta) & : e(\mathbf{p}_i, \theta) \leq \delta \\ \delta & : \text{resto} \end{cases} \quad (3.27)$$

donde δ es la tolerancia que determina cuándo un píxel se ajusta a un modelo y por tanto, distingue entre datos correctos y datos atípicos.

Seguidamente, se realizan estos pasos de forma iterativa y si el coste es el menor de los obtenidos se guarda el modelo de transformación actual como la transformación estimada. Este algoritmo sigue iterando hasta que el coste obtenido se sitúa por debajo de un umbral predefinido o bien hasta que se alcance un número máximo de iteraciones predefinidas.

Finalmente, los parámetros del modelo estimados con MSAC no suelen ser muy precisos, de modo que para obtener una mayor exactitud se puede recalcular por mínimos cuadrados el modelo a partir de los datos correctos (inliers) del mejor resultado obtenido [26].

3.6. Segmentación del movimiento: Comparación de flujos ópticos real y artificial

La fase de segmentación de movimiento tiene como objetivo determinar cuáles son los píxeles correspondientes a objetos móviles. Para ello en este trabajo se llevará a cabo una comparación de los flujos ópticos real y artificial [27].

En la figura 3.5 se muestra el resultado de superponer el flujo óptico real y el flujo óptico artificial. En general, existe una pequeña diferencia (offset) entre los dos flujos producida por el error en la estimación del movimiento de la cámara. Sin embargo, se puede observar que existen otros píxeles, en los que la diferencia entre ambos flujos es significativa. Estos píxeles son los correspondientes a los objetos móviles en su mayoría, aunque puede quedar algún píxel que todavía sea producto de algún ruido o error.

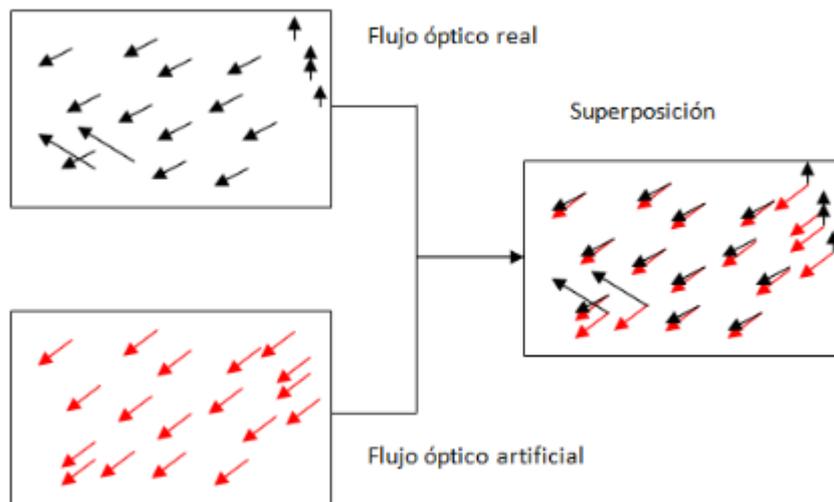


Figura 3.5: Representación ilustrativa del efecto que tiene la superposición de ambos flujos. Figura basada en la presentada en [27].

El procedimiento que se seguirá para realizar la clasificación de estos píxeles es el siguiente:

En primer lugar se calcula la diferencia de ángulos entre los dos flujos. Aquellas píxeles cuya diferencia sea mayor que un determinado umbral, que se ha fijado en 30° , se clasifican como píxeles dinámicos.

En segundo lugar se calcula la diferencia entre los módulos de ambos flujos para los píxeles que no hayan sido clasificados como dinámicos en la comparación de ángulos. Aquellos píxeles cuyo flujo óptico real difiera en un umbral de un 20% con respecto al módulo del flujo óptico artificial serán clasificados como píxeles dinámicos.

El resultado de esta fase son una serie de píxeles clasificados como dinámicos, que tendrán que ser agrupados en objetos en las fases posteriores puesto que algunos de estos píxeles podrían ser píxeles dinámicos aislados producidos por errores o ruido.

3.7. Agrupamiento de píxeles dinámicos: Algoritmo K-means

Esta fase tiene como objetivo agrupar los píxeles dinámicos para formar objetos móviles. Mediante este agrupamiento se consigue eliminar aquellos píxeles clasificados como dinámicos que en realidad son producto de errores o ruido y también reducir la carga computacional del algoritmo puesto que a partir de este momento se almacenarán los objetos conteniendo información únicamente de su tamaño, posición de su centro e intensidad de sus colores. Esto permite reducir información de tal forma que el procesamiento se vuelve mucho más rápido, lo cual sería necesario si se quisiera ejecutar el algoritmo en tiempo real.

El procedimiento llevado a cabo para agrupar píxeles es el siguiente [27]:

Primero se descartan los vectores con módulo largo, puesto que estos pueden estar casados por falsos emparejamientos durante la fase de seguimiento de KTL, lo que genera un flujo óptico real incorrecto. El umbral se ha fijado en el 6% respecto a la longitud de la menor dimensión de la imagen.

Después se eliminan los vectores aislados, por la misma razón que los anteriores. Los vectores que se consideran aislados son aquellos cuya distancia entre su vector más cercano es mayor que el 20% la longitud de la menor dimensión de la imagen.

A continuación se agrupan los píxeles dinámicos en objetos mediante un algoritmo de agrupamiento. En este proyecto se utiliza el algoritmo K-means, que se describirá más adelante. Este agrupamiento se realizará tomando como observaciones las variables multidimensionales $[x, y, v_x, v_y]$ correspondientes a cada uno de los píxeles dinámicos. Para determinar el número de clústeres se realizarán agrupaciones con distintos números de clústeres y se elegirá el número de clúster al que mejor se adapten los datos de las observaciones. De cada clúster se descartarán aquellos vectores que se desvíen más de un valor umbral de la media, que se ha fijado en el 30% respecto del valor promedio.

Por último se descartarán los objetos que tengan menos vectores que un umbral que se ha fijado en 10.

Una vez que se hayan construido todos los objetos se calculará la posición de su centro y su tamaño y se tomarán los valores de intensidad de cada uno de los colores R, G, B, de forma que cada objeto quedará descrito de la siguiente forma: $x_i = [x, y, tamaño, R, G, B]$.

K-means

HISTORIA

K-means es un algoritmo de agrupamiento divisivo, creado por MacQueen en 1967 [28]. Es el algoritmo de agrupamiento más conocido y utilizado debido a que es muy simple y resulta muy eficaz. Sigue un procedimiento simple de clasificación de un conjunto de objetos en un determinado número de K agrupaciones que debe ser definido a priori.

Este método está inspirado y supone una mejora a los anteriores métodos de agrupamiento propuestos por Steinhaus (1956), Lloyd (1957), Forgy/Jancey (1965/66) [29].

DESCRIPCIÓN

K-means es un algoritmo que se traduciría por k-medias en castellano. Se trata de un algoritmo de agrupamiento basado en particiones. Estos algoritmos dividen el conjunto de entrenamiento u observaciones, D, en K particiones.

Su objetivo consiste en dado un número K de particiones del espacio, dividir el conjunto de observaciones D en K grupos que optimicen el criterio de partición, que está basado en minimizar la media ponderada de sus puntos a su centroide [30].

La representación mediante centroides tiene la ventaja de que tiene un significado gráfico y estadístico inmediato.

ALGORITMO

El algoritmo de agrupamiento divisivo de las k -medias parte de un conjunto de observaciones y objetos $D_n = (x_1, x_2, \dots, x_n)$, para todo i , x_i reales y k , v_i , los centros de las K agrupaciones.

El algoritmo se realiza en 4 pasos [31]:

En primer lugar, se debe definir el número de agrupaciones, K , en las que se desea dividir las observaciones, y se eligen aleatoriamente K objetos que forman las K agrupaciones iniciales. Para cada una de las K agrupaciones, se toma como valor inicial del centro, la observación escogida; es decir, x_i . Los únicos objetos de D_n pertenecientes a agrupaciones en este momento son x_i .

En segundo lugar, se reasignan los objetos a las agrupaciones. A cada objeto x_i se le asigna la agrupación a la que se encuentra más próximo de acuerdo a una medida de distancia que suele ser la media euclidiana.

En tercer lugar, una vez que todos los objetos han sido asignados a una agrupación, se recalculan los centros de las K agrupaciones, llamados baricentros.

Por último, se repiten las etapas 2 y 3 hasta que no haya más reasignaciones; es decir, hasta que se alcance la convergencia del algoritmo. Sin embargo, la solución obtenida no se asegura que sea óptima, puesto que los datos que forman inicialmente las agrupaciones son elegidos al azar, y el resultado depende en gran medida de cómo hayan sido elegidos dichos datos. Para mejorar el resultado, se suele repetir varias veces el algoritmo, de forma que los puntos iniciales serán distintos en cada repetición por haber sido elegidos aleatoriamente, y la solución que se elegirá será aquella que proporcione mejores resultados.

Es necesario establecer un número máximo de agrupaciones posibles puesto que si fuera posible agrupar en cualquier cantidad, se tendería a la situación en la que cada agrupación estuviera constituida por un solo vector o un número muy bajo de éstos. En la implementación realizada se ha fijado un número máximo de 8 agrupaciones puesto que se espera que el número de objetos en movimiento presentes en los vídeos sea menor de esa cantidad.

En la figura 3.6 se muestra un ejemplo de agrupamiento obtenido mediante el método de las k -medias:

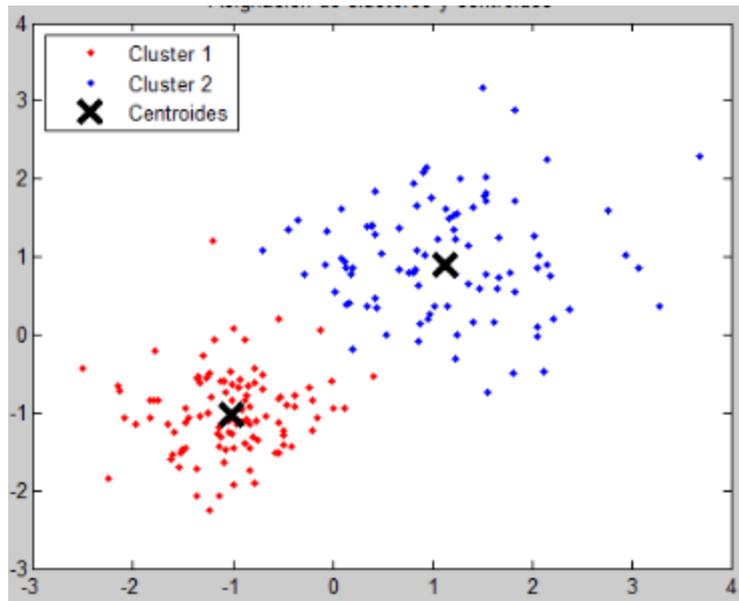


Figura 3.6: Ejemplo de agrupamiento divisivo mediante k-medias en dos agrupaciones.

3.8. Seguimiento de objetos: Algoritmo basado en seguimiento de componentes conectados y filtro de Kalman

La fase de seguimiento de objetos tiene como objetivo realizar un filtrado espacial y temporal de los objetos que se consideran móviles.

De esta forma se consigue distinguir entre objetos que realmente se mueve y otros objetos que son ruido causado por las etapas anteriores o también objetos que vibran, como por ejemplo un arbusto que se mueve ondeado por el viento.

Para que los objetos sean considerados como tales, deben poder ser seguidos durante un número mínimo de 5 cuadros y tener un desplazamiento mayor que un umbral fijado en un 4% respecto a la longitud de la dimensión menor de la imagen. De este modo se distingue entre dos tipos de objetos: objetos que han sido seguidos durante un tiempo mínimo y que han recorrido una distancia mínima y objetos que han sido detectados pero no han sido seguidos suficiente tiempo y espacio. El primer tipo de objetos será marcado con un rectángulo que defina su posición y tamaño en el vídeo, y el segundo tipo de objetos será añadido a la lista de objetos pero todavía no se representará ni se tomará como verdadero objeto móvil.

Todos los objetos que hayan sido detectados en las fases anteriores se almacenarán en una lista de objetos, que se actualizará en cada iteración y cuyo seguimiento se realizará mediante el filtro de Kalman, que se explicará más adelante en profundidad.

En principio, el filtro de Kalman es un algoritmo que permite estimar la posición del estado de los objetos en el siguiente cuadro y cuando se realicen las mediciones de dicho cuadro tomará realimentaciones para corregir el error que tuviera. De este modo, existirá una lista, L_g , que

contendrá los objetos que han sido detectados en algún instante y L_a , los objetos que han sido detectados en este preciso instante y son el resultado de las fases anteriores del algoritmo.

Será necesario un algoritmo de emparejamiento que haga corresponder los objetos de L_a con los de L_g siguiendo como criterio de emparejamiento la minimización de la distancia de Mahalanobis entre los objetos de ambas listas.

Algoritmo de seguimiento basado en componente conectados

Este tipo de seguimiento tiene como objetivo seguir a través de los sucesivos cuadros, unos objetos definidos únicamente por la posición de su centro, su tamaño y la intensidad de su color. De esta forma se facilita que los cálculos sean rápidos y que el almacenamiento de los objetos ocupe poco. Sin embargo, tiene como inconveniente que si el número de objetos es alto, los objetos podrían quedar fusionados en un sólo componente conectado como consecuencia de la oclusión o de otros factores. Sin embargo, si el número de objetos a detectar es bajo, esta solución es la más sencilla de implementar.

Debido a que los vídeos tomados por los UAVs se espera que contengan pocos objetos en movimiento, este es el método más simple para realizar el seguimiento.

El algoritmo puesto en práctica en este proyecto se basa en la existencia de dos listas: L_a y L_g .

La lista L_a contiene los objetos que ha sido detectados en este cuadro y la lista L_g contiene todos los objetos que han sido detectados alguna vez. Los objetos de estas dos listas tienen que ser emparejados para determinar qué objetos de L_g están presentes en este cuadro.

Se pueden dar las siguientes situaciones:

- Ambas listas están vacías, lo que significa que ningún objeto ha sido detectado en ningún instante y tampoco en este cuadro. Esto sucede al principio del algoritmo, puesto que una vez que se detecta algún objeto, L_g dejará de estar vacía.
- L_g contiene objetos pero L_a está vacía, lo que corresponde a la situación en que han sido detectados objetos en instantes anteriores pero en este cuadro no se ha detectado ningún objeto. Esto se debe a que los objetos o bien han salido de la escena o bien han sido tapados por otros objetos.
- L_a contiene objetos pero L_g está vacía, lo que significa que los objetos detectados en este cuadro son los primeros objetos que se han detectado desde que se ha puesto en marcha la herramienta.
- Ambas listas contienen objetos, lo que significa que en este cuadro se han detectado objetos, pero pueden darse varias situaciones distintas en este caso:
 - Los objetos que se han detectado en este cuadro, ya han sido detectados anteriormente, y por tanto, forman parte de la lista L_g .
 - Los objetos detectados en este cuadro no han sido detectados anteriormente, lo que significa que a los objetos de L_g se les ha perdido el rastro y estos nuevos objetos deben ser añadidos a la lista de objetos globales, L_g .
 - Parte de los objetos detectados corresponden a objetos que ya habían sido detectados anteriormente, y que por consiguiente ya se encontraban en la lista L_g y

los restantes objetos de L_a son nuevos objetos que no habían sido detectados anteriormente.

Estas distintas situaciones deben ser tenidas en cuenta para garantizar que el seguimiento de los objetos es correcto y que no existen interferencias entre los objetos seguidos.

Por ello, es necesario establecer un umbral de distancia entre los objetos de la lista para determinar si los objetos emparejados de cada una de las listas pertenecen efectivamente al mismo objeto o por el contrario, no se trata del mismo objeto y se trata por un lado de un nuevo objeto detectado y por otro lado de un objeto existente en L_g al que se le ha perdido el rastro.

La medida que se usará para calcular la distancia entre los objetos de L_a y L_g es la distancia de Mahalanobis, cuya ecuación es la siguiente:

$$d(\hat{x}_{k|k-1}^i, \hat{z}_k^j) = \sqrt{\sum_{v=1}^6 \frac{(\hat{x}_{k|k-1}^i(v), z_k^j(v))^2}{P_{k|k-1}^i(v)}} \quad (3.28)$$

donde $x_i = [x, y, tamaño, R, G, B]$ son los objetos almacenados en la lista global de objetos, L_g , $z_i = [x, y, tamaño, R, G, B]$ son los objetos almacenados en la lista de objetos actuales, L_a , los superíndices i y j hacen referencia a la posición que ocupan los objetos en las listas L_g y L_a , respectivamente, los subíndices $k-1$ y k , hacen referencia al cuadro anterior y actual, respectivamente, y el valor v corresponde a las magnitudes $x, y, tamaño, R, G, B$. El acento circunflejo de la x indica que se trata de estimaciones y P es la varianza de cada variable v obtenida durante el proceso del filtro de Kalman.

Como se puede comprobar en la ecuación anterior, la lista L_a contiene los estados de los objetos detectados en el momento actual, que corresponden a lo que podría llamarse "mediciones", y la lista L_g contiene los estados correspondientes a cada objeto detectado en algún momento así como una estimación de los estados que tendrán los objetos en el siguiente cuadro, lo que se denomina "estimaciones". Estas estimaciones se realizan con ayuda del filtro de Kalman.

El pseudocódigo utilizado para realizar los emparejamientos está basado en el presentado en [27] y se muestra a continuación:

Algoritmo 1: Emparejamiento de objetos de las listas

Se necesita: Lista de objetos actuales (L_a) con n_a objetos

Se necesita: Lista de objetos actuales (L_g) con n_g objetos

Asegurarse: Las dos listas contienen objetos ($n_a > 0 \wedge n_g > 0$)

```

Preparar  $n_a \times n_g$  pares: MatrizDistancias
Preparar  $n_a \times n_g$  pares: PosiblesParejas
MatrizEmparejamientos
VectorObjetosNoDetectadosEstaVez
VectorObjetosNuevos
VectorObjetosMalEmparejados
Calcular la distancia de Mahalanobis entre objetos de las dos listas
Lista de parejas seleccionadas: ParejasSeleccionadas
mientras NúmeroEmparejamientos < min([ $n_a$ ,  $n_g$ ])
    Seleccionar pareja con menor distancia
    si Alguno de los objetos ya se encuentra en ParejasSeleccionadas entonces
        Poner la distancia máxima en dicha posición en MatrizEmparejamientos
    y si
        Incluirlo en ParejasSeleccionadas
        Poner la distancia máxima en dicha posición en MatrizEmparejamientos
    fin si (ya se han acabado los objetos de alguna de las listas)
fin mientras (ya se han acabado los objetos de alguna de las dos listas)
si  $n_a > n_g$  (Hay objetos nuevos) entonces
    Añadir los objetos sobrantes en  $L_a$  a VectorObjetosNuevos
y si  $n_a > n_g$  (Hay objetos nuevos) entonces
    Añadir los objetos sobrantes en  $L_a$  a VectorObjetosNoDetectadosEstaVez
fin si
si la distancia de algún objeto es mayor que el umbral distancia
    Eliminar la pareja
    Añadir objeto de  $L_a$  a VectorObjetosNuevos
    Añadir objeto de  $L_g$  a VectorObjetosNoDetectadosEstaVez
fin si

```

Filtro de Kalman

HISTORIA

El filtro de Kalman es un algoritmo que proporciona una herramienta para solucionar problemas de filtrado lineal de datos discretos de forma recursiva. Este algoritmo fue publicado por R.E Kalman en 1960 [32]. Desde entonces, debido en gran medida a los avances en computación digital, el filtro de Kalman ha sido objeto de una gran investigación y aplicación especialmente en el área de la navegación autónoma o asistida [33].

DESCRIPCIÓN

El filtro de Kalman es un conjunto de ecuaciones matemáticas que proporcionan un modo eficiente de cálculo recursivo para la estimación del estado de un proceso con el objetivo de minimizar el error en el sentido de mínimos cuadrados. Este filtro es muy potente en varios aspectos puesto que permite realizar estimaciones de estados del pasado, del presente e incluso del futuro y permite realizar dichas estimaciones incluso cuando se desconoce la naturaleza del sistema del cual se quieren estimar sus estados [33].

ALGORITMO

La descripción del algoritmo y las imágenes que se presentan y ecuaciones que se presentan en esta sección se basan en el artículo [33] de Greg Welch y Gary Bishop.

El filtro de Kalman tiene como objetivo estimar el estado de un proceso que se puede definir mediante la siguiente ecuación estocástica [32] [34]:

$$x_k = A \cdot x_{k-1} + B \cdot u_{k-1} + w_{k-1} \quad (3.29)$$

donde x_k es el proceso $x \in \mathbb{R}^n$ en el estado k , del cual aún no se han realizado mediciones y se puede estimar a partir del mismo proceso en el instante anterior, $k-1$, instante en el que sí se han realizado mediciones. u_{k-1} es una entrada de control opcional y w_{k-1} es un proceso de ruido. A y B son dos matrices que contienen las variables de entorno que definen el problema.

En el caso concreto del problema de este proyecto la ecuación quedaría de la siguiente forma:

$$\begin{pmatrix} x_n \\ y_n \\ t_n \\ R_n \\ G_n \\ B_n \end{pmatrix} = \begin{pmatrix} 1 & & \dots & 0 \\ & 1 & & \\ & \vdots & \ddots & \vdots \\ & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ t_{n-1} \\ R_{n-1} \\ G_{n-1} \\ B_{n-1} \end{pmatrix} + \begin{pmatrix} 1 & & \dots & 0 \\ & 1 & & \\ & \vdots & \ddots & \vdots \\ & 0 & \dots & 1 \end{pmatrix} \cdot \begin{pmatrix} v_x \cdot \Delta t \\ v_y \cdot \Delta t \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} n_x \\ n_y \\ n_t \\ n_R \\ n_G \\ n_B \end{pmatrix} \quad (3.30)$$

Se puede observar que el proceso $x = (x_n, y_n, t_n, R_n, G_n, B_n)'$ corresponde a los valores de posición, tamaño y colores RGB de cada objeto en el instante n . El proceso de ruido está definido por $x = (n_x, n_y, n_t, n_R, n_G, n_B)'$; la entrada de control es $u_{k-1} = (v_x \cdot \Delta t, v_y \cdot \Delta t, 0, 0, 0, 0)'$, donde v_x y v_y son las velocidades horizontal y vertical de cada objeto y Δt es el tiempo transcurrido entre cada cuadro; y la matriz de variables de entorno A es la matriz identidad, puesto que las diferentes variables de cada estado son independientes entre sí.

La ecuación 3.29 se puede expresar de forma homogénea para que sea más cómodo su manejo, como se muestra en la ecuación 3.31:

Las ecuaciones vistas anteriormente pertenecían a la fase de estimación del proceso. La fase de medición está definida por la ecuación (3.34):

$$z_k = Hx_k + v_k \quad (3.34)$$

donde z_k representa las mediciones de $z \in \mathbb{R}^m$ en el instante k y v_k el ruido en las mediciones. La matriz H es la matriz identidad puesto que las distintas variables de cada estado no están relacionadas entre sí.

Las variables aleatorias w_k y v_k , que representan el ruido del proceso y de las mediciones respectivamente, se suponen independientes entre sí, blancas y con distribuciones de probabilidad gaussianas:

$$p(w) \sim N(0, Q)$$

$$p(v) \sim N(0, R)$$

En la práctica, la covarianza del ruido del proceso, Q , y la covarianza del ruido de la medición, R , pueden cambiar en cada instante; sin embargo, por simplicidad se pueden suponer constantes.

Para fijar el valor inicial de Q y de R , en este proyecto se han tomado unos valores iniciales relacionados con cada una de las imágenes tomadas.

En concreto, el valor de Q se toma del siguiente modo: Se calcula para los 10 primeros cuadros del vídeo que se desee analizar, la varianza de cada uno de los tres colores, R, G y B y se toma el 80% de dicho valor para inicializar las variables de n_R , n_G y n_B respectivamente. Para inicializar las variables de n_x y n_y se toma un valor de 2km/h, que es la velocidad que se estima que puede llevar una persona caminando, puesto que las personas serán los objetos móviles de los vídeos y se modifica dicho valor en función de la escala a la que estén tomadas las imágenes, para hacer corresponder velocidades reales de los objetos sus respectivos flujos ópticos calculados mediante el desplazamiento en píxeles de la imagen y el tiempo, que está relacionado con la tasa de los vídeos. El valor de n_t se toma como el 30% del tamaño del vídeo.

El valor de R , por su parte se toma como un valor 10 veces superior al calculado para Q .

Las ecuaciones del filtro de Kalman, tienen un origen probabilístico, tal y como se describe a continuación:

Sea $\hat{x}_k^- \in \mathbb{R}^n$ la estimación a priori en el instante k a partir del conocimiento que se tiene sobre dicho objeto en instantes anteriores y $\hat{x}_k \in \mathbb{R}^n$ la estimación a posteriori en el instante k , una vez se tiene conocimiento acerca de la medición z_k . Se pueden definir los errores de estimación a priori y a posteriori como

$$\hat{e}_k^- \equiv x_k - \hat{x}_k^-$$

$$e_k \equiv x_k - \hat{x}_k$$

La covarianza del error a priori y a posteriori son entonces, respectivamente

$$P_k^- = E[e_k^- e_k^{-T}]$$

$$P_k = E[e_k e_k^T]$$

Para deducir las ecuaciones del filtro de Kalman, se trata de buscar una ecuación que calcule la estimación a posteriori de un estado \hat{x}_k como una combinación lineal de la estimación a priori de dicho estado \hat{x}_k^- y una diferencia ponderada entre la medición real z_k y la medición que se había predicho $H\hat{x}_k^-$. La ecuación que se obtiene es la siguiente:

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-) \quad (3.35)$$

La diferencia $(z_k - H\hat{x}_k^-)$ se denomina "innovación en la medición" y refleja la discrepancia entre la medición predicha $H\hat{x}_k^-$ y la medición real z_k .

K , por su parte es la ganancia que minimiza el error de covarianza a posteriori. Existen varias representaciones distintas para K , pero una de las más habituales es la siguiente:

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} = \frac{P_k^- H^T}{H P_k^- H^T + R} \quad (3.36)$$

El algoritmo que sigue el filtro de Kalman, consiste fundamentalmente en realizar dos fases distintas: una de predicción que permita estimar el estado de un sistema a partir de las mediciones de los estados anteriores y una fase de medición que permita determinar el valor real del estado de los objetos y sirva de realimentación al sistema para corregir los errores que se produzcan. Cada una de estas dos fases tiene unas ecuaciones asociadas de modo que la predicción y la corrección se realicen de forma cíclica como se muestra el esquema de la figura 3.7:

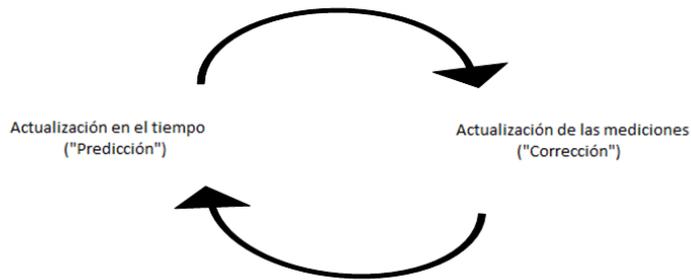


Figura 3.7. Esquema cíclico del funcionamiento del algoritmo del filtro de Kalman. La actualización en el tiempo proyecta el estado actual en el siguiente instante de tiempo. La actualización de la medida corrige la estimación proyectada con ayuda de la medición real en ese instante.

Las ecuaciones de actualización en el tiempo y actualización de medición se muestran en las tabla 3.1-3.2:

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q\end{aligned}$$

Tabla 1.1. Ecuaciones de actualización en el tiempo del filtro de Kalman discreto.

$$\begin{aligned}K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k &= (I - K_k H)P_k^-\end{aligned}$$

Tabla 1.2. Ecuaciones de actualización de las mediciones del filtro de Kalman discreto.

Estas fases y sus ecuaciones se pueden expresar conjuntamente como se muestra en la figura 3.8:

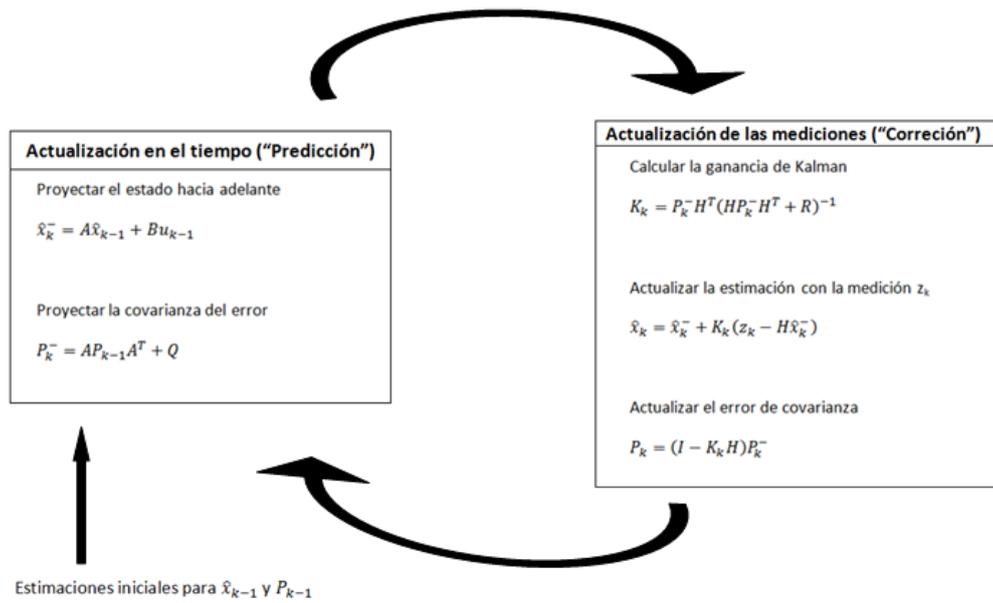


Figura 3.8. Imagen completa de la operación del filtro de Kalman, que combina el diagrama de nivel alto de la figura 3.7 con las ecuaciones de las tablas 1.1 y 1.2.

Capítulo 4: Pruebas y resultados experimentales

4.1. Bases de datos utilizadas

La base de datos utilizada para la realización de las pruebas experimentales se compone de dos bancos de vídeos diferentes.

En primer lugar consta de un primer banco de vídeos grabados por una cámara de vídeo marca Samsung y modelo Schneider Kreuznach. Estos vídeos han sido tomados desde un segundo piso apuntando hacia la calle. El fondo de la imagen contiene texturas correspondientes a los diferentes adoquines de la acera. Los vídeos han sido tomados con diferentes condiciones de iluminación. Estos vídeos tienen una duración aproximada de entre 10 y 15 segundos cada uno y una resolución de 1280 x 720 px y formato mp4. En estos vídeos los objetos están suficientemente lejos como para que se detecten los objetos en movimiento que aparecen en escena, pero no lo suficiente como para que el tamaño calculado sea exacto. Esto se debe a que la distancia a la que se encuentra la cámara hace que los objetos sean grandes y se formen varias agrupaciones en un mismo objeto real.

La segunda base de datos utilizada consiste en vídeos grabados por un UAV marca Parrot modelo AR Drone 2 de dimensiones 77.7 x 38.3 x 12.5 mm y cuyo peso es de 31 g. Los vídeos tomados tienen una resolución de 1928 x 1080 px, una duración de entre 10 y 15 s y formato mp4. Estos vídeos han sido grabados en un espacio cerrado de un polideportivo, de modo que la distancia a los objetos en movimiento es pequeña y la grabación no tiene vista aérea sino que el UAV vuela a la misma altura que los objetos en movimiento, lo que hace que dichos objetos no sean detectados correctamente puesto que existen efectos de perspectiva en los vídeos.

4.2. Descripción de las pruebas experimentales

En este capítulo se realizarán una serie de pruebas experimentales para comprobar el funcionamiento de la herramienta realizada. Las pruebas experimentales que se realizarán estarán basadas en las realizadas por Gonzalo R. Rodríguez-Canosa, Stephen Thomas, Jaime del Cerro, Antonio Barrientos, Bruce MacDonald [27]. Se ha decidido llevar a cabo estas pruebas puesto que el proyecto está basado fundamentalmente en dicho trabajo.

Las pruebas principales que se realizarán, por tanto, consistirán en la comprobación visual del funcionamiento de la máquina. Para ello se producirá a partir de un vídeo de entrada, un vídeo de salida idéntico al vídeo de entrada salvo por los recuadros introducidos en dicho vídeo que están asociados a los objetos en movimiento y pretenden situarse conteniendo a dichos

objetos. De este modo, se podrá comprobar si la herramienta cumple con la función de detectar y seguir a los objetos en movimiento durante el transcurso del vídeo.

Además se realizarán unas pruebas añadidas, que realizan un estudio cuantitativo comparativo entre el rendimiento alcanzado por la herramienta realizada en comparación con una herramienta que no realice comparación de flujos ópticos. En este sentido, se pretende demostrar que el método de comparación de flujos ópticos basado en el cálculo de un flujo óptico artificial permite deshacer el movimiento de la cámara y es adecuado a la situación de videovigilancia desde cámaras en movimiento.

Las secciones siguientes detallan cada una de las pruebas.

4.2.1. Análisis cualitativo del funcionamiento de la herramienta

Para comprobar cómo funciona la herramienta se ha llevado a cabo un análisis cuantitativo de los datos, que consiste en introducir un vídeo en la herramienta y comprobar si el video procesado por ésta detecta y sigue los objetos en movimiento que aparecen en la escena. Para ello, se ha decidido dibujar un recuadro situado en el centroide del objeto con valores idénticos de longitudes de anchura y altura. De esta forma se puede comprobar fácilmente si el objeto en movimiento se encuentra enmarcado en un recuadro lo que indica que ha sido detectado correctamente.

Los vídeos procesados que se presentan a continuación se han grabado con diferentes tipos de iluminación y en diferentes entornos.

El análisis se centra en los siguientes casos: cámara fija con objeto en movimiento, cámara en movimiento con objeto en movimiento y cámara en movimiento sin objetos en movimiento.

Por último se presentan los resultados obtenidos con los vídeos grabados por el UAV y se analizan los efectos que se producen cuando el UAV no vuela a una altura grande y paralelo al suelo.

Cámara fija con objeto en movimiento

El análisis del funcionamiento de la herramienta para el caso de cámara fija con objeto en movimiento se ha realizado a partir de un vídeo extraído del banco de vídeos grabado por la cámara de vídeo Samsung. Se ha observado que los resultados son sensiblemente mejores cuando se realiza un diezmado del vídeo. Esto se debe a que si la cámara está fija o se mueve muy lentamente, las vibraciones de los objetos correspondientes a objetos sin movimiento presentes en la escena son suficientemente grandes entre dos cuadros consecutivos como para ser tomadas como objeto en movimiento. Esto produce errores en la detección de objetos en forma de falsos positivos que afectan negativamente en el funcionamiento de la herramienta. Sin embargo los vídeos diezmados sí que logran realizar la detección y seguimiento de los objetos como se muestra en la figura 4.1:



Figura 4.1: Representación de cuatro cuadros consecutivos donde se ha producido el seguimiento de un objeto con cámara fija.

En la figura 4.1 se puede observar que el objeto el movimiento ha sido detectado y seguido. Sin embargo el tamaño del objeto no se ha calculado de forma exacta. Esto se debe a que los objetos grandes son tomados en ocasiones como varios objetos diferentes o sólo es detectada una parte de ellos. En esta imagen concretamente la detección se ha producido en las piernas de la persona puesto que es la región del objeto que tiene mayor velocidad.

También se observa que la sombra, aunque se encuentra en movimiento no ha sido detectada. Se presenta aquí un compromiso de diseño que consiste en aumentar la sensibilidad de la herramienta para detectar mejor los objetos, pero que produciría errores correspondientes a falsas detecciones.

En esta figura también se puede observar que los cambios de iluminación provocan vibraciones en la imagen. En este caso se aprecia en el centro de la imagen unas vibraciones en el suelo producidas por la luz de una farola. Mediante el diezmo se consigue reducir estas vibraciones de tal forma que no sean consideradas como objetos en movimiento

Cámara en movimiento con objeto en movimiento

El análisis del funcionamiento de la herramienta para el caso de cámara en movimiento con objetos en movimiento se ha realizado a partir de un vídeo extraído del banco de vídeos grabado por la cámara de vídeo Samsung. Debido al movimiento de la cámara no se requiere diezmar este vídeo. En la figura 4.2 se presentan las imágenes correspondiente al seguimiento de objetos detectados:

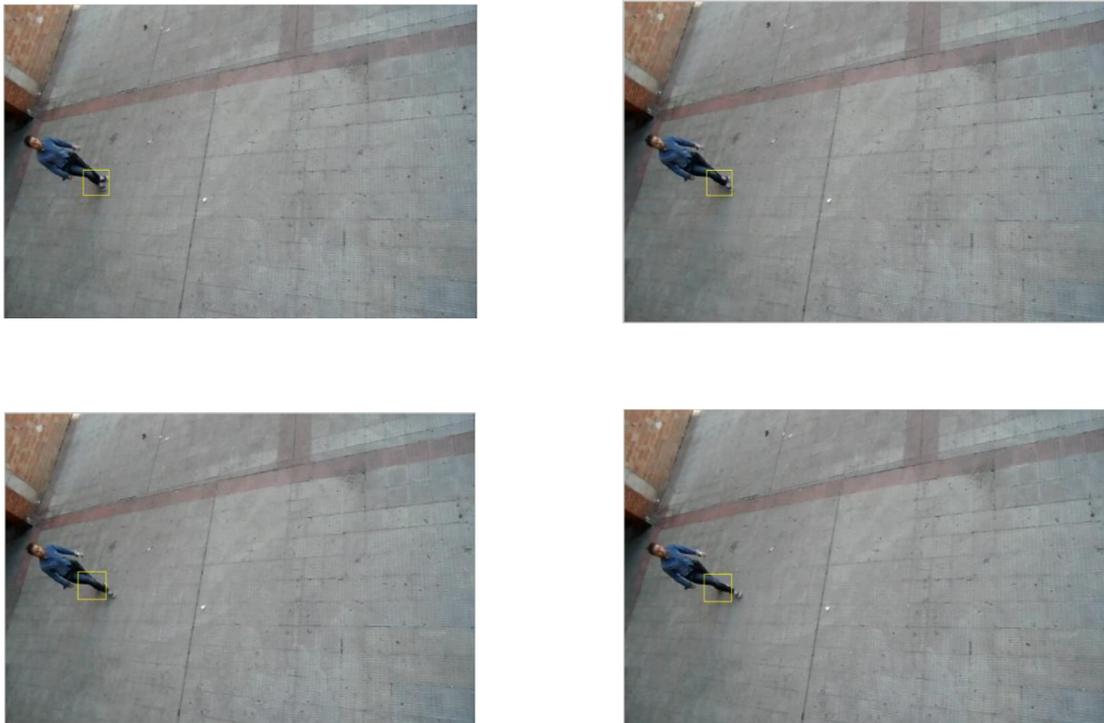


Figura 4.2: Representación de cuatro cuadros consecutivos donde se ha producido el seguimiento de un objeto con cámara en movimiento.

En la figura 4.2 se puede observar que el objeto el movimiento ha sido detectado y seguido. Sin embargo el tamaño del objeto no se ha calculado de forma exacta. Esto se debe a que los objetos grandes son tomados en ocasiones como varios objetos diferentes o sólo es detectada una parte de ellos. En esta imagen concretamente la detección se ha producido en las piernas de la persona puesto que es la región del objeto que tiene mayor velocidad.

Además la cámara no está paralela al suelo, luego se producen ciertas variaciones en la perspectiva que hacen que el recuadro no contenga al objeto en su totalidad.

En esta figura también se puede observar que la iluminación es muy favorable, lo que hace que no se produzcan tantas vibraciones en la escena. Esto influye en que no sea necesario un diezmo del vídeo.

Cámara en movimiento sin objetos en movimiento

El análisis del funcionamiento de la herramienta para el caso de cámara en movimiento con objetos en movimiento se ha realizado a partir de un vídeo extraído del banco de vídeos grabado por el UAV. En la escena no aparece ningún objeto en movimiento; sin embargo sí aparecen varios objetos fijos que en ocasiones se han detectan como objetos y es la fase de filtrado la encargada de eliminarlos.

En la figura 4.3 se muestran un ejemplo en el que se ha detectado un falso positivo y la etapa de filtrado lo ha eliminado.

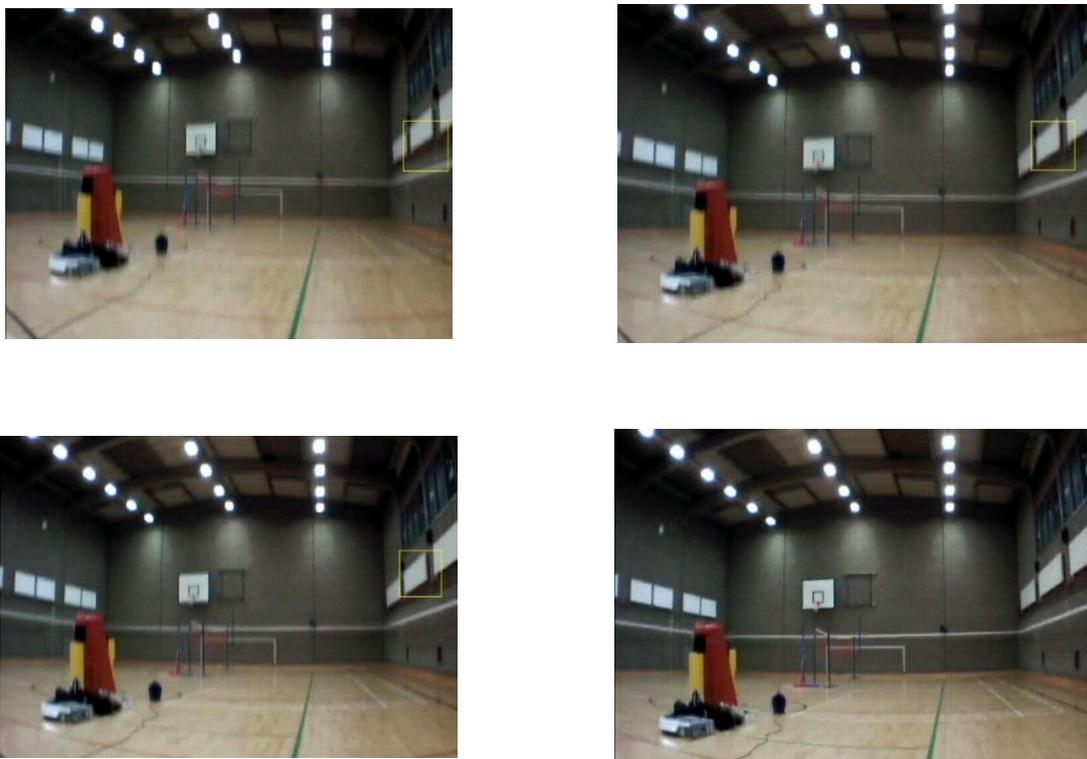


Figura 4.3: Representación de cuatro cuadros consecutivos. Los tres primeros representan la detección de un falso positivo y el cuarto su corrección mediante la etapa de filtrado.

En la figura 4.3 se puede observar que los escenarios donde el fondo contiene muchas texturas e incluso objetos sin movimiento, en ocasiones se producen falsas detecciones, que son corregidas durante la etapa de filtrado.

Cámara en movimiento con objetos en movimiento (vídeo realizado por el UAV)

Los vídeos realizados con el UAV se han realizado dentro de un recinto cerrado de un polideportivo, como se ha mencionado anteriormente. Por este motivo, los objetos no son detectados correctamente. La figura 4.4 presenta los efectos que se producen cuando la cámara se acerca mucho al objeto en movimiento y no lo graba desde arriba sino que lo graba frontalmente.

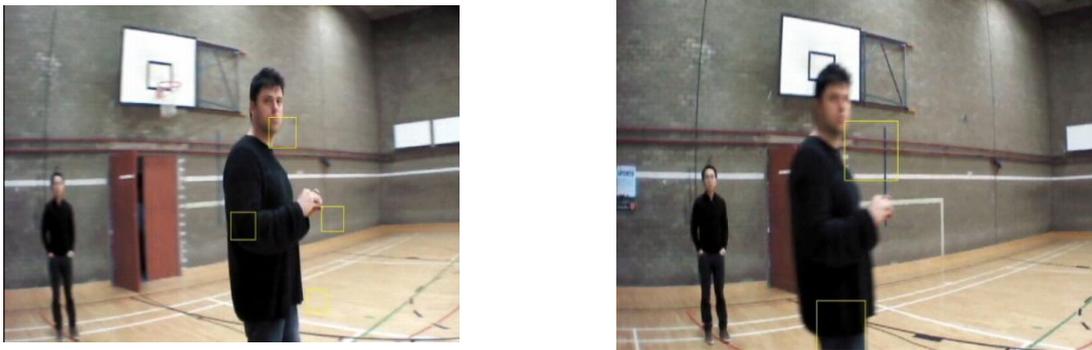


Figura 4.4: Representación de dos cuadros de un vídeo grabado por un UAV con objetos en movimiento a una distancia muy corta de éstos. La figura ilustra los efectos que se producen debido a la corta distancia entre el objeto en movimiento y la cámara.

En la figura 4.4 se observan los efectos que se producen cuando la cámara en movimiento está cerca del objeto que se quiere detectar y no está por encima de éste y paralelo al suelo. En primer lugar se puede comprobar que sí que se producen detecciones del objeto, puesto que el único objeto que realmente se encontraba en movimiento, correspondiente a la persona que se muestra en primer plano, ha sido detectado. Sin embargo ha sido detectado como varios objetos diferentes debido que se trata de un objeto muy grande en comparación con la escena. De este modo queda demostrado que para que los resultados sean buenos, la cámara debe estar más alejada de los objetos.

4.2.2. Análisis cuantitativo de la mejora que supone el método de comparación de flujos ópticos

A continuación se lleva a cabo un análisis cuantitativo del funcionamiento de la herramienta. Este análisis tiene como objetivo poner de manifiesto la mejora que supone realizar una comparación entre flujo óptico real y flujo óptico artificial en el rendimiento de la herramienta.

Para ello se ha ejecutado la herramienta con el mismo vídeo en dos situaciones diferentes. La primera situación consiste en que la herramienta implemente la fase de comparación de flujos ópticos real y artificial. Y la segunda situación consiste en hacer funcionar la herramienta obviando el cálculo de flujo óptico artificial y detectando como píxeles dinámicos todos aquellos cuyo vector de flujo óptico real sea mayor que 0.

Las pruebas se han realizado para un vídeo grabado con cámara en movimiento con un objeto en movimiento.

Para este vídeo se han realizado las siguientes medidas [35]:

- **Verdadero negativo, VN:** Número de cuadros donde el patrón de oro y los resultados del sistema coinciden en la ausencia de cualquier objeto.
- **Verdadero positivo, VP:** Número de cuadros donde el patrón de oro y los resultados del sistema coinciden en la presencia de uno o más objetos.
- **Falso negativo, FN:** Número de cuadros donde el patrón de oro contiene al menos un objeto que el sistema no contiene o que no coincide con la posición de los objetos del patrón de oro.
- **Falso positivo, FP:** Número de cuadros donde los resultados del sistema contienen al menos un objeto o ninguno de los objetos del patrón de oro está contenido dentro de un recuadro.

Debido a que los resultados en los vídeos son aleatorios como consecuencia de la aplicación de los algoritmos MSAC y k-means, se han realizado 7 repeticiones del vídeo, para realizar un promedio de éste y calcular también su varianza y su valor cuadrático medio (V.C.M).

A partir de las mediciones, el objetivo consiste en medir el rendimiento de la herramienta. Para ello utilizan las siguientes ecuaciones:

$$Tasa\ de\ detección\ y\ seguimiento\ (TDS) = \frac{VP}{PO}$$

$$Tasa\ de\ falsa\ alarma\ (TFA) = \frac{FP}{VP + FP}$$

$$Tasa\ de\ detección\ (TD) = \frac{VP}{VP + FN}$$

$$Especificidad\ (Espec.) = \frac{VN}{FP + VN}$$

$$Exactitud\ (Exact.) = \frac{VP + VN}{CT}$$

$$Predicciones\ positivas\ (PP) = \frac{VP}{VP + FP}$$

$$Predicciones\ negativas\ (PN) = \frac{VN}{FN + VP}$$

$$\text{Tasa de falsos negativos (TFN)} = \frac{FN}{FN + VP}$$

$$\text{Tasa de falsos positivos (TFP)} = \frac{FP}{FP + VN}$$

PO es el patrón de oro, que representa el número de cuadros totales para el patrón de oro de los objetos.

CT es el número de cuadros totales analizados.

A continuación se presentan los resultados obtenidos:

	Falso positivo	Falso negativo	Verdadero positivo	Verdadero negativo	Cuadros totales	Patrón de oro
	3	236	90	0	329	329
	3	220	106	0	329	329
	0	199	130	0	329	329
	0	191	138	0	329	329
	6	238	85	0	329	329
	10	219	100	0	329	329
	1	250	78	0	329	329
Media	3	222	104	0	329	329
Var.	13	456	514	0	0	0

Tabla 4.1. Medidas realizadas para vídeo grabado con cámara fija con objetos en movimiento con comparación de flujos ópticos.

	TDS	TFA	TD	Espec.	Exact.	PP	PN	TFN	TFP
	0,2736	0,0323	0,2761	0,0000	0,2736	0,9677	1,0000	0,7239	1,0000
	0,3222	0,0275	0,3252	0,0000	0,3222	0,9725	1,0000	0,6748	1,0000
	0,3951	0,0000	0,3951	inf	0,3951	1,0000	1,0000	0,6049	inf
	0,4195	0,0000	0,4195	inf	0,4195	1,0000	1,0000	0,5805	inf
	0,2584	0,0659	0,2632	0,0000	0,2584	0,9341	1,0000	0,7368	1,0000
	0,3040	0,0909	0,3135	0,0000	0,3040	0,9091	1,0000	0,6865	1,0000
	0,2371	0,0127	0,2378	0,0000	0,2371	0,9873	1,0000	0,7622	1,0000
Media	0,3157	0,0328	0,3186	inf	0,3157	0,9672	1,0000	0,6814	inf
Var.	0,0047	0,0012	0,0046	inf	0,0047	0,0012	0,0000	0,0046	inf
V.C.M	0,1044	0,0023	0,1061	inf	0,1044	0,9367	1,0000	0,4689	inf

Tabla 4.2. Ecuaciones realizadas para vídeo grabado con cámara fija con objetos en movimiento con comparación de flujos ópticos.

	Falso positivo	Falso negativo	Verdadero positivo	Verdadero negativo	Cuadros totales	Patrón de oro
	0	322	7	0	329	329
	1	314	14	0	329	329
	1	312	16	0	329	329
	1	321	7	0	329	329
	0	317	12	0	329	329
	1	314	14	0	329	329
	1	315	13	0	329	329
Media	1	316	12	0	329	329
Var.	0	14	12	0	0	0

Tabla 4.3. Medidas realizadas para vídeo grabado con cámara fija con objetos en movimiento sin comparación de flujos ópticos.

	TDS	TFA	TD	Espec.	Exact.	PP	PN	TFN	TFP
	0,0213	0,0000	0,0213	inf	0,0213	1,0000	1,0000	0,9787	inf
	0,0426	0,0667	0,0427	0,0000	0,0426	0,9333	1,0000	0,9573	1,0000
	0,0486	0,0588	0,0488	0,0000	0,0486	0,9412	1,0000	0,9512	1,0000
	0,0213	0,1250	0,0213	0,0000	0,0213	0,8750	1,0000	0,9787	1,0000
	0,0365	0,0000	0,0365	inf	0,0365	1,0000	1,0000	0,9635	inf
	0,0426	0,0667	0,0427	0,0000	0,0426	0,9333	1,0000	0,9573	1,0000
	0,0395	0,0714	0,0396	0,0000	0,0395	0,9286	1,0000	0,9604	1,0000
Media	0,0360	0,0555	0,0361	inf	0,0360	0,9445	1,0000	0,9639	inf
Var.	0,0001	0,0019	0,0001	inf	0,0001	0,0019	0,0000	0,0001	inf
V.C.M	0,0014	0,0050	0,0014	inf	0,0014	0,8940	1,0000	0,9292	inf

Tabla 4.4. Ecuaciones realizadas para vídeo grabado con cámara fija con objetos en movimiento sin comparación de flujos ópticos.

La figura 4.5 comparar los resultados mostrados en las tablas 4.2 y 4.4 :

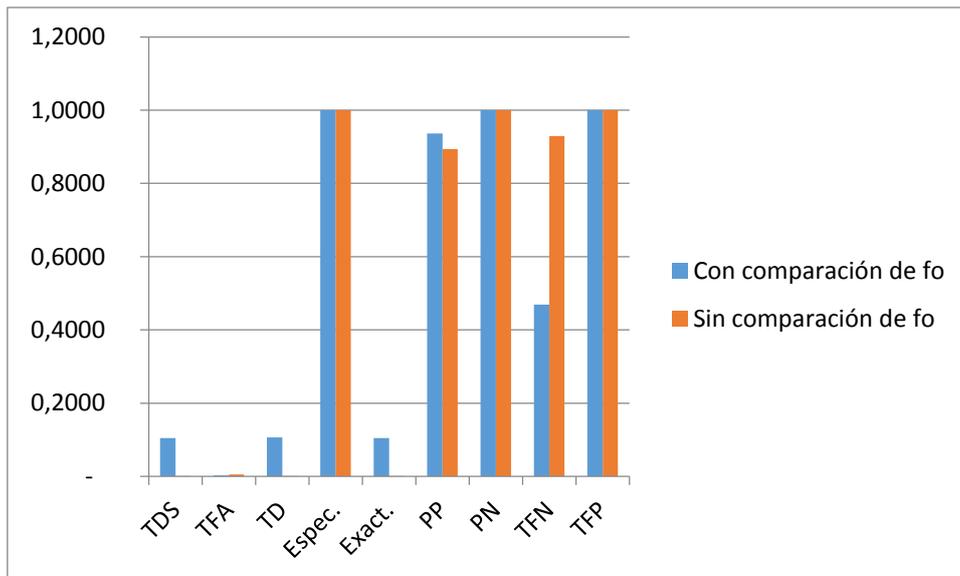


Figura 4.5: Gráfica comparativa del rendimiento de la herramienta para vídeos grabados con cámara en movimiento y objetos en movimiento.

La figura 4.5 muestra un comportamiento claramente superior de la herramienta que implementa cálculo de flujo óptico artificial. Concretamente, la tasa de detección y seguimiento para el caso con flujo óptico artificial supera en un 97,80% a la herramienta sin cálculo de flujo óptico artificial, y además la tasa de falsas alarmas es un 72,00% más baja. En la figura 4.6 se pueden apreciar los resultados correspondientes a la tasa de detección y la tasa de falsas alarmas.

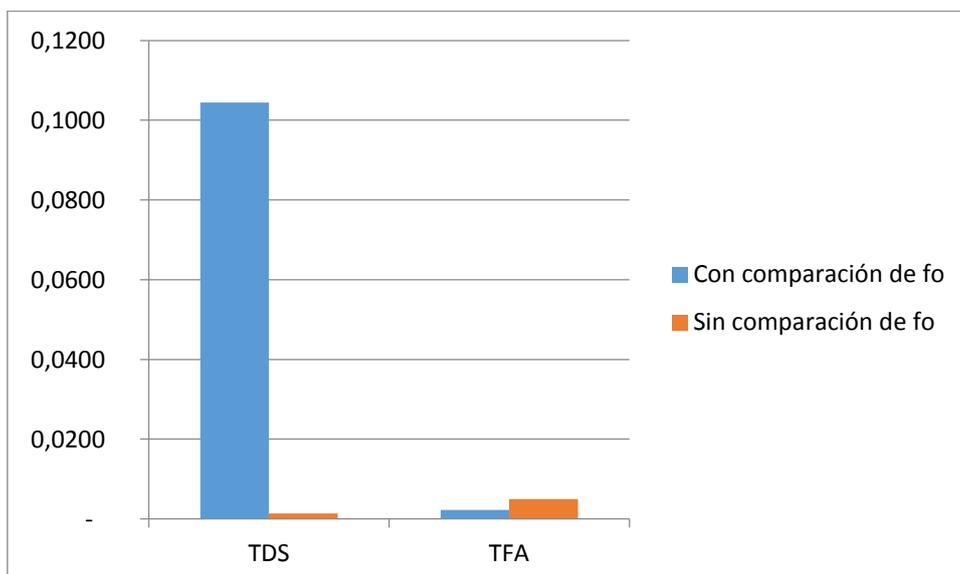


Figura 4.6: Gráfica comparativa del rendimiento de la herramienta para vídeos grabados con cámara en movimiento y objetos en movimiento: Resultados correspondientes a la tasa de detección y la tasa de falsas alarmas.

Capítulo 5: Conclusiones, trabajo futuro y presupuesto

5.1. Conclusiones

El proyecto que se ha llevado a cabo a consistido fundamentalmente en la creación de una herramienta capaz de detectar objetos en movimiento y seguir sus trayectorias desde vehículos aéreos no tripulados (UAVs) en particular y cámaras móviles en general.

La videovigilancia desde objetos en movimiento presenta retos adicionales a los que presentaba la video vigilancia desde cámaras estáticas y por ello se ha buscado un método que resuelva estos retos.

En el estado del arte se ha realizado una descripción bibliográfica que repasaba todos los métodos y algoritmos utilizados en tareas de vigilancia con cámara móvil. Varios de los algoritmos más utilizados para cámara fija no son aplicables para cámara en movimiento luego han tenido que ser descartados.

Los métodos más utilizados en el problema de seguimiento de objetos desde cámaras en movimiento pueden clasificarse en tres grupos que son los que realizan alineamiento global, los que realizan alineamiento por características y los que no realizan alineamiento. Las técnicas que no realizan alineamiento se han descartado en primer lugar puesto que su coste computacional es demasiado alto, lo que impediría perfeccionar la herramienta creada para su ejecución en tiempo real. Los métodos que realizan alineamiento de las imágenes se han considerado, por tanto, más adecuados. Sin embargo, los métodos que realizan alineamiento global también son computacionalmente pesados y requieren un tiempo muy alto de ejecución, de modo, que como uno de los requisitos que se desean de esta herramienta es su rapidez, el método de alineamiento por características es el más adecuado para el propósito de este proyecto.

Una vez decidido el método que se utilizará, se han estudiado los posibles algoritmos que realizan cada una de las fases del proceso, que son las siguientes: extracción de puntos, seguimiento de dichos puntos, alineamiento de la imagen, segmentación del movimiento, agrupamiento en objetos y seguimiento de dichos objetos. Para cada una de las fases de la herramienta, se han estudiado las diferentes opciones existentes y sus algoritmos, y se han escogido los métodos siguientes:

La extracción de puntos se realiza mediante el algoritmo SURF, puesto que es un método relativamente rápido que tiene la ventaja de que los puntos que proporciona tienen buenas características para ser seguidos, especialmente puesto que los puntos que proporciona son invariantes frente a rotaciones, cambios de escala y cambios en la iluminación de las imágenes. Esto hace que sea muy recomendable, ya que se ha comprobado que el movimiento de la cámara y el exceso de iluminación en las grabaciones, provoca cambios de iluminación en las imágenes.

Posteriormente, se realiza el seguimiento de dichos puntos. Para tal efecto se ha utilizado el algoritmo de seguimiento de Lucas-Kanade (KLT), que permite realizar un seguimiento de los puntos en dos cuadros consecutivos de vídeo y que proporciona muy buenos resultados cuando el movimiento que existen entre ambos es pequeño. Los vídeos que se van a procesar cumplen este requisito, de forma de que este algoritmo es adecuado para este proyecto.

A continuación se lleva a cabo una de las partes más relevantes de este proyecto, puesto que es esta fase la que diferencia a las herramientas que procesan vídeos con cámara fija de aquellas que procesan vídeos con cámara móvil. La técnica utilizada consiste en comparar dos flujos ópticos para determinar los píxeles móviles. Uno de estos flujos es el flujo óptico real, que describe el movimiento que se ha producido en cada uno de los píxeles seleccionados, y el otro flujo es el flujo óptico artificial, que representa el movimiento global que se produce en la imagen y tiene como origen el movimiento de la cámara. Mediante la comparación de ambos flujos se consigue un alineamiento de la imagen y una segmentación del movimiento, que permite detectar los píxeles en los que hay movimiento producido por los objetos móviles.

La siguiente fase ha consistido en realizar un agrupamiento de los píxeles dinámicos de la imagen. Este agrupamiento ha conseguido dos objetivos: En primer lugar se consiguen eliminar píxeles que han sido definidos como dinámicos pero que no pertenecen a ningún objeto y son producto de errores en la fase de segmentación, y por otro lado se consigue también reducir considerablemente la carga computacional puesto que se ha decidido guardar de cada objeto únicamente información sobre la posición de su centro, su tamaño y los valores de intensidad R, G y B de la imagen.

Y por último, la fase del seguimiento es otra de las fases más cruciales del proyecto, puesto que permite eliminar objetos que han sido detectados erróneamente en las fases anteriores mediante un filtrado temporal y espacial y además permite conocer la trayectoria que siguen los objetos móviles.

A continuación se ha realizado un análisis experimental del funcionamiento de la herramienta. Las pruebas realizadas se han llevado a cabo a partir de dos bancos de vídeos diferentes. El primero de ellos consta de una serie de vídeos grabados por una cámara de fotos y el segundo banco de vídeos han sido grabados por el UAV en la universidad de "University of the West of Scotland". El UAV está manejado de forma teledirigida y los vídeos se han tomado dentro de un polideportivo.

Mediante estos vídeos se han fijado los parámetros de funcionamiento de la herramienta de forma que se maximizara el rendimiento de la misma y posteriormente se ha realizado la comprobación de que la herramienta realizaba la detección y seguimiento de objetos en movimiento a partir de cámaras fijas o en movimiento. Para ello se ha realizado una comprobación visual de su funcionamiento. Además, se ha llevado a cabo un estudio cuantitativo que comparara el funcionamiento de la herramienta con el método de comparación de flujos ópticos frente a su funcionamiento sin el cálculo del flujo óptico artificial ni dicha comparación de flujos. Este análisis ha permitido comprobar que el funcionamiento de la herramienta es muy superior cuando se incluye el cálculo de flujo óptico artificial y la comparación de flujos ópticos.

5.2. Trabajo futuro

Las conclusiones obtenidas en este proyecto consisten fundamentalmente en el hecho de que el conjunto de métodos utilizados para la detección y seguimiento de objetos móviles desde cámaras en movimiento resuelven el problema planteado; sin embargo los resultados pueden mejorarse en varios aspectos.

Uno de los aspectos que puede tratar de mejorarse es la robustez con la que opera la herramienta, de forma que se consiga seguir a los objetos durante más tiempo sin perder su rastro. Sin embargo, una mejora más necesaria sería tratar de reducir las falsas detecciones. Esto podría conseguirse mejorando la fase de alineamiento de imágenes. Para ello podría buscarse otro método de seguimiento de puntos que permitiera emparejar puntos que hayan sufrido transformaciones más complejas como las transformaciones proyectivas, lo cual podría conseguir un flujo óptico real más preciso, que no diera lugar a la detección de píxeles dinámicos que no corresponden a objetos móviles.

También podría tratar de mejorarse la fase de agrupamiento de objetos. Para ello podrían estudiarse algoritmos alternativos al de las k-medias o realizar fases de post-agrupamiento. Estas fases podrían consistir en combinar los objetos detectados que se encuentren muy próximos entre sí y eliminar los objetos que tras esta fase de combinación, no alcancen un tamaño mínimo.

El seguimiento de los objetos móviles también podría mejorarse si se realizara una fase previa de determinación de los parámetros que caracterizan el filtro de forma estadística. Sin embargo esto podría suponer que el tiempo de inicialización se incrementara. Otro trabajo futuro que podría considerarse en este sentido sería estudiar el funcionamiento de un filtro de Kalman sin esencia para el seguimiento de los objetos.

Otras de las mejoras que se podría estudiar sería utilizar la información de los sensores inerciales del UAV para mejorar el cálculo del vector de flujo óptico artificial.

Y por último, uno de los trabajos futuros más interesantes podría consistir en tratar de aumentar la velocidad de funcionamiento de la herramienta para poder trasladarla a una ejecución en tiempo real. Si se consiguiera una ejecución en tiempo real podría realizarse el procesamiento de las imágenes a bordo del UAV y este procesamiento permitiría controlar la trayectoria del UAV por medio de órdenes enviadas a sus controladores de forma que se dotaría al UAV de visión artificial y no sería necesario controlarlo de forma teledirigida.

5.3. Presupuesto económico

Concepto	Descripción	Coste
Coste mano de obra: Grado en Ingeniería de Tecnologías de Telecomunicación	Aproximadamente 500h	15€/h * 500h = 7.500€
Ordenador portátil con Windows 7	Coste total: 600€ 12,5€/mes	15,63€
Microsoft Office Hogar y Estudiantes 2013	Coste total: 119€ Amortización: 1 año	12,40€
MATLAB Student R2013b	Coste total: 35€ Amortización: 6 meses	7,29€
Toolbox "computer vision" MATLAB	Coste total: 20€ (sólo se ha utilizado en el presente proyecto)	7€
Material de oficina	Folios y bolígrafos	8€
TOTAL:		7550,32€

Referencias

1. Héctor López Paredes. Detección y seguimiento de objetos con cámaras en movimiento. Proyecto fin de carrera, Universidad Autónoma de Madrid, Septiembre 2011.
2. M. Valera and SA Velastin. Intelligent distributed surveillance system: a review. *Vision, Image and Signal Processing*, 152(2):192–204, 2005.
3. Francisco Javier García Mata. Videovigilancia: CCTV usando vídeos IP. Editorial Vértice, 21 jun. 2010.
4. José Portillo Portillo. Detección de movimiento de objetos mediante secuencias de vídeo. Tesis doctoral, Instituto Politécnico Nacional de México, Mayo 2012.
5. Marcos Arturo Rosales García. Análisis forense en imágenes digitales. Tesis doctoral, Universidad Calhuacan, Méjico D.F, Junio 2008
6. B. Aldalur y M. Santamaría. Realce de imágenes: filtrado espacial. *Revista de teledetección*, 2002.
7. CP Ramos, DR Rodríguez. Reconocimiento automático de matrículas. Universidad Carlos III de Madrid, 2006.
8. A Samal, PA Iyengar. Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern recognition*, 1992.
9. Harris, C.G., Mohr, R., Bauckhage, C. A combined corner and edge detector. *Alvey Vision Conference*, 1998.
10. Mónica Ballesta, Arturo Gil, Óscar Reinoso, Luis Payá, Óscar Martínez Mozos. Evaluación de detectores de puntos de interés para SLAM Visual. XXVII Jornadas de automática, Universidad de Huelva.
11. Lowe, D.G. Object recognition from local scale-invariant features. *Int. Conf. on Computer Vision*, 1999.
12. Herbert Bay, Tinne Tuytelaars, Luc Van Gool. SURF: Speeded Up Robust Features, 2006.
13. Antonio Redondo López. Extracción de características de imagen para navegación de robots móviles. Proyecto fin de carrera, Universidad de Extremadura, Febrero 2011.
14. S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, 1997.
15. Edward Rosten, Tom Drummond. Machine learning for high-speed corner detection, 2006.
16. D. Pascual, F. Pla, S. Sánchez. Algoritmos de agrupamiento. *Método Informáticos Avanzados*, 2007.
17. Francisco Hernán Ledesma. Seguimiento robusto de contornos para el monitoreo de tráfico vehicular. Tesis de grado, Universidad de Buenos Aires, Julio 2012.
18. Rafael Aracil López. Desarrollo de un sistema cognitivo de visión para la navegación robótica. Proyecto fin de carrera, Universidad Politécnica de Valencia, Julio 2012.
19. Pablo Pérez García. Composición de imágenes digitales. Proyecto fin de carrera, Universidad Politécnica de Madrid, Noviembre 2004.
20. Lucas, Bruce D., Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision, *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, April, 1981, pp. 674–679.

21. Carlo Tomasi, Takeo Kanade. Detection and Tracking of Point Features, Computer Science Department, Carnegie Mellon University, April, 1991.
22. P. Torr, A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15(591–605), año 1997.
23. M. A. Fischler, R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, año 1981.
24. Eusebio de la Fuente-López, Félix Miguel Trespaderne. Guiado de robots industriales con visión artificial. Robotización de un proceso de fabricación de radiadores. *Dyna*, Vol. 84 nº7, 565/576. Octubre 2009.
25. Raúl Montoliu Colás. Generalized least squares-based parametric motion estimation and segmentation. Tesis doctoral, Universidad Jaume I, Septiembre 2008.
26. Mónica Ballesta, Oscar Reinoso, Arturo Gil, Miguel Juliá, Luis Payá. Métodos para el alineamiento de mapas de características visuales. XXIX Jornadas de Automática (Tarragona, 3-5 Septiembre 2008).
27. Gonzalo R. Rodríguez-Canosa, Stephen Thomas, Jaime del Cerro, Antonio Barrientos, Bruce MacDonald. A Real-time method to detect and track moving objects from unmanned aerial vehicles using a single camera. *Remote sensing*, Abril 2012.
28. J. MacQueen. Some methods for classification and analysis of multivariate observations. Universidad de California, 1967.
29. Hans-Hermann Bock: Clustering methods: a history of k-means algorithms. Springer Verlag, Heidelberg, 2007, 161-172.
30. Carlos J. Alonso González. Métodos no supervisados: Agrupamiento. Apuntes de la asignatura Inteligencia Artificial I. Universidad de Valladolid.
31. Cristina García Cambroner, Irene Gómez Moreno. Algoritmos de aprendizaje: knn y kmeans. Apuntes de la asignatura Inteligencia en redes de telecomunicación. Universidad Carlos III de Madrid.
32. Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME—Journal of Basic Engineering*, pp. 35-45, Marzo 1960.
33. Greg Welch, Gary Bishop. An introduction to the Kalman Filter. Department of Computer Science. Universidad de Carolina del Norte, Julio 2006.
34. Pedro José Gómez del Hoyo. Técnicas de seguimiento de blancos de radar. Desarrollo de soluciones para radares pasivos en aplicaciones terrestres. Universidad de Alcalá. Septiembre de 2014.
35. Faisal Bahir, Fatih Porikli. Performance Evaluation of Object Detection and Tracking Systems. Mitsubishi Electric Research Laboratories. <http://www.merl.com>