



Universidad Autónoma de Madrid
Escuela Politécnica Superior



SEGUIMIENTO Y PRODUCCIÓN AUTOMÁTICA MEDIANTE CÁMARAS PTZ EN ENTORNOS DE RED

Trabajo de Fin de Máster

por

Antonio González Huete

Tutor: Dr. Jesús Bescos Cano

Septiembre, 2013

Abstract

El seguimiento automático de objetos en secuencias de vídeo ha sido desde siempre un campo de gran interés dentro de la comunidad científica. No obstante y dentro de este ámbito, el campo del seguimiento de objetos mediante cámaras móviles es probablemente uno de los menos evolucionados, lo que no quiere decir que sea de los de menor interés.

En este trabajo se presentará un framework completo de seguimiento y producción automática mediante cámaras móviles en entornos de red, el cual será aplicado además a la problemática del seguimiento de profesores y producción automática, para la emisión de clases presenciales virtuales.

Como consecuencia de ello, se presentará también el conjunto de algoritmos de seguimiento desarrollados a lo largo de toda la investigación, donde se expondrá un nuevo método de detección y aprendizaje de objetos para su seguimiento en secuencias de vídeo de larga duración. Y donde dicho método no sólo demostrará ser tremendamente efectivo en dicha tarea, sino que también demostrará una aparente mejora con respecto al estado del arte actual en lo referente al ámbito general del seguimiento de objetos.

Por otro lado y para que los métodos de seguimiento implementados, puedan ser aplicados dentro del campo del seguimiento automático mediante cámaras móviles a través del framework desarrollado, se detallará también el sistema de estabilización de vídeo inter-frame implementado para tal efecto.

Índice de contenido

1	Introducción.....	4
1.1	Motivación.....	4
1.2	Objetivos.....	4
1.3	Organización de la memoria.....	6
2	Estado del arte.....	7
3	Arquitectura.....	9
3.1	Infraestructura.....	9
3.2	Diseño de la arquitectura.....	10
4	Sistema de adquisición.....	12
4.1.1	Diseño del sistema.....	13
4.1.2	Implementación del protocolo MJPEG.....	15
5	Sistema de control de la cámara PTZ.....	18
5.1	Control absoluto.....	19
5.2	Control relativo.....	19
6	Sistema de estabilización de vídeo.....	21
6.1	Diseño del estabilizador.....	22
6.2	Detección de características.....	23
6.3	Las KLT.....	24
6.4	Corrección de perspectiva.....	25
6.5	Implementación y resultados.....	26
7	Sistema de seguimiento.....	29
7.1	Seguimiento mediante KLT's.....	30
7.1.1	Diseño.....	30
7.1.2	Cálculo estadístico y filtrado de outliers.....	32
7.1.3	Impresiones.....	34
7.2	Seguimiento mediante Filtros de Partículas de movimiento.....	36
7.2.1	Filtros de partículas.....	36
7.2.2	Diseño.....	37
7.2.3	Impresiones.....	39
7.3	Seguimiento mediante Partículas KLT (PKLT).....	41
7.3.1	La importancia de aprender y combinar características.....	41
7.3.2	Diseño.....	42
7.3.3	Impresiones.....	44
7.4	Seguimiento mediante PKLT y Mean-Shift (PKLTGA).....	45
7.4.1	Mean-Shift.....	45
7.4.2	Diseño.....	47
7.4.3	Impresiones.....	48
8	Sistema de producción automática.....	50
8.1	Lógica de control en cámara móvil.....	50
8.2	Lógica de control en la cámara fija (triangulación).....	51
8.3	Reglas de producción.....	52
9	Evaluación y Resultados.....	54
10	Conclusiones y Trabajo Futuro.....	59
	Referencias.....	60

1 Introducción

1.1 Motivación

El seguimiento automático de objetivos móviles ha sido desde hace décadas un área de gran interés con un enorme campo de aplicación, donde a lo largo de la historia se le han dedicado infinidad de recursos de investigación. Recursos los cuales, al contrario que en otras áreas de la visión artificial, han permitido una sustancial evolución de sus distintas vertientes. De todas estas vertientes, la menos evolucionada con diferencia es la referente al seguimiento automático con cámaras móviles, lo cual es causa directa de dos factores principales: el gran incremento de complejidad en el sistema que suponen, y la poca utilidad de las técnicas desarrolladas en otras vertientes para este tipo de escenarios.

El seguimiento automático mediante cámaras móviles a pesar de ser el menos evolucionado, sigue siendo de gran importancia e interés para la comunidad científica, contando también con una enorme variedad de campos de aplicación: vídeo-vigilancia, robótica, realidad aumentada, conducción autónoma de vehículos no tripulados, sistemas de filmación y producción automática, etc. No obstante, cada uno de estos campos posee su propio entorno de aplicación con sus respectivos detalles que lo diferencian del resto, y que actualmente a falta de soluciones más genéricas, suelen desembocar en soluciones más específicas para cada uno de ellos, sin llegar a cumplir expectativas la mayoría de las veces. Este campo fue siempre explotado en ámbitos profesionales de carácter bastante específico (como el sector militar, robótica, vídeo-seguridad, etc), pero actualmente y gracias a la realidad aumentada, es un campo en auge. No obstante, aún existen multitud de problemas y barreras tecnológicas no superadas que evitan su despegue y explotación definitivas.

A raíz todo esto, en este trabajo se pretende abordar esta última vertiente aplicándola al ámbito de la filmación y producción automática para la emisión de clases presenciales. Trabajo el cuál, pretende servir como continuación a un proyecto previo de esta misma Escuela [9], donde los resultados del seguimiento automático no llegaron a cumplir con las expectativas deseadas. A causa de ello y de la arquitectura ya existente, se hace necesario trabajar con cámaras IP PTZ, con el incremento en la problemática que ello implica: baja e inestable tasa de cuadros por segundo, retardos en el envío, recepción y ejecución de los comandos de movimiento a la cámara, etc.

1.2 Objetivos

Según los resultados del proyecto anterior [9] y la particular arquitectura ya existente, se hace necesario un replanteamiento de toda la parte de análisis de vídeo en general. Lo cual combinado a la presente situación del estado del arte referente al ámbito del seguimiento automático mediante cámaras móviles [1-16], hace necesaria la exploración de nuevas vías de investigación que nos permitan resolver la problemática concreta actual.

Se hace por lo tanto necesario un sistema de seguimiento que permita lidiar de forma robusta con:

- La latencia de la red.
- La demora intrínseca no solo al propio protocolo de adquisición, sino también a las correspondientes implementaciones del mismo en los dos extremos.

- La demora de la propia cámara a la hora de ejecutar los comandos de movimiento.
- La inestable tasa efectiva de “frame rate”. Causada principalmente por la enorme cantidad de imágenes no válidas producidas mientras la cámara se está moviendo, las cuales entre otras cosas habrá que detectar e ignorar.

Estos retardos desgraciadamente pueden limitar enormemente el alcance del propio aplicativo, ya que al menos a nivel de seguimiento exclusivo mediante cámara móvil, el seguimiento siempre quedará limitado a la propia velocidad de la cámara a la hora de recibir, procesar y ejecutar los propios comandos de movimiento.

Teniendo esto en cuenta toda esta problemática, para la realización del presente proyecto será necesario abordar los siguientes objetivos:

- Estudio del estado del arte actual: Teniendo en cuenta toda la problemática expuesta anteriormente, se realizará un estudio en profundidad del estado del arte actual, dirigiendo dicho estudio a la resolución de la problemática concreta actual.
- Análisis y diseño de la arquitectura del sistema: Con el conocimiento adquirido mediante el estudio del estado del arte, será necesario el diseño detallado de la arquitectura del sistema para la solución a implementar.
- Desarrollo de los sistemas de captura y control de la cámara: Independientemente de la solución a implementar y de su arquitectura, será necesario el desarrollo de los módulos de adquisición de vídeo y control PTZ respecto a las cámaras a utilizar. Para ello habrá que evaluar las posibles opciones (implementaciones propietarias, libres o desarrollo propio) y optar por la más adecuada.
- Desarrollo del/los sistemas de seguimiento: Independientemente de la arquitectura definida, el objetivo principal del presente proyecto sigue siendo el desarrollo de un sistema de seguimiento que cubra las necesidades propuestas por el proyecto heredado original, es decir, el seguimiento a largo plazo de un profesor dentro de un aula durante una clase normal.
- Desarrollo del sistema de producción automática: Por otro lado, y como objetivo secundario, será también necesaria la implementación en la medida de lo posible, de un módulo para la producción automática del vídeo, permitiendo así una posible y posterior emisión de las clases en forma virtual.
- Pruebas, evaluación y resultados del sistema desarrollado: Independientemente del sistema desarrollado, será necesaria una medida objetiva de la calidad del sistema implementado mediante un pequeño set de pruebas que permitan la evaluación y extracción de los resultados del mismo, siendo preferiblemente hacerlo de forma que éstos puedan ser contrastados y comparados con otras soluciones o trabajos.
- Conclusiones finales y evaluación del posible trabajo futuro. Finalmente y en base a los resultados obtenidos, será necesario un análisis en profundidad de los mismos, de tal forma que se extraigan tanto las conclusiones finales del proyecto, como su posible trabajo a futuro.

1.3 Organización de la memoria

La organización de la memoria esta realizada en base a las distintas etapas definidas para el presente proyecto, cuya naturaleza en forma secuencial se distribuye como sigue:

- En el segundo capítulo se hace un estudio del estado del arte actual, estudiando, analizando y evaluando las distintas soluciones proporcionadas por el mismo dentro del seguimiento automático de objetos mediante cámaras móviles.
- En el tercer apartado se presenta el análisis y diseño de la arquitectura implementada para el presente trabajo.
- Del cuarto al noveno apartados se presentan y detallan los principales módulos implementados en el desarrollo del sistema actual.
 - En el cuarto apartado se presenta el sistema de adquisición de vídeo implementado en el lado del cliente.
 - En el quinto apartado se presentan los dos principales sistemas de control implementados para el control de la cámara PTZ.
 - En el sexto apartado se detalla el sistema de estabilización de vídeo inter-frame implementado para el sistema de seguimiento mediante cámara móvil.
 - En el séptimo apartado se exponen los principales sistemas de seguimiento implementados para las tareas de seguimiento automático, enumerados por orden de implementación, y representando con ello la evolución seguida en la investigación realizada con respecto al ámbito del seguimiento automático.
 - En el octavo apartado se detallan los métodos y reglas utilizadas, para la la implementación de las distintas lógicas de control y producción automática.
- En el noveno punto se presentan y evalúan los resultados obtenidos en las pruebas del sistema.
- Finalmente en el décimo punto se exponen las conclusiones obtenidas respecto al campo de investigación abordado, proponiéndose también las posibles líneas de trabajo futuro a seguir.

2 Estado del arte

El campo del seguimiento automático mediante cámaras móviles presenta una solución funcional y atractiva para un conjunto de casos bastante amplio dentro del ámbito del seguimiento automático, pero desgraciadamente y a día de hoy, no existe mucha variedad de soluciones e iniciativas dentro de este campo. Normalmente las causas de este vacío suelen venir dadas por dos motivos principales: por representar dicha vía de investigación a un problema ya resuelto, o porque actualmente dicha vía se encuentre en un callejón sin salida respecto a la tecnología existente. Pero en este caso, no parece cumplirse ninguna de ambas.

En este campo, Varcheie et al. en su conjunto de trabajos sobre seguimiento exclusivo mediante cámaras móviles [1-3], presentan distintas aproximaciones al problema desde varios puntos de vista: primero un sistema de seguimiento mediante KLT en [1], posteriormente un segundo sistema de seguimiento mediante detección de movimiento y “blobs” con clasificadores difusos en [2], y finalmente un sistema de seguimiento mediante filtros de partículas adaptativos de color en [3]. En [10], Xie et al. proponen un sistema de seguimiento también basado en filtros de partículas, pero esta vez aplicados sobre el contorno del “omega shape” en vez de sobre el color, donde el objeto es detectado inicialmente mediante el método de Viola-Jones. Siguiendo esta línea, en [12] Chang et al. proponen un sistema de seguimiento similar, pero haciendo uso de Mean-Shift sobre el espacio de color HSV para el seguimiento automático. En [15], Xie et al. exponen otro sistema de seguimiento basado en la forma, pero esta vez desde una perspectiva multi-parte donde el objeto es dividido en trozos, realizando el seguimiento de cada uno de ellos mediante su descriptor HOG, y donde el zoom de la cámara es detectado mediante el ratio correspondiente presentado por la homografía que pone en correspondencia cada par de frames consecutivos.

Cucchiara et al. en [4], presentan una aproximación al problema desde otra vía más clásica, donde tras un registro de imagen mediante la compensación de movimiento de la cámara, realizan la clásica detección y seguimiento de objetos mediante segmentación de movimiento. Afín también a esta vía, Guillot et al. en [5] utilizan un sistema de registro de fondo mediante posiciones fijas de la cámara, donde la segmentación de movimiento es realizada mediante la disimilitud de la densidad de los puntos característicos registrados en el fondo. En [6], Robinault et al. realizan una aproximación similar mediante una estabilización y registro de imagen para posteriormente realizar la segmentación de movimiento mediante la extracción del fondo.

Como aproximación menos convencional pero más práctica, Yi Da Xu et al. presentan en [8] un sistema híbrido que utiliza una cámara fija de apoyo con alto campo de visión para la realización de las tareas de seguimiento, mientras que con la cámara móvil triangulan respecto a la cámara fija y realizan tareas de reconocimiento facial. De forma similar, Chen et al. en [7] presentan un sistema que extiende el concepto a múltiples cámaras de apoyo PTZ que colaboran entre sí para la realización del seguimiento.

En [11], Wu et al. realizan un estudio sobre el impacto que producen los errores mecánicos y aleatorios de las cámaras móviles, sobre su sistema de calibración durante tiempos de ejecución prolongados, proponiendo a su vez un sistema de registro de características de la escena que no sólo

permita la corrección de dichos errores, sino que también sirva de ayuda al proceso de seguimiento automático.

Por otro lado y en lo que respecta al seguimiento automático orientado a secuencias de larga duración, el vacío es aún mucho más grande, no habiéndose encontrado prácticamente referencias al respecto en la literatura actual. Sin embargo, el campo de seguimiento automático en secuencias de video para cámara fija y generalmente de corta duración, ha estado tremendamente activo en las últimas dos décadas, produciendo este último prácticamente todos los avances significativos dentro del ámbito general del seguimiento automático. De hecho dichas técnicas son precisamente las mismas técnicas que posteriormente fueron utilizadas en todos los métodos de seguimiento mencionados anteriormente: KLT [19-26], filtros de partículas [28-36], y Mean-Shift [37-44].

3 Arquitectura

3.1 Infraestructura

Antes de abordar el diseño de la arquitectura del sistema, es necesario analizar con detalle la infraestructura existente sobre la que ha de montarse el sistema.

Inicialmente se poseen dos cámaras SONY instaladas en el aula:

- Una cámara móvil de tipo PTZ, modelo SONY Ipela SNC-RZ50P.
- Una cámara fija con gran angular, modelo SONY Ipela SNC-CH260.

Al ser ambas cámaras del mismo fabricante, ambas son cámaras IP que utilizan los mismos protocolos de adquisición, control y configuración.

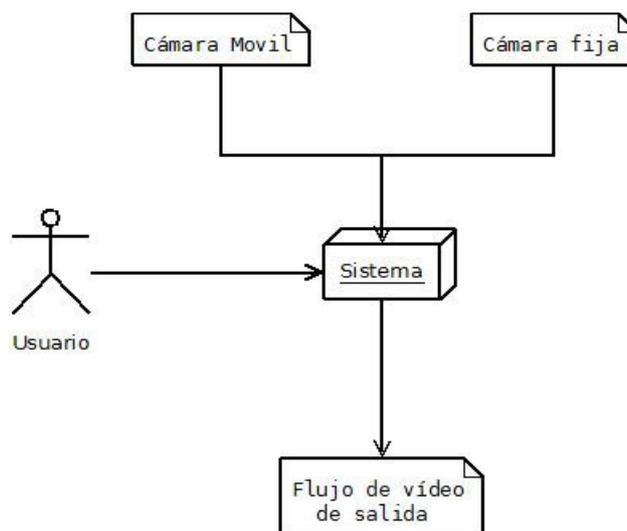


Figura 3.1: Infraestructura

Como se puede observar en la Figura 3.1, el sistema podrá recibir ambos flujos de vídeo, pudiendo éste tan sólo interactuar con la cámara móvil, donde existirá una mínima interacción del usuario para arrancar el sistema y seleccionar el objeto a seguir.

En cuanto a la salida del sistema, esta será tan sólo un flujo de vídeo, el cual podrá ser utilizado para la emisión virtual de la clase según el proyecto original [9], siendo dicho flujo de vídeo el correspondiente a la cámara móvil. De esta forma e independientemente de los flujos de entrada o la tecnología utilizada para el seguimiento, el sistema será el encargado de seguir al objeto a lo largo del aula mediante la cámara móvil de la forma más suave posible, pudiendo hacer uso también del zoom para las labores de producción automática.

3.2 Diseño de la arquitectura

El estudio realizado sobre estado del arte en el apartado 2 revela la poca disponibilidad de soluciones robustas a la problemática específica a abordar, al menos al nivel de los resultados deseados. Las soluciones ofrecidas por el estado del arte actual son principalmente de tres tipos:

- Seguimiento mediante cámara móvil haciendo uso de características [1-3].
- Seguimiento mediante cámara móvil utilizando métodos de segmentación de movimiento [4-6].
- Seguimiento mediante cámara móvil con apoyo de una cámara fija [7-8].

Cada una de estas vías presenta unas ventajas e inconvenientes distintos, pero ninguna de ellas por si misma provee de una solución completa al problema a resolver. Por ello, para este trabajo se propone una arquitectura híbrida, que no sólo pretende mezclar los mejores puntos de cada una de ellas, sino que también propone avances significativos respecto a la vía más representativa y de mayor interés: la vía correspondiente al seguimiento automático mediante cámara móvil por características.

Por otro lado, y teniendo en mente la infraestructura existente (Figura 3.1), la arquitectura diseñada contemplará tanto el seguimiento exclusivo mediante cámara móvil, como el seguimiento exclusivo mediante cámara fija, como el seguimiento combinado mediante ambas cámaras (aunque desgraciadamente este último no pudo ser abordado por falta de tiempo).

Por todo ello, y con el fin de proporcionar flexibilidad a la hora tanto de desarrollar el sistema de seguimiento como de evaluar su rendimiento, se ha definido la arquitectura modular presentada en la Figura 3.2. Como se puede ver en dicha figura, se definen cuatro bloques principales bien diferenciados: Adquisición, análisis, producción y control.

El bloque de adquisición será el encargado de adquirir y normalizar los flujos de vídeo de las distintas cámaras utilizadas. Normalmente este módulo suele encargarse tan solo de la gestión y direccionamiento de vídeo, delegando la parte de adquisición a una librería externa (como OpenCV o la propia librería de Sony, etc), pero desgraciadamente y por un error en la autenticación de la propia librería, ha sido necesaria también la implementación de los protocolos correspondientes.

El bloque de control de la cámara PTZ será el encargado de implementar los protocolos correspondientes para el envío de los comandos a la cámara. A causa del enorme impacto en el rendimiento de los algoritmos de seguimiento que puede presentar el tiempo de envío y ejecución de dichos comandos, se hace obligatorio el intentar minimizar la latencia del proceso lo máximo posible.

En cuanto al bloque de producción, éste tendrá una doble función. Por una parte la propia lógica de producción automática, la cual puede suponer por si sola todo un campo de estudio e investigación. Y por otro lado, los correspondientes sistemas de control que gestionen y traduzcan las decisiones de producción en movimientos de cámara. Dichos sistemas de control habrán de contemplar tanto el uso exclusivo de la cámara móvil, como su uso combinado con la cámara de entorno de apoyo (la

cámara fija), lo cual hará necesario entre otras cosas, de una calibración y puesta en correspondencia entre ambas cámaras.

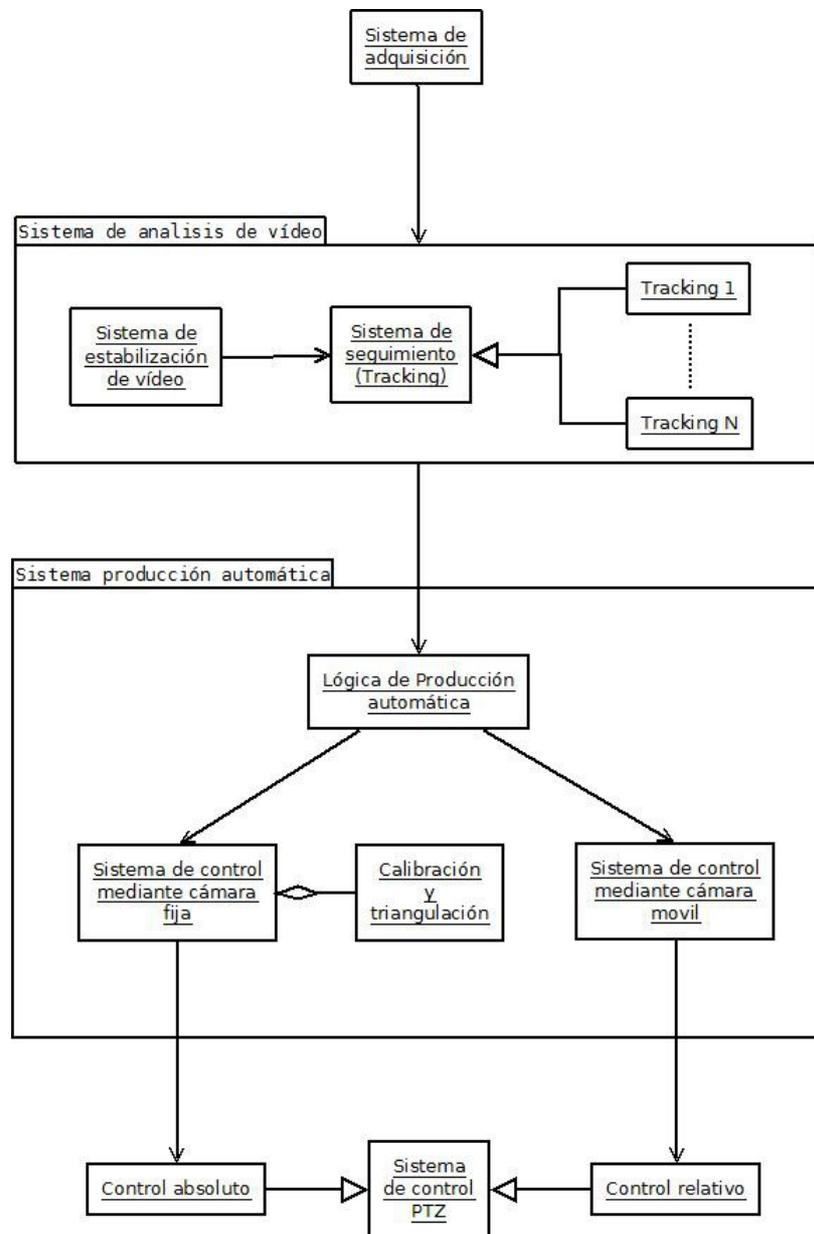


Figura 3.2: Arquitectura del sistema

Por último y como objetivo principal de este trabajo, se encuentra el bloque de análisis de vídeo, el cual tendrá como objetivo la selección y seguimiento del objeto deseado a largo plazo en el entorno definido. Aquí se irán desarrollando los múltiples algoritmos de seguimiento según vaya avanzando la investigación, generándose con ello múltiples módulos de seguimiento que podrán ser fácilmente intercambiados mediante la configuración del módulo principal. Por otro lado y como la adquisición puede venir de una cámara en movimiento, se hace necesario también la implementación de un sistema de estabilización de vídeo inter-frame que dote a los algoritmos de seguimiento de independencia respecto al origen del flujo de vídeo (cámara fija o móvil).

4 Sistema de adquisición

Al estar limitados al uso de las cámaras ya instaladas previamente, el sistema de adquisición queda bastante restringido, en particular a los protocolos implementados por Sony para sus cámaras. Por lo tanto y antes de abordar el problema, se evaluaron distintas opciones:

- Utilizar las librerías de terceros proveídas por Sony.
- Utilizar el sistema de adquisición proporcionado por la comunidad de “OpenCV” u otra librería especializada libre.
- Implementar alguno de los protocolos soportados y especificados en la propia documentación de Sony.

La tercera opción de implementar el protocolo correspondiente, obviamente en principio estaba fuera del objetivo de este proyecto, y por ello se descartó de inmediato.

La primera opción fue la utilizada en el proyecto previo, y en nuestro caso fue descartada principalmente por la complejidad que introducía el tener que mantener y depender de esta librería de terceros en concreto. Además, el uso de dicha librería de por sí sólo presentaba otros problemas y desventajas adicionales respecto al esfuerzo necesario para su integración, limitando también por otro lado todo el sistema a esta marca de cámaras en concreto. No obstante, esta solución proporcionaba la ventaja de tener implementados protocolos bastante complejos y optimizados, como por ejemplo el H264, que en caso de requerir trabajar con múltiples flujos de video bajo la misma red a una alta tasa de cuadros por segundo, sería algo prácticamente obligatorio. Pero que en este caso concreto al estar limitados a un máximo de dos flujos de vídeo, dicho problema no se encontraba presente, siendo por lo tanto preferible maximizar la compatibilidad.

Por otro lado la opción de utilizar una librería libre, y más concretamente OpenCV, sólo presentaba ventajas: posibilidad de integrar cualquier otro tipo de cámara posteriormente, contar también con la posibilidad de utilizar otros protocolos más complejos y óptimos como H264, etc. No obstante y desgraciadamente, tras varias pruebas se detectó un importante error en la versión utilizada de OpenCV, el cual impedía el poder realizar la autenticación con la cámara de una forma correcta a través de red.

A causa de esto y tras revisar la documentación del SDK proporcionado por Sony [17], se decidió implementar el protocolo MJPEG proporcionado tanto por estas cámaras, como por la mayoría de las existentes en el mercado, independizando de esta forma lo máximo posible el sistema de la cámara a utilizar.

4.1.1 Diseño del sistema

Con el objetivo de no limitar exclusivamente el proyecto a las cámaras de Sony, se diseñó una interfaz abstracta para aumentar la modularidad del módulo de adquisición. De esta forma, mientras se respete la interfaz definida, se podrían añadir nuevos módulos de soporte de video, protocolos, etc...

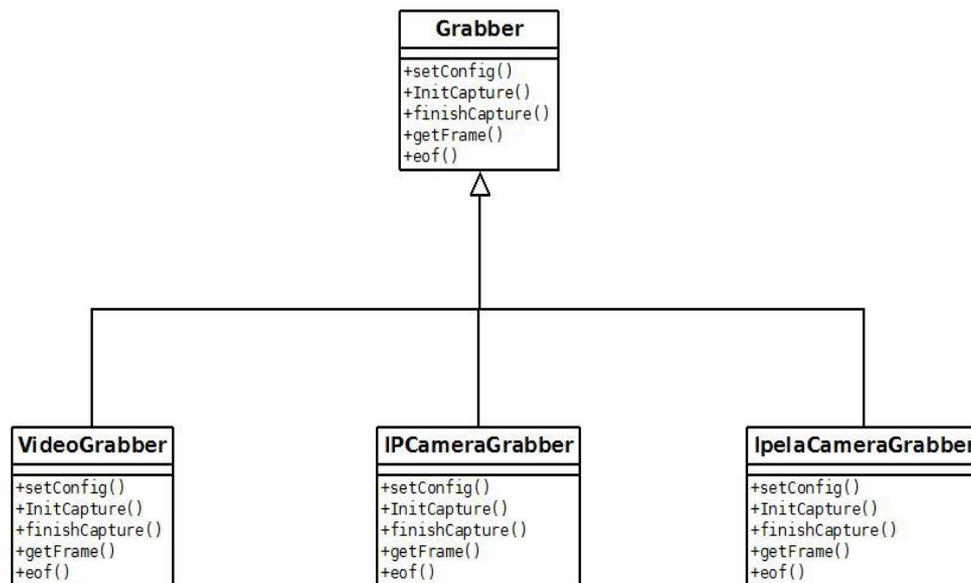


Figura 4.1: Diseño del sistema de adquisición

El diseño final realizado para el sistema de adquisición se puede ver en la figura 4.1, donde se puede observar cómo se implementaron distintos módulos de adquisición según las necesidades del momento.

El módulo de adquisición “VideoGrabber” fue realizado con el objetivo de implementar una cámara virtual, con la que poder evaluar así el rendimiento y eficacia de los distintos algoritmos de seguimiento implementados. De esta forma, se puede así evaluar el comportamiento de cada algoritmo de seguimiento contra el mismo conjunto de videos, comparando así las ventajas e inconvenientes de cada algoritmo dentro de una misma escena y situación.

El módulo de adquisición IP “IPCameraGrabber”, fue implementado con el objetivo de integrar el módulo de adquisición de OpenCV en el proyecto, permitiendo así la adquisición de la mayoría de cámaras del mercado. Desgraciadamente y por un error interno de la versión utilizada de OpenCV, la autenticación no es enviada como debería a la cámara. Este hecho fue detectado y reportado a la comunidad mediante el análisis de las tramas IP enviadas a la cámara, haciendo uso del reconocido analizador de tramas “Wireshark”. No obstante, el modulo es funcional y permite la adquisición a la mayoría de cámaras IP del mercado siempre y cuando no se haga uso de la autenticación.

Finalmente, el módulo de adquisición utilizado para las cámaras ya instaladas de Sony, fue el módulo “IpelaCameraGrabber”, el cual implementa fielmente el protocolo MJPEG especificado por Sony en su documentación [17]. No obstante, la correcta implementación de un protocolo de este estilo requiere un sustancial esfuerzo de desarrollo: gestión de la conexión, reintentos de conexión, gestión de los buffers de entrada, etc. Obviamente todo esto se encuentra fuera del objetivo del proyecto actual, por lo que se hizo una implementación básica del protocolo a la espera de la solución de la comunidad del error detectado en OpenCV.

Cabe destacar también, un sustancial retardo en la adquisición del vídeo, de entre uno y dos segundos. En la solución híbrida, esto no supone un gran problema, pero como es obvio, en la solución exclusiva mediante cámara móvil, esto representa un inconveniente difícilmente superable. No obstante, y aunque este tema se dejó aparcado por salirse fuera del ámbito del proyecto, muy probablemente sea algo superable tanto mediante el uso de la librería de OpenCV tras la corrección del error detectado, como con una mejor gestión tanto de la cámara como de los buffers de sistema utilizados en la implementación del protocolo MJPEG.

4.1.2 Implementación del protocolo MJPEG

El protocolo MJPEG aunque presenta numerosos inconvenientes, siendo el más importante la gran cantidad de ancho de banda que requiere con respecto a otras alternativas, se puede considerar como uno de los protocolos más utilizados actualmente. Las razones principales de ello son: la relativa simplicidad de su implementación, y la independencia temporal de cada una de las imágenes. No obstante y teniendo en cuenta el volumen de ancho de banda demandado por los flujos necesarios para la aplicación, y la modesta tasa de cuadros por segundo requerida, el protocolo MJPEG presenta una solución más que suficiente y sin inconvenientes para este caso.

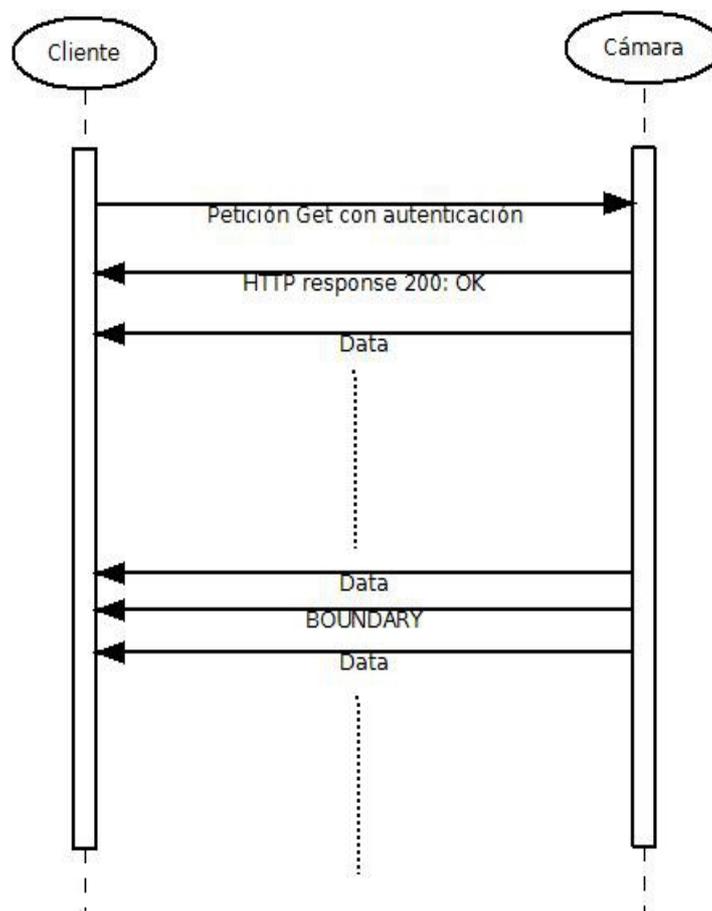


Figura 4.2: Paso de mensajes del protocolo MJPEG

En este caso, para la implementación del protocolo se hizo uso de la documentación proporcionada por Sony [17] (que detallaba su protocolo), y del estándar de autenticación web “basic access authentication” definido por el estándar HTTP 1.0. La descripción del paso de mensajes del protocolo implementado se puede ver en la Figura 4.2, donde se puede observar la simplicidad del proceso:

- Se envía la petición “GET” con la autenticación según el estándar “basic access authentication” del estándar HTTP 1.0.

- Se espera a recibir la respuesta de aceptación o denegación del servicio.
- Y finalmente se empiezan a transmitir los datos separados por un delimitador de frame definido en la respuesta de validación de la cámara.
 - Al principio de cada frame se envía información útil como la longitud de los datos, etc.
 - Luego se envía un salto de línea con retorno de carro.
 - Finalmente se envían los datos hasta que se envía la marca de delimitación de frame.

En la figura 4.3 se puede ver un ejemplo de respuesta de la cámara ante una petición “GET” válida.

```
HTTP/1.0 200 OK\r\n
Content-Type: multipart/x-mixed-replace;boundary=--myboundary\r\n
\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
CamTim: 2004-05-18 Tue 10:13:05\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
CamTim: 2004-05-18 Tue 10:13:05\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
Content-Type: image/jpeg\r\n
CamTim: 2004-05-18 Tue 10:13:06\r\n
\r\n
<JPEG image data>\r\n
--myboundary\r\n
. . . . .
```

Figura 4.3: Ejemplo de respuesta NJPEG extraído de [17]

En cuanto al protocolo de autenticación web “basic access authentication”, es el método por defecto implementado por todos los navegadores existentes para el envío de credenciales. No obstante, y aunque es el método más utilizado actualmente, este protocolo también es el menos recomendado a causa de su más que sustancial falta de seguridad, ya que las credenciales tan sólo van codificadas (y no cifradas) en Base64. Se puede ver un ejemplo completo de petición MJPEG con su correspondiente autenticación en la Figura 4.4.

```
Hypertext Transfer Protocol
GET /mjpeg HTTP/1.1\r\n
Host: 150.244.56.99\r\n
Connection: keep-alive\r\n
Authorization: Basic VHJhY29pbm91cy90cm90GU1\r\n
\r\n
```

Figura 4.4: Ejemplo de petición GET con autenticación

5 Sistema de control de la cámara PTZ

El sistema de control de las cámaras móviles es hasta el momento un campo bastante poco estandarizado desde el punto de vista que a nosotros nos interesa. Ya que aunque sí que es cierto que existen varios estándares de control (los protocolos PELCO-X, Visca, etc), la mayoría de ellos no son demasiado aptos desde el punto de vista de un análisis de vídeo.

Por otro lado y en el caso concreto con el que nos encontramos: la cámara "Sony Ipela SNC-RZ50P", ésta muestra en su manual [17] dos tipos de control recomendados (a parte de otros protocolos específicos de poca utilidad para nuestros objetivos): uno absoluto y otro relativo. Realmente ambos protocolos recomendados encajan bastante bien con las necesidades actuales de los dos sistemas que se desean combinar, por lo que se implementaron ambos para este proyecto. Se puede ver el diseño del módulo de control que contempla ambos tipos de control en la Figura 5.1.

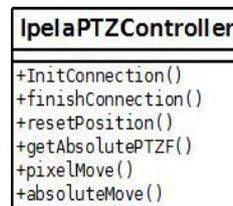


Figura 5.1: Módulo de control PTZ

A pesar de no pertenecer a ningún estándar, el método de control relativo a nivel de píxel es idóneo para el control mediante cámara móvil, lo que supone una enorme ventaja frente a otros métodos de control, al menos desde el prisma referente a tareas de análisis de vídeo.

Por otro lado el método de control absoluto, resulta ideal para el control de la cámara en un sistema híbrido como el que tenemos entre manos, ya que nos permite triangular y poner en correspondencia las posiciones en una imagen o cámara externa, con las posiciones Pan-Tilt de la cámara móvil.

En cuanto al protocolo de envío de los comandos, éste es el mismo que el utilizado para el protocolo de adquisición, siendo realizado en cualquier caso como petición GET estándar de HTML 1.0. En este caso, lo que varía según el tipo de comando es:

- Por un lado la URL de entrada.
- Por otro lado los argumentos permitidos, los cuales en cualquier caso son también enviados como parámetros web según el estándar HTML.

Por otro lado y como la apertura continua de este tipo de conexiones puede llegar a ser muy costosa (sobre todo para la cámara), pudiendo llegar incluso a saturar la propia cámara, se hizo necesaria la gestión y limitación automática de las mismas. Cabe destacar también a este respecto, que la

ejecución de los comandos por parte la cámara no es ni mucho menos inmediata, agravando con ello enormemente este último problema, ya que incluso aunque lleguen las peticiones, éstas pueden ser descartadas si la anterior no terminó de ejecutarse. Todo esto hizo, que en contra de lo que cabría esperar, se hiciese prácticamente obligatorio el uso de una política asíncrona y restrictiva para el envío de comandos a la cámara, teniendo que ignorar la respuesta de los mismos para evitar los bloqueos producidos por la propia cámara durante la ejecución de los comandos solicitados.

5.1 Control absoluto

El control absoluto proporcionado por la cámara de Sony es realizado en función del propio hardware utilizado para la fabricación del la propia cámara, teniendo que definir los valores máximos y mínimos de movimiento en función tanto del fabricante como de la cámara [Figura 5.3].

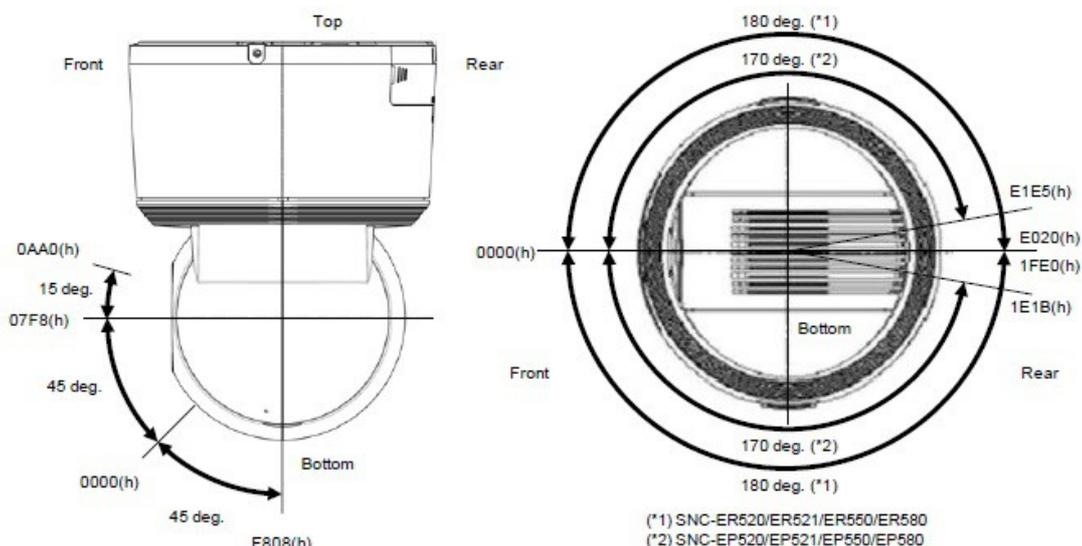


Figura 5.3: Diagrama de valores para el control absoluto de la cámara

No obstante, como el uso de este tipo de control es realizado exclusivamente para su utilización en las soluciones que usan la cámara fija de apoyo, tan sólo será necesario poner en correspondencia los valores de los motores en las distintas posiciones de calibración, con respecto sus respectivas coordenadas de imagen en la otra cámara.

5.2 Control relativo

El comando implementado por Sony para este tipo de control es el denominado “AreaZoom”, y que originalmente estaba pensado para desplazar la cámara y realizar un zoom sobre una región concreta de la imagen actual. No obstante, lo que realmente hace es desplazar el centro de la imagen a las nuevas coordenadas de imagen indicadas, utilizando el nuevo ancho y alto especificados, como

parámetros de zoom, manteniendo el zoom anterior en caso de permanecer inmutables o ser igual a cero. Esto permite a este comando ser utilizado de forma funcional como comando de desplazamiento relativo a nivel de píxel. Se puede ver un ejemplo gráfico del funcionamiento de este comando en la Figura 5.2.

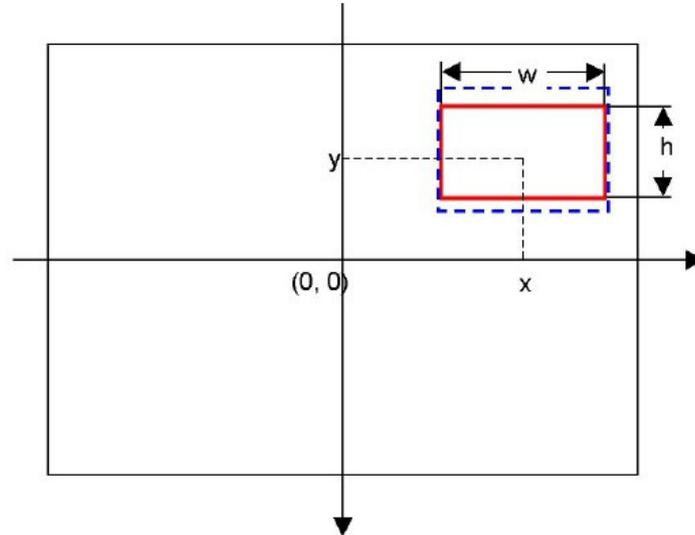


Figure 5.2: Desplazamiento relativo

Este tipo de control no solamente será útil para el sistema de control mediante cámara móvil, sino que también permitirá simplificar enormemente el módulo de calibración para el sistema de control híbrido, permitiendo así una calibración semi-automática rápida y efectiva, eliminando con ello prácticamente todos los inconvenientes del control absoluto. Esto hará que la calibración sea prácticamente transparente para el usuario.

6 Sistema de estabilización de vídeo

El estabilizador de vídeo implementado cumple dos funciones principales. La primera es la propia implementación del sistema de estabilización de vídeo para la cámara móvil, y la segunda y más importante, es la comprensión, implementación, uso y comparación de algunas de las herramientas más importantes utilizadas posteriormente en la mayoría de los sistemas de seguimiento implementados: detección de características, seguimiento de las mismas mediante las KLT, uso de la matriz homográfica para proyección de coordenadas, etc. De ahí que en este capítulo se describa con especial detalle este sistema.

Por otro lado, la estabilización de vídeo es un campo altamente recurrido en muchas aplicaciones de visión artificial, sobre todo dentro del campo de la robótica, sin olvidar también campos como la vídeo-seguridad, realidad aumentada, cámaras embarcadas, móviles, etc. En la mayoría de estos sistemas, a la hora de realizar cualquier tipo de análisis de vídeo sobre una cámara en movimiento, se tienen casi siempre dos opciones:

- Trabajar exclusivamente con objetos conocidos intentando reconocerlos dentro de la imagen en base a la información conocida de ellos, ignorando con esto cualquier información referente al resto de la escena.
- Hacer algún tipo de registro de imagen que permita poner en correspondencia espacial un frame con el siguiente, permitiendo así obtener una mucho mayor cantidad de información del vídeo. En este caso se puede tener en cuenta no sólo el conocimiento de los objetos de interés, sino también del resto de la escena.

La primera opción presenta un enorme e importante conjunto de inconvenientes: incapacidad de segmentar con robustez los objetos, incapacidad de detectar cualquier otra cosa para la que no esté calibrado el sistema, imposibilidad de realizar ningún tipo de aprendizaje a causa del enorme envenenamiento causado por el fondo u otros objetos, etc. Por todo ello y mucho más, la utilización de esta filosofía suele ser casi siempre poco recomendable, a no ser que:

- Se trabaje con objetos muy diferenciados y con una muy baja tasa tanto de falsos positivos como de falsos negativos.
- Que la puesta en correspondencia sea inviable para ese problema, situación o escena en cuestión.

La segunda opción por otro lado, no sólo provee de un mucho mayor abanico de técnicas aplicables para todos los ámbitos, sino que proporciona también una mucho mayor cantidad de información a todos los niveles. Teniendo en cuenta todo esto, junto a la intención de trabajar parcialmente con una cámara en movimiento, se decidió desarrollar un sistema de estabilización de vídeo inter-frame que permitiera a los algoritmos de seguimiento, la puesta en correspondencia espacial entre cada par de frames consecutivos. De esta forma, los algoritmos de seguimiento podrán entre otras cosas: tener en cuenta el posible fondo, poder detectar el movimiento de la cámara, poder detectar el movimiento de los objetos de la escena, poder discriminar entre ambos movimientos, minimizar el envenenamiento producido por el fondo en las tareas de aprendizaje automático, etc.

6.1 Diseño del estabilizador

A la hora de abordar el problema de la correspondencia espacial para un posterior análisis de vídeo, tenemos varias opciones: hacer un registro de imagen, una estabilización total, o una estabilización parcial. En cualquier caso, el registro de imagen es la opción más completa, y por lo tanto también la más compleja, pudiendo en este caso complicarse su problemática bastante. Sin embargo y en un principio, para nuestros algoritmos de seguimiento tan sólo es necesaria una correspondencia espacial con respecto al frame anterior, permitiendo así la eliminación del movimiento de la cámara móvil y su discriminación con respecto al resto de objetos en movimiento. De esta forma, cada algoritmo de seguimiento será el encargado de decidir qué hacer con dicha información según sus propias necesidades, decidiendo así que tipo de registro, estabilización o corrección hacer.

Bajo estas circunstancias, y teniendo en cuenta las herramientas disponibles, se diseñó un sistema de estabilización de vídeo inter-frame según el diagrama de la Figura 6.1.

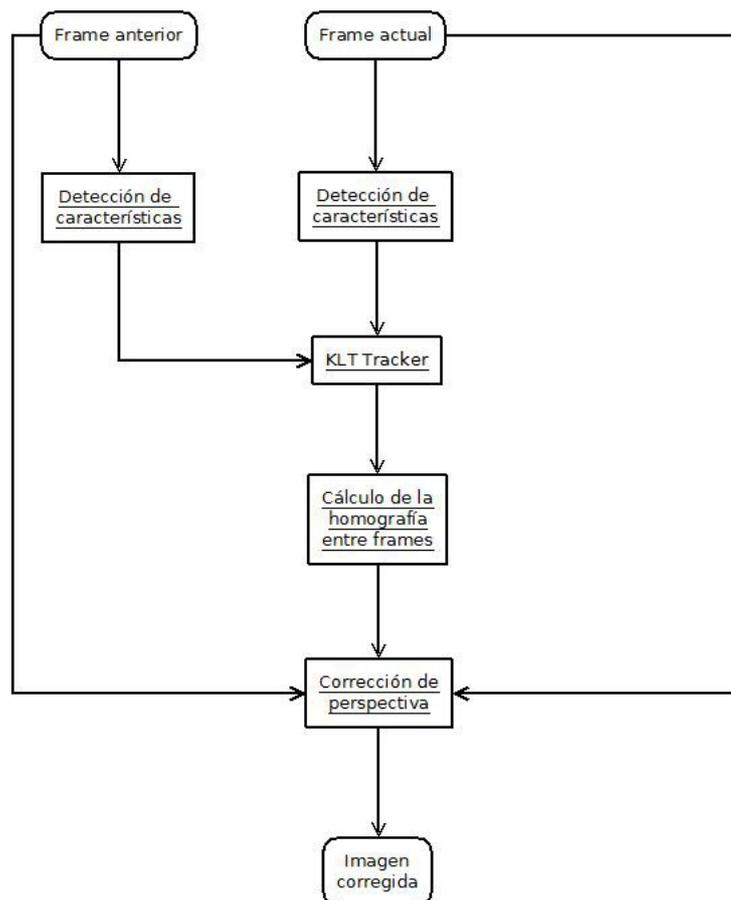


Figura 6.1: Sistema de estabilización de vídeo inter-frame

6.2 Detección de características.

Existen varias opciones a la hora de extraer características de una imagen para realizar luego la correspondencia puntual, pero en el caso de un estabilizador de vídeo éstas deben cumplir al menos los siguientes puntos:

- Ser muy numerosas, ya que estadísticamente las características del fondo deben de ser bastante superiores a las características de los objetos en movimiento. Por otro lado, y aunque técnicas como RANSAC permiten mayor flexibilidad en este punto, es siempre recomendable tener el mayor número de características posible.
- Las características deben ser suficientemente descriptivas, para que así puedan ser localizadas de forma exacta, no dando lugar nunca a ambigüedades.
- Las características tienen que ser lo más repetibles posible.
- Cuanta mayor invariancia en las mismas mejor.
- Su cómputo debe ser lo más rápido posible.

Teniendo en cuenta estas premisas, muy probablemente las mejores opciones sean:

- El detector de esquinas de Harris y Stephens [18].
- El detector de esquinas de Shi-Tomashi [19].
- El detector rápido de esquinas FAST [20].
- La alternativa rápida a descriptores más invariantes que las esquinas: ORB [21].

No obstante, y para ser coherentes con la técnica de tracking de características utilizada, es decir las KLT [22, 23 y 19], en nuestro caso se hizo uso del detector de características de Shi-Tomashi [Figura 6.2].



Figura 6.2: Detección de características mediante el método de Shi-Tomashi.

El detector de características de Shi-Tomashi está basado en el detector de esquinas de Harris, donde para cada punto de la imagen se calcula su valor de *cornicidad*, el cual no es otra cosa que una medida de la variabilidad de la imagen en dicho punto. Se puede ver una descripción detallada del método de Shi-Tomashi en [19].

6.3 Las KLT

El comúnmente conocido como “KLT feature tracker”, esta originalmente basado en el trabajo desarrollado por Lucas-Kanade para el cálculo del flujo óptico en [22], trabajo el cual fue terminado posteriormente por Tomashi-Kanade en [23], y que fue finalmente presentado y aclarado por Shi-Tomashi en [19]. Esta técnica está basada en el seguimiento de los puntos característicos, mediante las ecuaciones desarrolladas por Lucas-Kanade para el cálculo del flujo óptico, que a su vez implementan el método iterativo de Newton-Raphson para una búsqueda de ascenso por gradiente.

En las KLT se parte de los puntos característicos obtenidos mediante el método de Shi-Tomashi explicado en el apartado anterior, para así posteriormente calcular para cada punto su vector de desplazamiento respecto al tiempo. Para ello, centrado en cada punto característico se define una ventana (en nuestros experimentos de 11), sobre la cual es realizado el ascenso por gradiente mediante el método desarrollado por Tomashi-Kanade en [23].

No obstante, y como se puede deducir de un método iterativo de ascenso por gradiente, esta aproximación tan sólo es válida si el desplazamiento es lo suficientemente pequeño como para que el gradiente no cambie de dirección. Por ello y para desplazamientos mayores, se debe aplicar dicho procedimiento de forma piramidal, tal y como se realiza en [24 y 25].



Figura 6.3: Resultados de las KLT.

6.4 Corrección de perspectiva

Una vez realizada la puesta en correspondencia de los puntos característicos entre las dos imágenes consecutivas, es necesario realizar una corrección de perspectiva sobre alguna de ellas. Para ello es necesario obtener la homografía que permite la proyección de las coordenadas espaciales desde una imagen a la otra, y viceversa. Dicha matriz de transformación homográfica actúa como matriz de proyección entre los espacios de coordenadas de ambas imágenes, permitiendo así no sólo hacer la corrección de perspectiva, sino que en caso de no hacerse un registro de imagen, también puede ser necesario su uso dentro de los algoritmos de seguimiento.

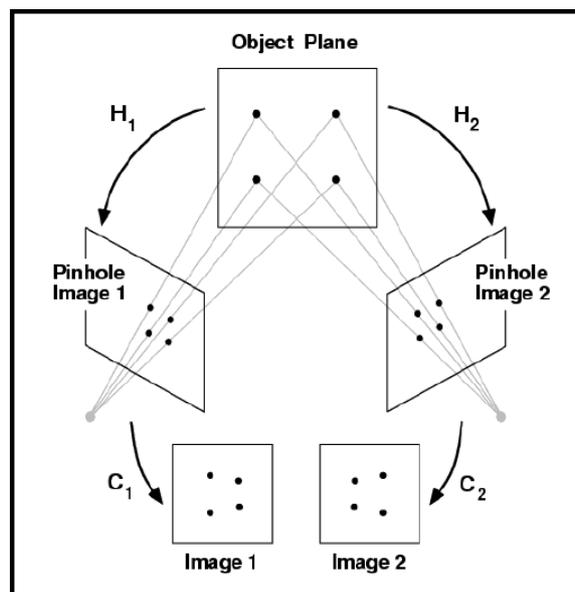


Figura 6.4: Relación homográfica entre dos vistas

La forma de obtener dicha matriz de proyección es mediante el cálculo de la matriz de transformación H , comúnmente denominada homografía. Esta matriz tal y como se puede ver en la [Figura 6.4], permite proyectar los puntos pertenecientes al plano de una vista concreta, sobre el plano de otra vista diferente.

El algoritmo original para el cálculo de la homografía entre dos vistas es el comúnmente conocido como algoritmo de la DLT (Direct Linear Transform), el cual puede encontrarse descrito con detalle en [46].

No obstante, el error numérico introducido por los outliers puede llegar a ser un verdadero problema en la mayoría de los casos, por lo que con el fin de contrarrestar dicho problema, se hará uso de RANSAC para un ajuste robusto del sistema (véase [46]), el cual puede llegar a permitir hasta un 40% de outliers.

6.5 Implementación y resultados

Como trabajo previo a este proyecto ya se tenían implementaciones propias tanto de la detección de características, como de la parte para el cálculo de la matriz homográfica, teniendo tan sólo que implementar la parte para el cálculo del flujo óptico para las KLT. Por ello y con objetivo de su estudio para su posterior aplicación y uso en los algoritmos de seguimiento implementados posteriormente, se realizó una implementación propia de la técnica propuesta por Lucas-Kanade para el cálculo del flujo óptico. No obstante, la complejidad de implementar también la parte piramidal y sobre todo la de optimizar todo el proceso, excedía claramente el objetivo del proyecto, por lo que desde el principio se decidió utilizar las implementaciones proveídas por la librería de OpenCV.

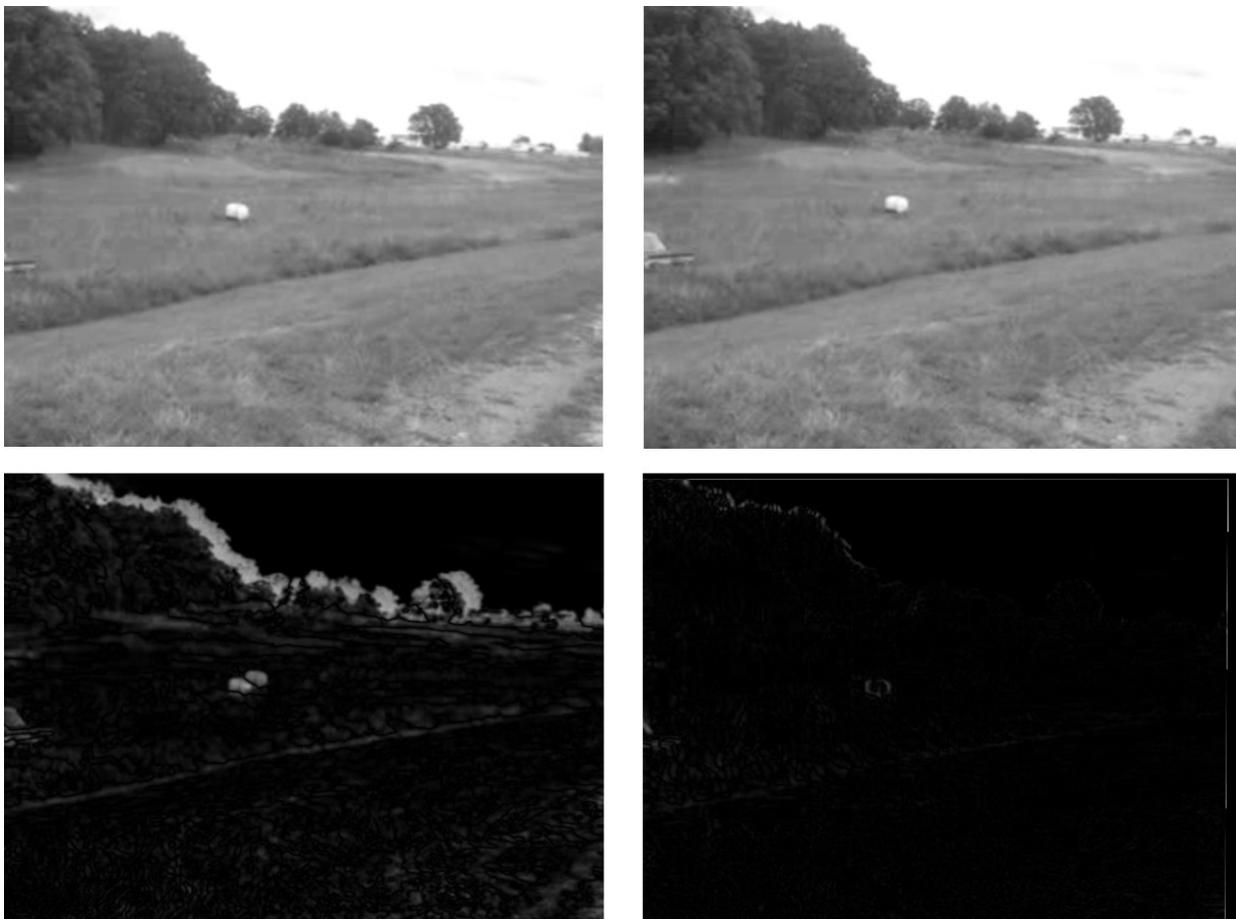


Figura 6.5: Imágenes diferencia: Las dos imágenes de arriba son dos frames consecutivos, la imagen de abajo a la izquierda es la imagen diferencia antes de estabilizar, y la imagen de abajo a la derecha es la imagen diferencia después de estabilizar.

En cuanto a los resultados finales de la estabilización de vídeo, estos fueron bastante satisfactorios, tal y como se pueden observar en las imágenes diferencia antes y después de estabilizar mostradas en la [Figura 6.5].

Se pueden ver también los resultados de la evaluación del algoritmo implementado en las gráficas de la [Figura 6.6]. Dichas gráficas representan la energía de las dos imágenes diferencia (antes y después de estabilizar) a lo largo de todo el vídeo. El vídeo utilizado para la evaluación eran un vídeo de un vehículo todo-terreno con cámara embarcada y circulando por terreno abrupto. En esta gráfica se puede ver como la energía de la imagen diferencia es minimizada, dejando tan sólo:

- El contenido residual causado por la interpolación a la hora de deformar la imagen.
- El error provocado por los cambios de perspectiva, y que a mayor desplazamiento físico de la cámara mayor error.
- El movimiento provocado por los propios objetos móviles dentro de la escena.

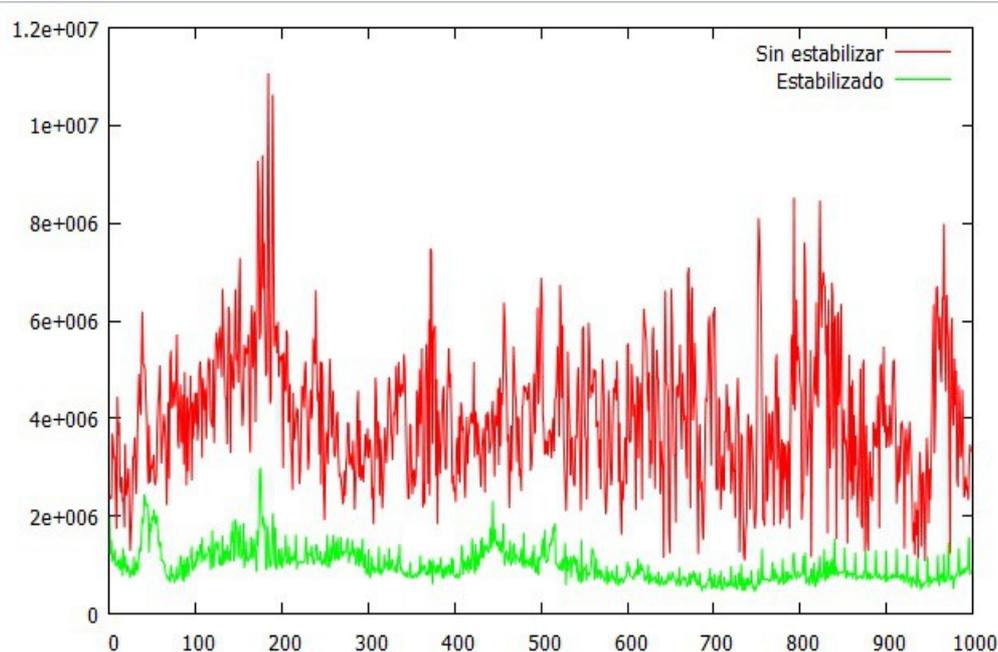


Figura 6.6: Resultados del estabilizador de vídeo. La gráfica muestra la proyección de energía por frame para el vídeo original y el vídeo estabilizado.

En el caso concreto de la cámara móvil contemplada en este proyecto, debido a la enorme velocidad de movimiento de la cámara y la propia naturaleza del sensor, algunas imágenes quedaban bastante deformadas y difuminadas, lo cual las hacía inútiles a efectos prácticos tal y como se puede ver en [Figura 6.7]. No obstante y gracias al gran deterioro de dichas imágenes, su detección y eliminación será relativamente fácil, ya que el número de características detectadas en dichas imágenes será exponencialmente inferior a lo normal.



Figura 6.7: Imagen antes y durante el movimiento de la cámara.

Por otro lado y como ya se ha introducido anteriormente, la corrección de perspectiva no es nunca perfecta, siendo sobre todo perjudicial en los bordes de los objetos, donde los gradientes se caracterizan por ser bastante inestables, introduciendo por lo tanto bastante ruido que luego será necesario filtrar o ignorar, sobre todo si se desea utilizar cualquier tipo de segmentación de fondo. No obstante, en lo que a nosotros respecta y teniendo en cuenta la naturaleza de las técnicas utilizadas posteriormente, dicho problema no afectará demasiado.

7 Sistema de seguimiento

El módulo de seguimiento es el principal objetivo del presente trabajo, presentando también el principal reto del mismo. Por otro lado, parte del desarrollo de la primera versión del sistema de seguimiento fue realizado junto al sistema de estabilización de vídeo presentado en el apartado anterior, lo cual era parte esencial de la motivación para la implementación del sistema de estabilización.

El módulo de seguimiento implementado, tal y como ya se introdujo en el apartado de arquitectura, fue diseñado con la idea de hacer totalmente intercambiables los distintos módulos de seguimiento implementados mediante la propia configuración del módulo principal, la cual se puede realizar a través del fichero de configuración de la aplicación. De esta forma, con tan sólo cambiar una línea del fichero de configuración, se puede seleccionar un método de seguimiento u otro. Esto se hizo así para facilitar el desarrollo del sistema de seguimiento, ya que con ello no sólo se permite la rápida evaluación de los nuevos algoritmos, sino también una fácil comparación con respecto a los implementados anteriormente.

Para hacer esto posible, fue necesario definir tres elementos principales:

- Una clase abstracta que definiera la interfaz para todo módulo de seguimiento a implementar [Figura 7.1].
- Un módulo de inicialización interactiva del objeto a seguir común a todos los módulos de seguimiento [Figura 7.1].
- Un módulo de cámara virtual que permitiese la adquisición a partir de vídeo (véase [Figura 4.1] del capítulo 4).

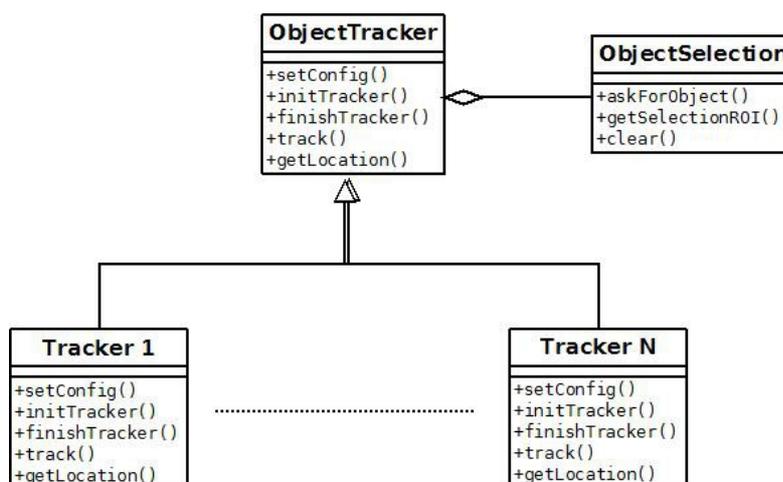


Figura 7.1: Diseño del módulo abstracto del sistema de seguimiento.

Gracias a esto, es el programa principal el que dependiendo de la configuración de la aplicación instanciará un módulo u otro. De esta forma y con el módulo de selección de objeto común a todos los módulos de seguimiento, cada módulo tan sólo tendrá que encargarse de la implementación específica del sistema de seguimiento para la que haya sido diseñado.

En cuanto al módulo de selección de objeto, este tan sólo se encarga de presentar el flujo de vídeo al usuario, mientras que espera a que el propio usuario defina un polígono válido sobre el objeto al que desea realizar el seguimiento.

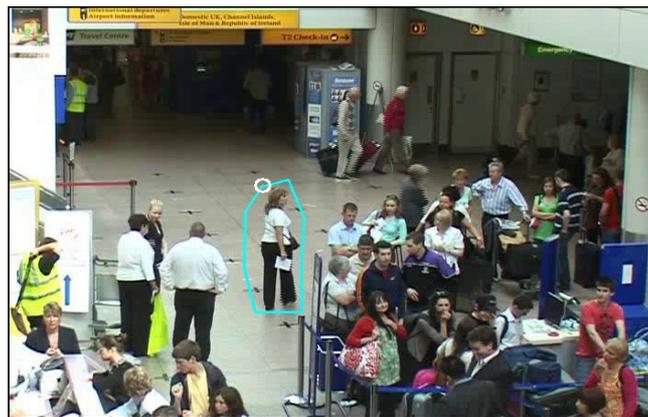


Figura 7.2: Pantalla de selección de objeto sobre el que realizar el seguimiento.

A lo largo de este trabajo se desarrollaron un total de 10 algoritmos de seguimiento distintos, de los cuales a continuación se pasarán a detallar tan sólo los más relevantes y de mayor impacto para la investigación.

7.1 Seguimiento mediante KLT's

7.1.1 Diseño

El primer algoritmo de seguimiento implementado fue basado exclusivamente en los algoritmos de seguimiento por características mediante las KLT, para lo cual se hizo uso de parte del código desarrollado para el sistema de estabilización de vídeo.

Hay que tener en cuenta que todos los algoritmos de seguimiento son inicializados nada más instanciarlos con la región de interés seleccionada por el usuario durante la etapa de selección del objeto, estando todos ellos también obligados a mantener en todo momento actualizada tanto la región del objeto como su centro de masas.

Partiendo de esto, y para esta primera versión se realizó el diseño de [Figura 7.3], donde la idea principal se basaba, al igual que en la mayoría de sistemas de seguimiento por KLT, en mantener un

conjunto de características sobre las que ir realizando un seguimiento continuo a la vez que se van añadiendo nuevas características según se va teniendo la oportunidad.

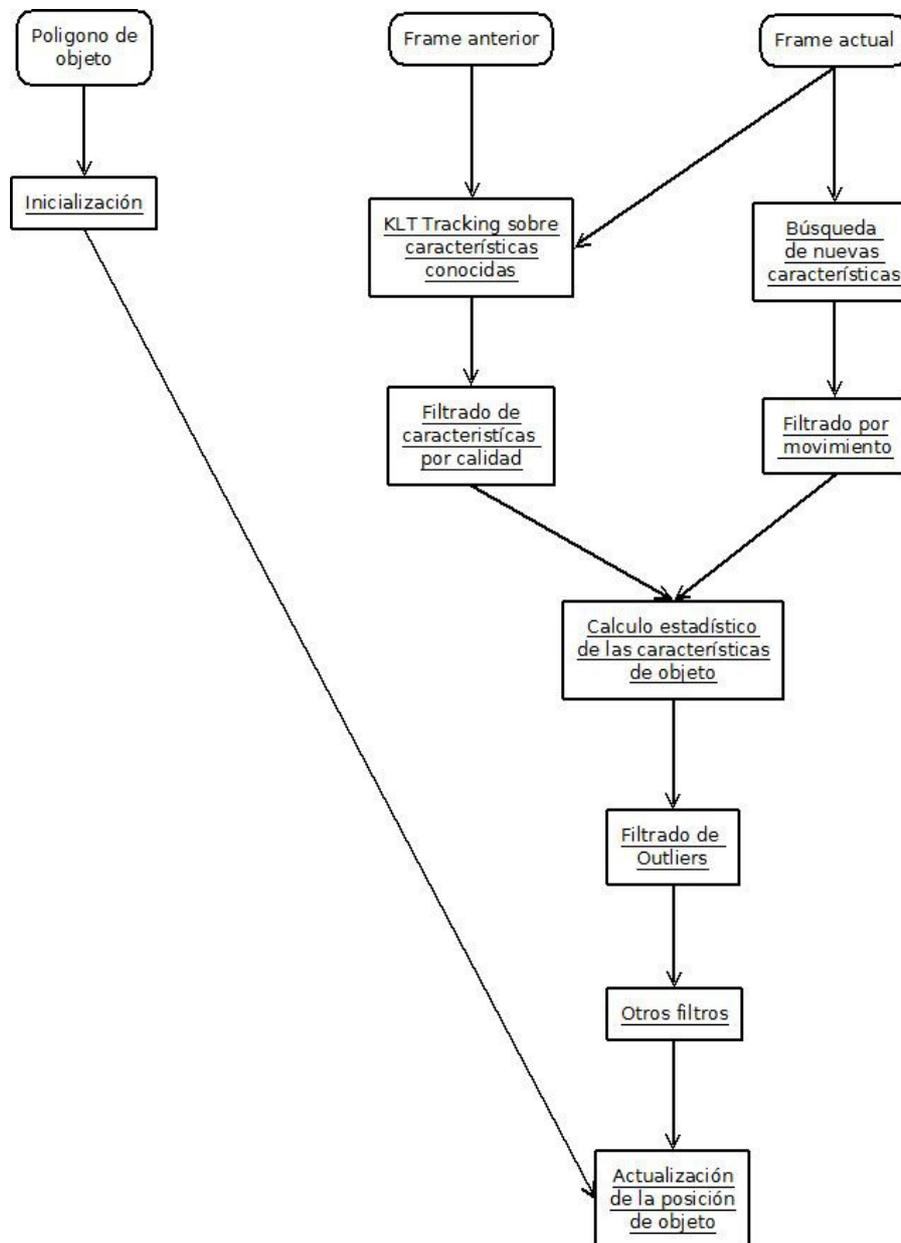


Figura 7.3: Diseño inicial del sistema de seguimiento mediante KLT

No obstante y como se puede ver en el diseño de [Figura 7.3], esto no es suficiente, ya que se hacen necesarios diversos mecanismos de filtrado que permitan ir seleccionando y filtrando las características más útiles, desechando de esta forma las menos útiles. La razón de esto es la enorme degeneración producida a nivel de característica con respecto al tiempo, ya que las características tienden a perderse por el camino (quedándose ancladas al fondo), re-ubicarse, o concentrarse todas en los mismos puntos.

Por otro lado, las nuevas características a detectar también tienen que ser significativas, es decir, pertenecer al objeto. Para ello se realizó un filtrado por movimiento, lo cual no tiene nada que ver con una segmentación de movimiento, ya que como se explicó anteriormente en el apartado de estabilización de vídeo, aunque la estabilización es buena, la estabilidad en los bordes es poco robusta a causa del ruido. Y es en este punto cuando se hace uso de la información obtenida a partir del flujo óptico proveído por las KLT, ya que al tener una medida bastante exacta del movimiento de las mismas, es relativamente fácil filtrar y seleccionar las que realmente están asociadas a un movimiento real.

7.1.2 Cálculo estadístico y filtrado de outliers

La principal ventaja del seguimiento de características KLT es no sólo la gran cantidad de características que se pueden detectar y manejar, sino también el seguimiento continuo que se le pueden realizar a las mismas. Pero desgraciadamente esto también conlleva al mayor inconveniente del mismo: la poca estabilidad de las características a lo largo del tiempo. La causa de ello es que según se pierde cornicidad en las características, éstas tienden a re-ubicarse en los bordes más cercanos y pronunciados, lo que provoca entre otras cosas:

- La pérdida de algunas características mientras estas se van quedando pegadas al fondo. Esto es causado principalmente por la evolución en la apariencia de dichas características, cuando estas en lugar de tender a re-ubicarse hacia el objeto, lo hacen hacia el fondo.
- La concentración de la mayoría de las características en las mismas zonas, y que generalmente es en los bordes que delimitan al objeto.

La única forma de resolver esto es detectar y filtrar dichas características, lo cual no es una tarea fácil, ya que no se ha encontrado mucha información al respecto en la literatura actual, pues en la mayoría de las publicaciones o implementaciones revisadas, este problema suele ser ignorado. En [26] este problema intenta ser resuelto directamente rompiendo la continuidad del seguimiento a través de la independencia del mismo a lo largo del tiempo, es decir, que en cada intervalo de tiempo se detectan y siguen todas las características para posteriormente eliminarse, y por lo tanto volver a ser detectadas íntegramente en el siguiente intervalo de tiempo.

Teniendo en cuenta todo esto y con el objetivo de filtrar tan sólo las características más útiles y relevantes, se diseñó un sistema de cálculo estadístico de las trayectorias, ángulos y desplazamientos más recurrentes a nivel de objeto, pudiendo de esta forma eliminarse todas aquellas características que se alejasen lo suficiente del modelo generado.

El modelo de filtrado estadístico implementado, es típico en muchos de los sistemas de detección de anomalías utilizados en el ámbito del aprendizaje automático, de los cuales se puede ver una revisión del estado del arte en [27].

En nuestro caso concreto, primero generamos para cada característica un pequeño descriptor de dos valores con:

- La magnitud del vector de movimiento de esa característica.
- El ángulo del vector de movimiento de esa característica.

Posteriormente se realiza un modelado gaussiano sobre dicho conjunto de descriptores, donde para cada componente son calculadas su media y su desviación típica.

Finalmente y para filtrar todas aquellas características que se alejen lo suficiente de la media, se hace uso del teorema de desigualdad de Chevishev, donde se filtran todos aquellos elementos cuyo valor este por encima de “K” veces la desviación típica, siendo “K” la constante de Chevishev y que en todos nuestros experimentos fue de 2.5.

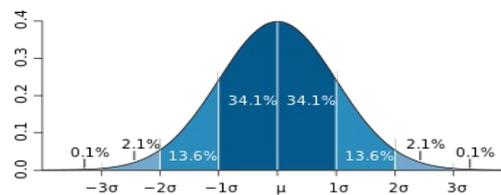


Figura 7.4: Modelo simple de discriminación gaussiana definido por Chevishev.

En la [Figura 7.5] se pueden ver algunos ejemplos del filtrado realizado en uno de los vídeos de prueba, donde se puede observar claramente la diferencia de calidad en las características utilizadas para el seguimiento. Como se puede observar, en las dos imágenes inferiores donde el filtro está activo, las características utilizadas presentan todas una buena calidad y son todas ellas pertenecientes al objeto, mientras que en las dos imágenes superiores donde el filtro no está activo, el resultado es bastante malo y muchas de las características han sido perdidas en el camino.

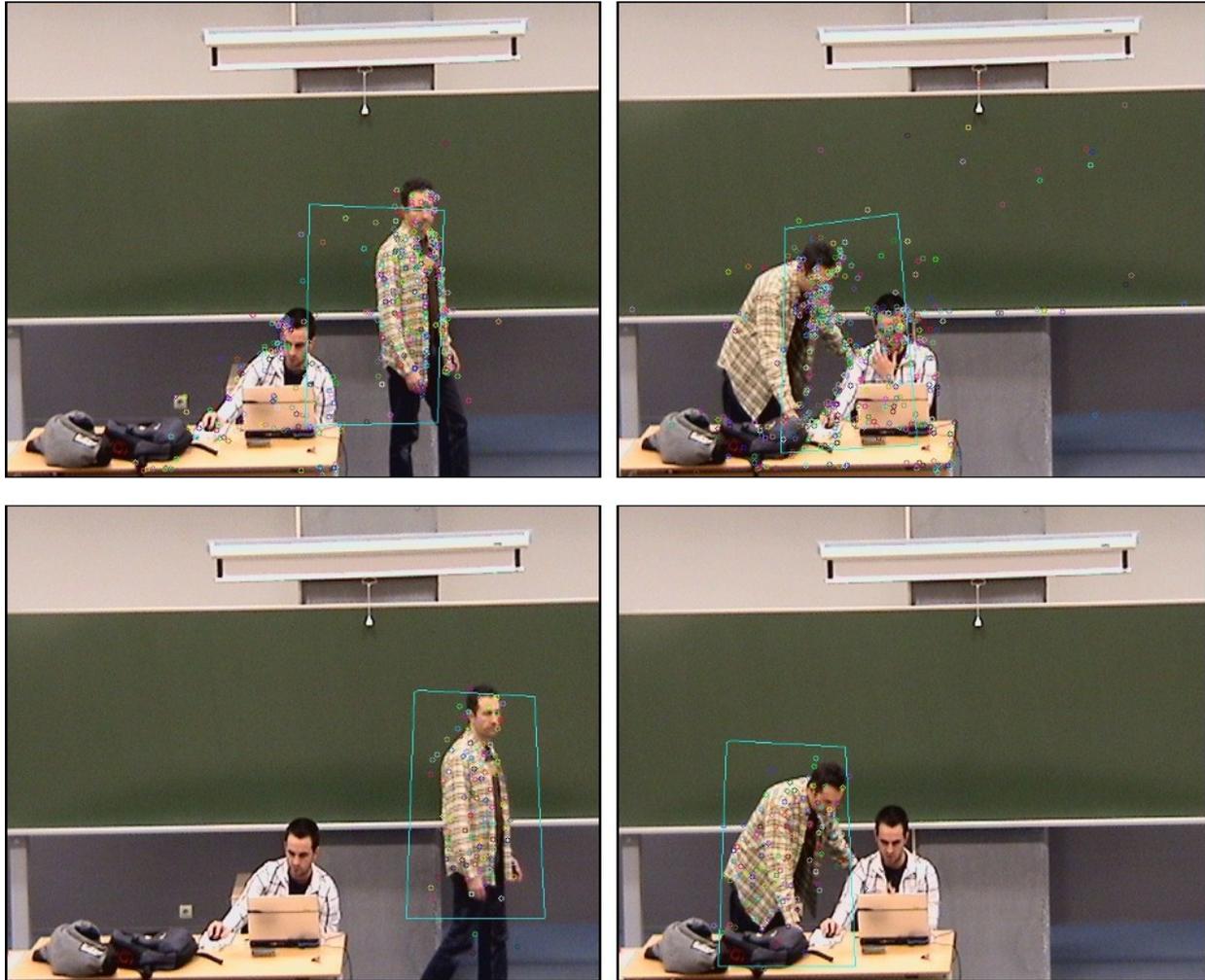


Figura 7.5: Imágenes del filtrado estadístico. En las dos imágenes superiores el filtro estadístico no se encuentra activo, mientras que en las dos imágenes inferiores el filtro estadístico sí que está activo.

Por otro lado, también se aplicaron otros filtros de utilidad con objetivos diversos:

- Eliminar las concentraciones de puntos.
- Eliminar los puntos con gradiente plano.
- Eliminar los puntos de baja calidad, es decir, puntos que no terminaron de converger o que presentan una alta diferencia en su vecindad respecto al punto original.
- Eliminación de puntos estáticos a la hora de detectar nuevas características.

7.1.3 Impresiones

En general, las primeras impresiones obtenidas mediante esta primera versión del sistema fueron bastante satisfactorias, sobre todo en comparación con los resultados presentados por el proyecto anterior. No obstante, en un principio los primeros intentos de utilizar las KLT como sistema de

seguimiento no fueron muy satisfactorios, pero tras el diseño e implementación tanto del sistema de filtrado estadístico como del sistema de renovación de características, la mejora fue realmente notoria.

Sin embargo, y aunque de forma aislada el seguimiento parecía bueno, un seguimiento “exclusivo” por KLT presenta un importante conjunto de inconvenientes difícilmente salvables sin ser combinado con otros métodos:

- Imposibilidad de lidiar con oclusiones totales.
- Imposibilidad de lidiar con algunas oclusiones parciales que se lleven la mayor parte de las características.
- Imposibilidad de distinguir entre objetos, y por lo tanto imposibilidad de lidiar con cruces de objetos.
- Dificultad de recuperar al objeto una vez perdido.
- Etc.

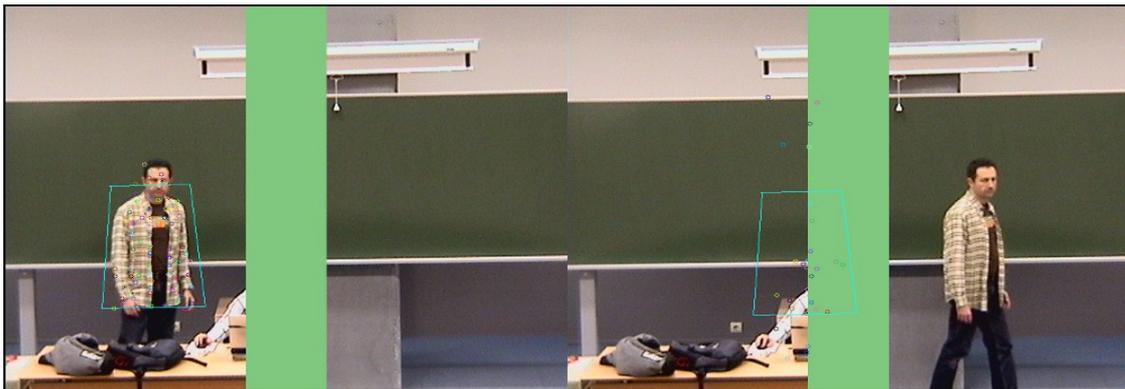


Figura 7.6: Pérdida de objeto ante oclusión.

En general, se puede decir que las KLT sólo aprovechan la información de movimiento del objeto a través de su flujo óptico para realizar su seguimiento. Esto hace que sus principales virtudes sean, tanto su gran capacidad para seguir movimientos rápidos, como su facilidad para detectar nuevas características de objeto según van apareciendo. Mientras que por otro lado, su mayor carencia y muy probablemente el origen de la mayoría de sus inconvenientes, sea su incapacidad para modelar de alguna manera al objeto sobre el que se está realizando el seguimiento.

7.2 Seguimiento mediante Filtros de Partículas de movimiento

7.2.1 Filtros de partículas

Como se pudo ver en el análisis del estado del arte realizado anteriormente, una de las vías de investigación más recurridas en los últimos tiempos en lo referente al ámbito del seguimiento automático, han sido los modelos basados en filtros de partículas, y precisamente por ello el segundo sistema de seguimiento implementado se basó precisamente en los mismos.

Las aplicaciones de los filtros de partículas al seguimiento de objetos tienen su base en los modelos de Bayes, los cuales modelan la incertidumbre del problema mediante dos elementos: el estado y observaciones actuales. Estos modelos, en espacios discretos suelen tratarse desde un prisma Markoviano, donde el estado actual depende exclusivamente del estado inmediatamente anterior, y la observación actual depende exclusivamente del estado actual. Se puede ver una descripción detallada de ambos modelos en [28].

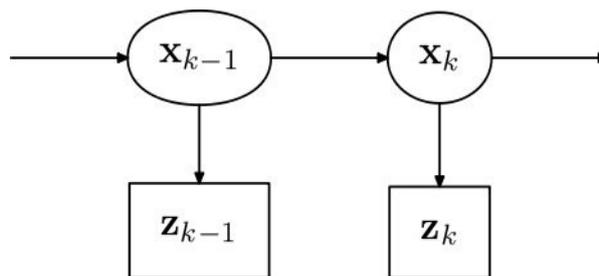


Figura 7.7: Modelo Oculto de Markov, donde X_k representa el estado actual y Z_k la observación actual.

Bajo este modelo teórico y para dar solución a problemas no lineales en espacios continuos, surgieron los comúnmente denominados filtros de partículas, los cuales proponen una solución basada en los modelos de integración de Monte Carlo y que por primera vez son utilizados en imagen en [29].

Los filtros de partículas son relativamente sencillos de implementar siguiendo un breve y bien definido conjunto de pasos:

- Inicialización: Inicialmente se definen un conjunto determinado de partículas, cuya posición es asignada aleatoriamente.
- Iterativamente:
 - Actualización: Se actualizan los pesos de las partículas según el modelo de probabilidad utilizado, lo cual en imagen será una métrica de similitud. De esta forma, las partículas tendrán un peso proporcional a la similitud que presente la observación correspondiente a su posición respecto al modelo de objeto, produciendo así pesos más grandes cuanto más se parezca la partícula al objeto.

- Muestreo: Posteriormente se hace un muestreo aleatorio de las partículas según sus pesos, de tal forma que las partículas con mayor peso, y por lo tanto más similares al modelo de objeto, sean más fácilmente seleccionables.
- Dispersión y Predicción: Finalmente y en base al movimiento observado anteriormente, se hace para cada partícula una predicción de cuál sería su posición en el siguiente intervalo de tiempo. No obstante, dicha predicción es afectada mediante un valor de ruido aleatorio según un parámetro de dispersión. Esto permite a varias selecciones de la misma partícula, seguir caminos diferentes en la búsqueda del objeto durante el siguiente intervalo de tiempo, además de permitir también la dispersión de las partículas en caso de pérdida u oclusión del objeto.

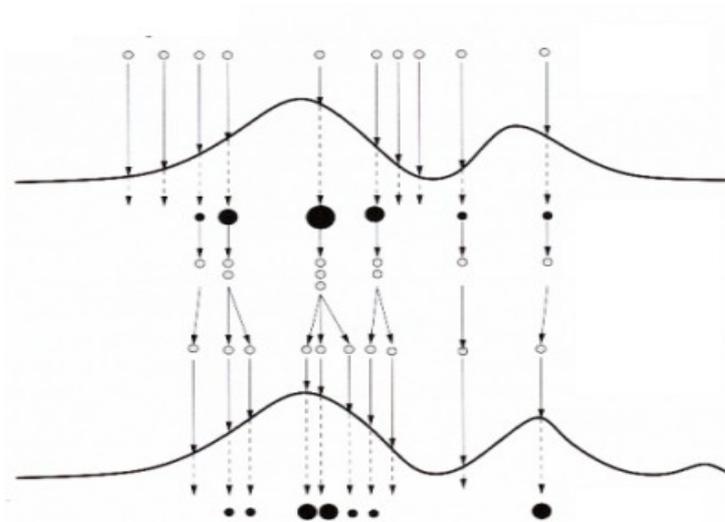


Figura 7.8: Representación gráfica de un filtro de partículas.

Se puede ver una definición completa de los sistemas basados en filtros de partículas en [28]. Como se puede observar, los modelos basados en filtros de partículas presentan un sistema de búsqueda bastante eficiente, que comparte principios muy similares a los utilizados en los sistemas de computación evolutiva para búsquedas en espacios infinitos (como por ejemplo en [30]).

7.2.2 Diseño

En imagen, tal y como se ha mostrado en el apartado anterior los filtros de partículas nos permiten, dado un estado y un modelo de objeto concretos, buscar el foco de mayor similitud con respecto a dicho modelo de objeto en el siguiente intervalo de tiempo. En el campo del seguimiento automático, usualmente los filtros de partículas hacen uso de la posición anterior del objeto como estado previo (a través de las partículas generadas en el intervalo anterior), y del histograma de color o descriptor de forma del objeto como modelo de similitud (véanse por ejemplo [3], [31]-[36]). No obstante, y para el diseño de este algoritmo de seguimiento, no se utilizaron ninguna de dichas configuraciones clásicas.

En esta etapa de desarrollo, todavía no se pretendía abordar el seguimiento del objeto por sus características de color o forma, sino por su flujo de movimiento. Por ello, y sobre los valores del flujo óptico disperso obtenido gracias al uso de las KLT, se diseñó un filtro de partículas que permitiese seguir al objeto según dicho movimiento, concentrándose así las partículas en las zonas de mayor movimiento. De esta forma, las partículas son concentradas en las zonas donde existe una mayor coherencia y cantidad de movimiento, filtrándose así automáticamente las características de poca fiabilidad o relevancia.

Cabe destacar que para ello, las características se filtraron por cantidad de movimiento, y el cálculo del flujo óptico se realizó de forma independiente para cada par de cuadros consecutivos (esto último a partir de aquí se referirá como modelado con independencia respecto al tiempo en el proceso de detección). De esta forma las características no se conservaban a lo largo del tiempo, sino que por el contrario eran re-calculadas por completo en cada intervalo de tiempo, eliminando con ello el problema de la degeneración en las mismas.

En cuanto a la métrica definida para la localización del objeto a través de su concentración de movimiento, se definió la siguiente:

$$F(x) = 1 / \left(\sum_{i=1}^N (D(x, C[i])) \right)$$

Donde “x” es la partícula a evaluar, “C” es el conjunto de características detectadas que superaron el umbral de movimiento, “D” es la función que mide la distancia entre la partícula actual “x” y la característica “i” a evaluar, y “N” es el número total de características.

Por otro lado y para relajar las restricciones de localización referentes al estado previo, se modificó el modelo de muestreo para que un pequeño porcentaje de las partículas se generasen de forma aleatoria por toda la imagen (de una forma muy similar a como lo haría un sistema evolutivo), permitiendo de esta forma ampliar al máximo el rango de búsqueda cuando el objeto se hubiera perdido.

Se puede ver el diseño final de este algoritmo de seguimiento en la Figura 7.9.

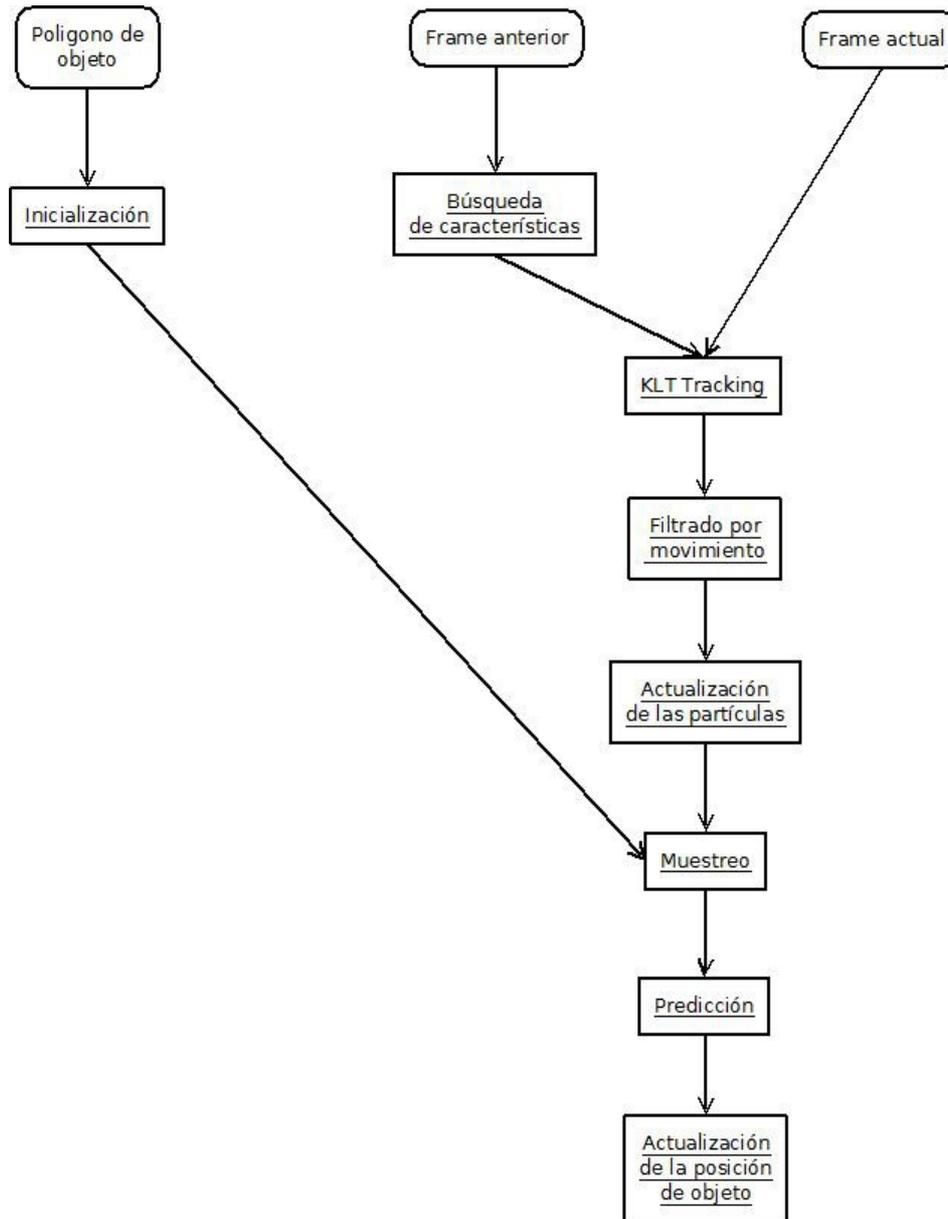


Figura 7.9: Diseño inicial del sistema de seguimiento por filtros de partículas de movimiento.

7.2.3 Impresiones

Como resultado de la presente implementación, se obtuvo un algoritmo el cual realizaba un seguimiento bastante efectivo de las mayores concentraciones de movimiento, permitiendo filtrar así de forma automática las características de objeto de poca fiabilidad o relevancia.

Como ventaja principal, el algoritmo proveía de una mucho mayor probabilidad de recuperar objetos perdidos. Pero por contra, al basarse exclusivamente en el movimiento absoluto de forma independiente a lo largo del tiempo, el algoritmo era totalmente incapaz no sólo de discriminar entre distintos objetos que se pudiesen cruzar entre sí, sino también de distinguir entre los distintos sub-

objetos pertenecientes al propio objeto a seguir, de tal forma que al no mantenerse registro de las características anteriores, si una parte del objeto se movía de forma independiente, esto podía despistar ligeramente al algoritmo e incluso provocar la pérdida del objeto.

Desgraciadamente, y a pesar de un conjunto de ventajas que podrían ser muy útiles en otras tareas, en lo referente a los objetivos perseguidos para el presente proyecto, el algoritmo no proveía de suficientes ventajas sobre la anterior implementación, mientras que si producía un considerable aumento del coste computacional.

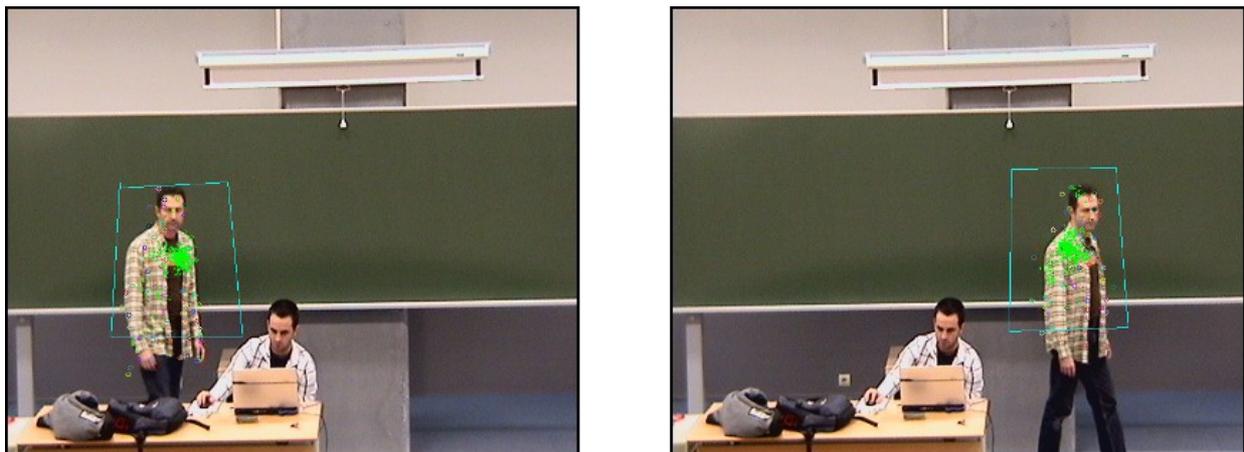


Figura 7.10: Muestra de las concentraciones de las partículas en el objeto a seguir.

No obstante, todo esto dejaba claramente patente algo que ya se sabía, y era la insuficiencia de información en las características de objeto utilizadas a la hora de tratar de seguir a un objeto tan sólo por su movimiento. Aunque por otra parte y como se verá en siguientes implementaciones, los principios que forman la base de los filtros de partículas, proporcionarán una herramienta tremendamente útil a la hora de añadir otros tipos de información a las características utilizadas, a la vez que proporcionarán una forma bastante efectiva de medición y filtrado de las mismas según su calidad.

Cabe destacar de este algoritmo, la fiabilidad y rapidez de agrupación de las partículas a la hora de detectar movimiento, lo que por ejemplo podría ser una herramienta especialmente útil a la hora de detectar el nacimiento de objetos en los sistemas de seguimiento múltiple. Para ello tan sólo habría que analizar de forma un poco más exhaustiva los datos de las concentraciones de las partículas, y una vez inicializado el objeto, excluir las características detectadas en los siguientes intervalos de tiempo.

7.3 Seguimiento mediante Partículas KLT (PKLT)

7.3.1 La importancia de aprender y combinar características

Como ya se expuso anteriormente, y para seguir un objeto a largo plazo, el uso exclusivo de sus características de movimiento resulta claramente insuficiente. Por ello y como segunda fase de la investigación, se pretendía crear un modelo que permitiese el uso de distintos tipos de características de forma simultánea, sensata y coherente.

De entre los algoritmos de seguimiento omitidos por su falta de eficacia y relevancia, hubo un par de ellos basados exclusivamente en el uso de filtros de partículas sobre los histogramas de color del objeto. Los resultados de dichos algoritmos fueron bastante insuficientes, presentando también un gran dilema: definir un modelo fijo de objeto durante la inicialización, o ir aprendiéndolo y evolucionando a lo largo del tiempo para ajustarse a los posibles cambios del mismo. Estas fueron precisamente las dos implementaciones realizadas, y en ambas se llegó a un callejón sin salida (al menos utilizando exclusivamente la información del color):

- Modelo fijo de objeto: En ésta vertiente y como era predecible el objeto era seguido correctamente tan sólo al principio, ya que en cuanto el objeto cambiaba su iluminación lo suficiente, pasaba por una región con colores lo suficientemente similares, o cambiaba su perspectiva variando lo suficiente su modelo de color, el objeto era perdido con facilidad.
- Modelo adaptativo de objeto: Aquí el problema no era el planteamiento auto-adaptativo, sino la incapacidad de extraer características lo suficientemente fiables. Ya que el envenenamiento producido por la inestabilidad en los bordes, por el fondo tanto estático como móvil, o por otros objetos que se cruzasen con el objetivo, provocaba una corrupción sustancial en el modelo de objeto que desembocaba en la mayoría de los casos en una temprana pérdida del objeto.

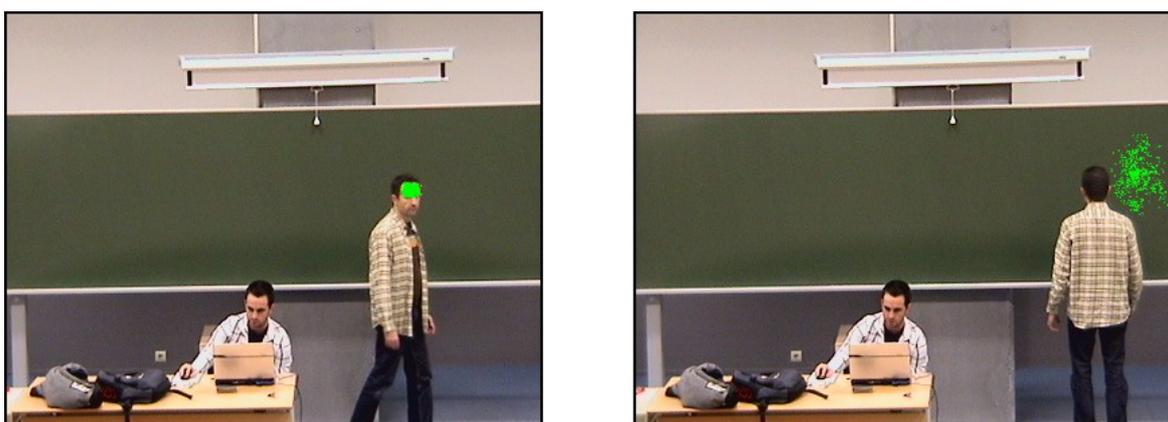


Figura 7.11: Pérdida de objeto por envenenamiento durante una rotación (PFCOLOR).

Con todo ello y bajo esta perspectiva, se tenían claros varios puntos:

- Se requería modelar al objeto mediante la combinación o fusión de distintos tipos de características: movimiento, color, forma, etc.
- Se requería auto-adaptación.
- La adaptación necesitaba ser lo suficientemente robusta y flexible como para filtrar y minimizar el impacto de los distintos factores que pudieran envenenar o corromper el modelo de objeto.

7.3.2 Diseño

Llegados a este punto de la investigación y bajo las nuevas premisas extraídas en el apartado anterior, se hacía necesario un algoritmo que tuviese en cuenta distintos tipos de características a la vez que estas puedan ser renovadas continuamente y filtradas en caso de degeneración. Para ello se diseñó un algoritmo que utilizando como base el método de detección y seguimiento de características mediante el uso de las KLT, se le pudiesen aplicar los principios de actualización y filtrado característicos de los filtros de partículas, usando para ello cualquier tipo de información (movimiento, color, forma, etc). No obstante, en este caso en concreto y de momento, tan sólo se utilizará la información de color y de movimiento como características de filtrado.

Tal y como se puede ver en el diseño del sistema en la Figura 7.12, para este algoritmo de seguimiento se define un sistema de partículas las cuales son renovadas continuamente, y filtradas en base la degeneración de las mismas, donde la medida de la degeneración es realizada en base a su información de color y de movimiento.

Inicialmente y al igual que en algoritmo original, las partículas son detectadas mediante el método de Shi-Tomashi para posteriormente ser calculado su flujo óptico disperso mediante el uso de las KLT. Un vez detectadas las partículas, éstas inician sus descriptores tanto de color como de movimiento, de tal forma que si posteriormente se produce una divergencia lo suficientemente grande con respecto a dichos descriptores originales, se entenderá como indicador de degradación de la misma y será eliminada (además de pesada acorde a dicha medida).

En cuanto al cálculo de los descriptores de movimiento y color, estos se definen de manera simple e inequívoca, ya que para el descriptor de movimiento se utiliza el vector de movimiento definido por el flujo óptico de la misma en cada intervalo de tiempo, y el de color es calculado en base a la media de los valores RGB de la partícula dentro de una pequeña ventana (utilizando para ello el mismo valor de ventana que el utilizado para las KLT, que en nuestro caso concreto es siempre 11). El criterio utilizado para la definición de estos descriptores fue totalmente empírico, pudiendo éstos ser sustituidos en cualquier momento por otros que provean de mayor o más robusta información a la hora de pesar y filtrar las características.

En base a todo esto, se define para cada partícula un peso de bondad, de tal forma que a mayor valor, mejor calidad tiene la partícula. Al igual que se hacía en los filtros de partículas, dicho valor

es actualizado en cada iteración en base a la similitud del descriptor actual de la partícula, con respecto al descriptor original de la misma, de tal forma que contra más varíe la partícula con respecto a cómo era originalmente, más bajo valor de peso tendrá.

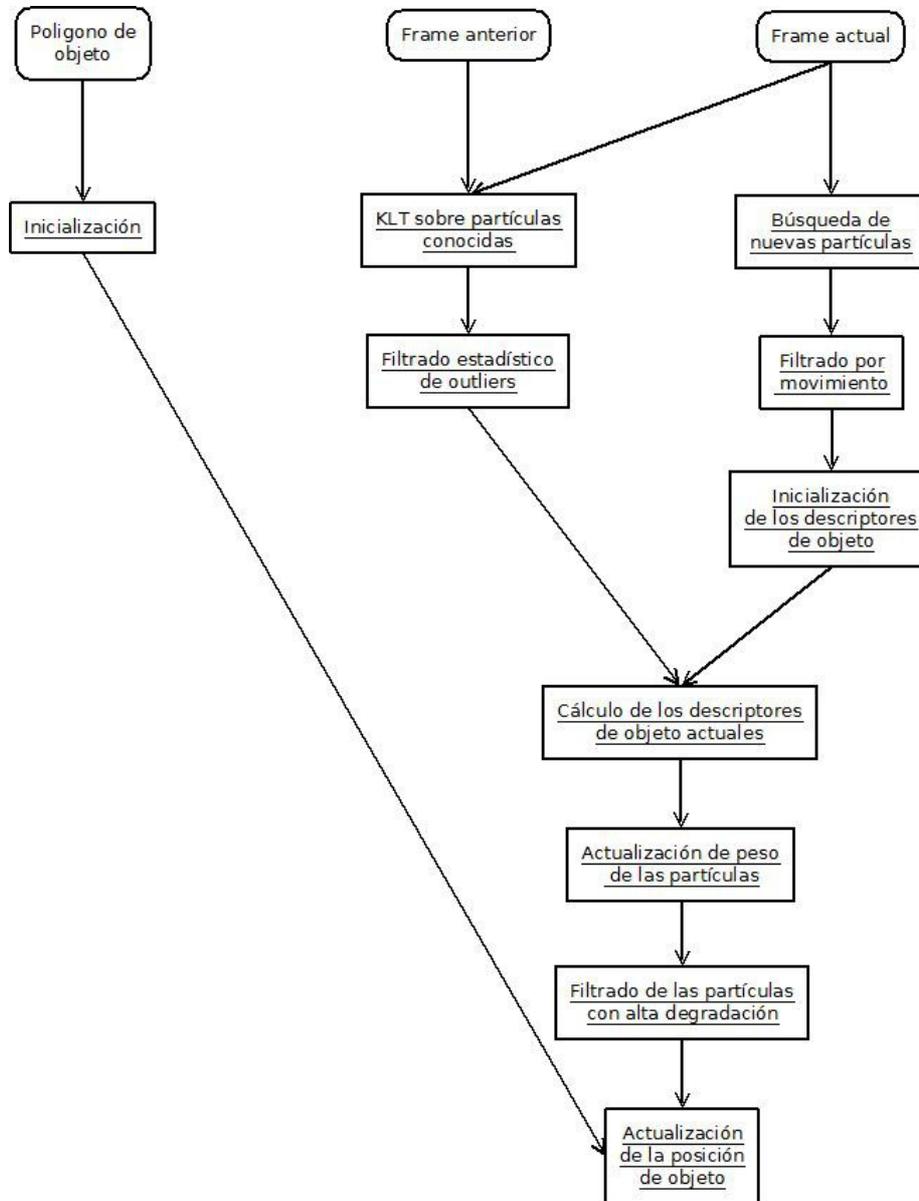


Figura 7.12: Diseño del sistema de seguimiento PKLT

En lo referente a las métricas de similitud utilizadas, estas se definieron de la siguiente manera:

- Métrica de movimiento: $F(V) = \sqrt{Vx * Vx + Vy * Vy}$, donde V es el vector de movimiento actual de la partícula, y (Vx, Vy) son sus componentes.
- Métrica de color: $F(\nabla C) = \frac{1}{1 + \nabla Cr + \nabla Cg + \nabla Cb}$, donde ∇C es la diferencia de color entre el descriptor original y el actual, y donde $(\nabla Cr, \nabla Cg, \nabla Cb)$ son sus componentes.

Por otro lado los pesos son actualizados secuencialmente mediante las siguientes reglas de aprendizaje:

- $W_i = (1 - ALFA_M) * W_i + ALFA_M * F(V)$, donde W_i es el peso de la partícula i-esima, y $ALFA_M$ es el factor de aprendizaje definido para el movimiento, el cual para nuestros experimentos era de 0,001.
- $W_i = (1 - ALFA_C) * W_i + ALFA_C * F(\nabla C)$, donde W_i es el peso de la partícula i-esima, y $ALFA_C$ es el factor de aprendizaje definido para el color, el cual para nuestros experimentos era de 0,1.

Finalmente y una vez actualizados los pesos de todas las partículas, éstas eran filtradas mediante un umbral calculado empíricamente, el cual para nuestros experimentos fue de 0.2.

7.3.3 Impresiones

Cuando se diseñó este algoritmo de seguimiento se hizo con una idea doble:

- Primero, empezar a combinar distintos tipos de características de forma más o menos eficiente, simple y escalable.
- Segundo, comenzar a definir un framework que posteriormente permitiese un aprendizaje robusto de alguna forma.

El caso es que el filtrado de partículas o características propuesto no sólo permitió ambas cosas, sino que además, provocó un salto cualitativo en la calidad del seguimiento automático de manera notable. Y ello se hizo patente sólo con ver la selección de características que se realizaba, ya que la mayor virtud de este algoritmo era la de saber escoger con alta robustez y fiabilidad las características de objeto más útiles, desechando el resto en base a su degeneración o poca coherencia con las demás.

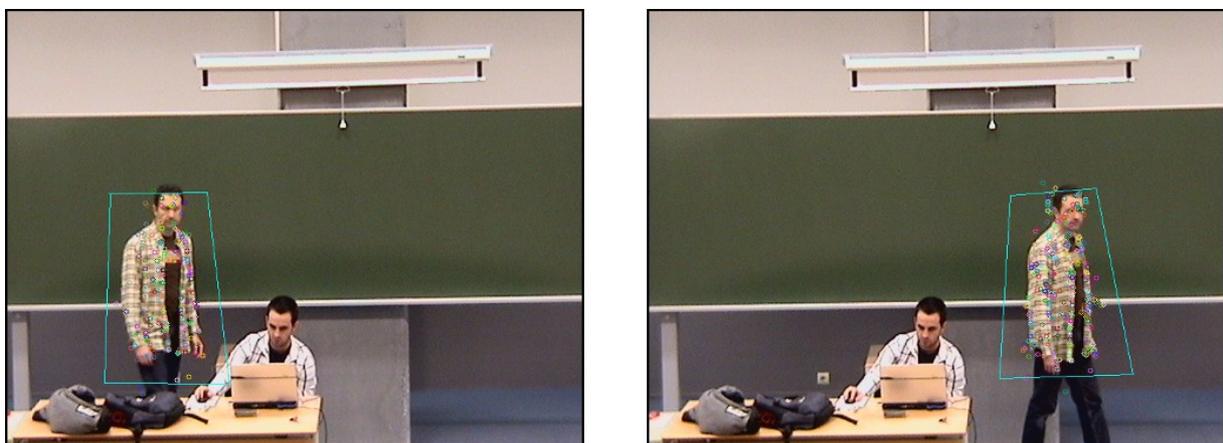


Figura 7.13: Mantenimiento de coherencia en características tras una oclusión parcial.

En este caso, los “outliers” no sólo eran filtrados mucho más rápidamente, sino que la mayoría no llegaban ni a serlo, ya que en cuanto las características degeneraban lo suficiente, éstas eran eliminadas en pos de ser sustituidas por unas nuevas inmediatamente. En la Figura 7.14 se puede ver una muestra de ello, donde durante un giro completo del objeto las características o partículas tienden a agruparse de repente en el borde a causa de dicha rotación, para inmediatamente después ser filtradas por degeneración y sustituidas por las nuevas características detectadas en las nuevas partes visibles del objeto.

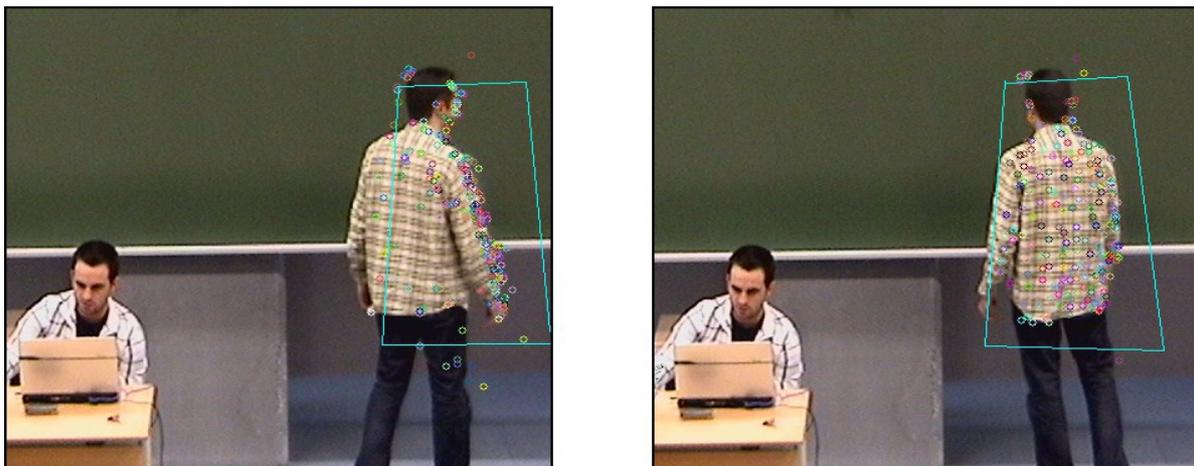


Figura 7.14: Filtrado y renovación de características durante una rotación.

No obstante, esto en principio no solucionaba uno de los problemas que se deseaban corregir, y que era el de poder realizar un aprendizaje robusto que permitiese modelar el objeto minimizando el efecto de los distintos tipos de envenenamiento. Pero esto no es del todo cierto, ya que de por sí (y como se verá en el siguiente apartado), la selección de características realizada puede considerarse un representante robusto de modelo de objeto, en el cual y gracias al tipo de filtrado realizado, se minimiza cualquier tipo de envenenamiento o corrupción que pudiera haber.

7.4 Seguimiento mediante PKLT y Mean-Shift (PKLTGA).

7.4.1 Mean-Shift

Otra de las técnicas más recurridas al ámbito del seguimiento de objetos en secuencias de vídeo, al menos en la literatura, ha sido Mean-Shift. Este más que famoso método de búsqueda mediante ascenso por gradiente (demostrado en [44]), fue introducido por Comaniciu et al. en el campo del seguimiento de objetos en [37], y desarrollado en dicho ámbito hasta el día de hoy en numerosas publicaciones (véase por ejemplo [37-43]).

OpenCv proporciona ya diversas herramientas para hacer seguimiento a través de Mean-Shift, aunque como los resultados iniciales no eran lo suficientemente alentadores, se realizó una

implementación propia que pudiera ser modificada según las necesidades. La implementación de Mean-Shift realizada es relativamente simple y se encuentra explicada con todo detalle en [37]. En la Figura 7.15 se puede ver el ejemplo gráfico intuitivo detrás del algoritmo Mean-Shift, donde el círculo azul representa la posición y límites de la ventana, y cuyo vector de gradiente en cuanto a densidad apunta al centro de masas de la ventana, en dirección a un máximo local (el cual es el objetivo perseguido).

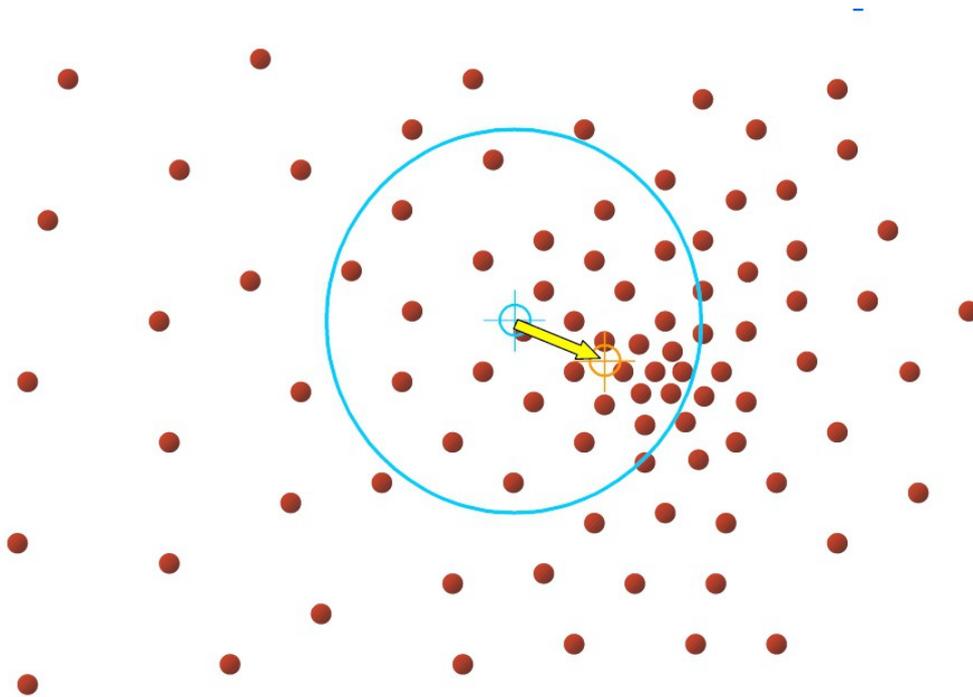


Figura 7.15: Idea intuitiva del algoritmo Mean-Shift.

Originalmente el algoritmo de seguimiento por Mean-Shift puede trabajar con cualquier tipo de descriptor, aunque en nuestro caso nos centramos exclusivamente en el color y parcialmente en el gradiente de luminancia. De hecho para que el algoritmo mostrase buenos resultados, tras múltiples intentos se terminó por definir como descriptor un histograma unidimensional, a partir de aquí denominado como histograma RGBE, compuesto por:

- Valores cuantificados y concatenados relativos a cuatro dimensiones: rojo, verde, azul, e indicativo de borde.
 - Donde los valores de color son cuantificados al rango de $[0, 16]$.
 - Donde el valor de indicativo de borde es discreto: 0 o 1.

De esta forma, el número de elementos final que formaban el histograma era de: $16 \times 16 \times 16 \times 2 = 8192$.

Todo esto dejó más que patente la importancia de escoger correctamente el descriptor a utilizar, ya que dependiendo de cómo se definiera éste, los resultados podían ser muy malos o muy buenos.

7.4.2 Diseño

Tras los prometedores resultados del sistema de seguimiento PKLT y ya con un conjunto de características robusto y fiable, lo que se pretendía a continuación era abordar finalmente el tema referente al aprendizaje y explotación del modelo de objeto.

Como evolución de todo lo implementado anteriormente, para este algoritmo de diseño un sistema de seguimiento en dos fases:

- Una primer seguimiento mediante el uso del método PKLT, lo cual nos permitía:
 - Seguir cualquier movimiento rápido de forma inmediata.
 - Seleccionar las características de objeto de forma fiable y robusta (por coherencia de color y movimiento).
- Una segunda fase de refinamiento mediante ascenso por gradiente que nos sitúe en el punto más correcto respecto a nuestro modelo de objeto, mediante:
 - Las características de objeto proporcionadas por PKLT para el aprendizaje del modelo.
 - Mean-Shift para el ascenso por gradiente.

Para la primera fase se utilizó íntegramente el algoritmo PKLT descrito en 7.3, mientras que para la segunda fase se hizo uso del algoritmo de ascenso por gradiente expuesto en el apartado anterior, donde el modelo de objeto fue representado mediante el descriptor detallado también en el apartado anterior (un histograma de 8192 elementos con la información del color y del gradiente).

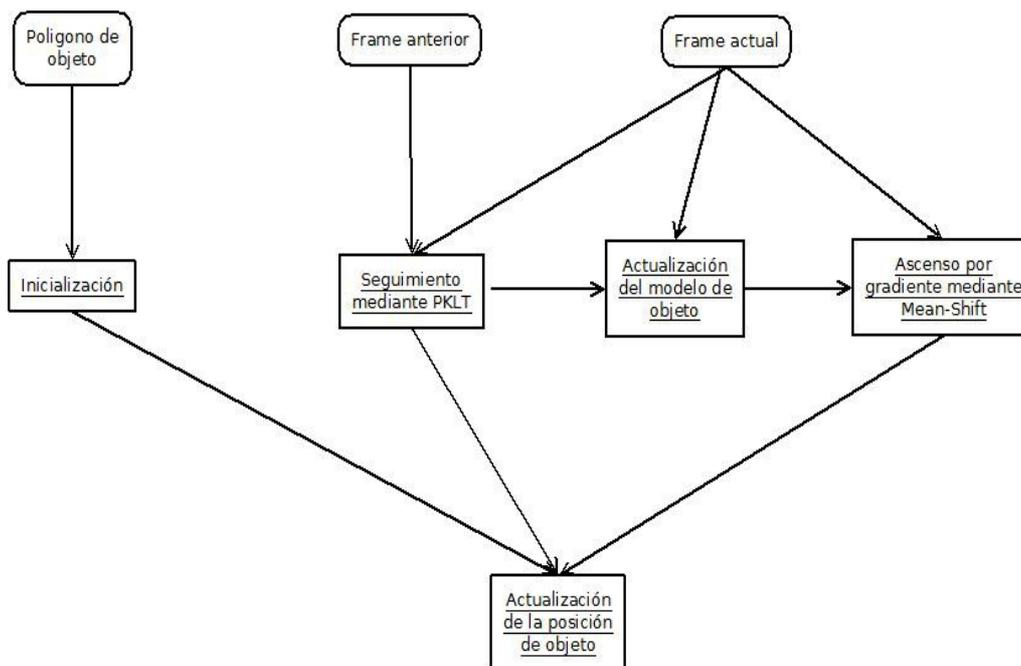


Figura 7.16: Diseño del algoritmo PKLTGA

Por otro lado y para la generación del modelo de objeto, se utilizaron ventanas de tamaño 11 centradas en cada una de las características (al igual que las usadas en PKLT), donde para cada pixel se incrementaba el valor del histograma correspondiente a su índice RGBE en 1. De esta forma, el modelo RGBE del objeto era compuesto de forma incremental por todas las características del mismo a lo largo del tiempo, permitiendo así el auto-aprendizaje. En otras variantes de este algoritmo y con resultados similares, se substituyó el aprendizaje incremental por la siguiente regla de aprendizaje: $H = (1 - ALFA_H) * H + ALFA_H * H'$. Donde H es el histograma que modela al objeto, H' es histograma del conjunto de características del actual intervalo de tiempo, y $ALFA_H$ es el factor de aprendizaje.

7.4.3 Impresiones

Tal y como se esperaba en su diseño, con este algoritmo de seguimiento se suplía la carencia de la falta de aprendizaje del objeto, permitiendo entre otras cosas seguir correctamente al objeto en la mayoría de los casos ante un cruce o cuando algún objeto en movimiento pasase por delante, siempre y cuando se hubiera podido aprender con suficiencia el modelo de objeto.

Además y gracias al uso de las PKLT a la hora de filtrar las características de objeto, el aprendizaje no degeneraba a largo plazo tal y como ocurre en muchos sistemas de este tipo, ya que la información utilizada para el aprendizaje estaba lo suficientemente libre de contaminación.



Figura 7.17: Seguimiento del objeto durante una sustancial oclusión parcial.

Llegados a este punto, hay pocas cosas que se le pudiese echar en falta al sistema de seguimiento en este caso en concreto, a excepción de añadir más información y complejidad al modelado de objeto, ya que a fin de cuentas la implementación realizada del mismo era relativamente simple.

Cabe destacar la importancia de seleccionar correctamente el umbral de filtrado por degeneración en algunos tipos de escenas, ya que en los casos de mucho movimiento de fondo, poco tiempo para el aprendizaje del objeto, y cruces casi totales con otros objetos (véase Figura 7.17), es necesario subir el umbral casi al máximo, lo que a efectos prácticos provoca la independencia de la detección

y seguimiento de características, respecto al tiempo. En estos casos, un umbral al máximo sería similar a detectar y luego a desechar todas las características al final de cada ciclo.

No obstante y como reflexión de toda la experiencia anterior, se extrajeron las siguientes conclusiones:

- La detección de características y el seguimiento de las mismas es un método robusto para identificar partes de objeto a través del movimiento.
- Con dichas partes de objeto se puede extraer todo tipo de información y modelar el objeto.
- El modelado de objeto tiene que ser robusto y tener la mínima corrupción posible.
 - Si las características pueden degenerar, entonces se necesita un más que buen sistema de filtrado de las mismas.
- La elección del algoritmo de seguimiento final no tiene tanto impacto como cuan de bueno sea el modelo de objeto.
- Más allá y para escenarios más complejos, es necesario tener conocimiento del resto de la escena, ya que sin información suficiente cualquier decisión es arbitraria:
 - Tener un modelado también de lo que pueda ser considerado como fondo estático.
 - Tener un modelado también del resto de objetos dentro de la escena.
 - Con dicha información se podrían generar los modelos de objeto de tal forma que no sólo se maximizase la propia similitud del objeto, sino también su disimilitud con respecto al resto de elementos conocidos.

8 Sistema de producción automática

La intención inicial era la de abordar de una forma más exhaustiva y desde un prisma científico mucho más interesante la parte de producción automática, pero desgraciadamente y debido a los recursos dedicados al sistema de seguimiento, no se le pudo dedicar a este sistema el esfuerzo que se hubiera querido.

Por ello y desde una perspectiva más simple, se definieron un conjunto de reglas fijas para la parte de producción automática de tal forma que se cumpliera el mínimo de la función estética requerida, tanto para las traslaciones como para las funciones de “Zoom”.

En cuanto a la lógica de control para el movimiento de la cámara móvil, se tuvieron que realizar dos implementaciones distintas en base al origen de vídeo utilizado para el análisis.

- Un módulo de control para cuando el origen del vídeo proviene de la propia cámara móvil.
- Un módulo de control para cuando el origen de vídeo proviene de la cámara fija, donde entre otras cosas se han de poner en correspondencia los sistemas de coordenadas de ambas cámaras.

8.1 Lógica de control en cámara móvil

Este módulo de control fue implementado para cuando el origen del vídeo analizado fuera el de la propia cámara móvil.

Teniendo en cuenta esto, las coordenadas devueltas por el algoritmo de seguimiento eran las propias coordenadas de imagen de la cámara móvil. Esto hacía que gracias al método de control relativo facilitado por la propia cámara PTZ, el cual fue detallado en el apartado 5.2 de este documento, el proceso de control fuera prácticamente directo.

No obstante, y por temas de optimización y prevención se tuvieron que definir un conjunto de restricciones:

- No realizar más de un movimiento de cámara por cada intervalo de 10 imágenes. Lo cual no sólo era innecesario a todos los efectos en lo referente a la cuantía de movimiento de los objetos en dicho intervalo de tiempo, sino que en caso contrario se saturaba el servicio de la cámara.
- No mover la cámara hasta que el movimiento del objeto fuera lo suficiente significativo. Lo cual se hacía necesario para evitar molestas vibraciones continuas causadas por la precisión del seguimiento. En nuestro caso se estableció un umbral de movimiento mínimo del 5% sobre el alto, o sobre el ancho de la imagen.
- Por otro lado y para suavizar el movimiento de la cámara a la hora de trasladarse horizontal

o verticalmente, la cantidad del movimiento a realizar por la cámara se definió de forma proporcional al movimiento realizado por el objeto (en nuestros experimentos de 0.5). De esta forma cuanto más lejos esté el objeto, más rápido se moverá la cámara, decrecentando su velocidad según se vaya acercando al centro objeto.

8.2 Lógica de control en la cámara fija (triangulación)

Este módulo de control fue implementado para cuando el origen del vídeo analizado fuera el de la cámara fija.

Teniendo en cuenta esto, se hace necesario un sistema de proyección que nos permita proyectar las coordenadas de imagen correspondientes a la cámara estática, sobre una de dos: las coordenadas de imagen de la cámara móvil, o directamente sobre las coordenadas de los motores de la cámara PTZ.

La primera opción requeriría mucha calibración y saltos entre los distintos espacios (imagen fija → espacio global cámara móvil → imagen móvil actual), mientras que la segunda opción puede realizarse directamente a través de una homografía que relacione ambos espacios (detallada en el apartado 6.4 de este documento). Ambos espacios son compatibles ya que se ven sujetos al mismo tipo de proyección perspectiva, en el caso de la cámara en su centro de proyección, y en el caso de los motores de la PTZ en su centro de giro (véase trabajo de esta misma escuela en [45]).

Para ello se diseñó un sistema de calibración donde:

- En caso de no haberse realizado todavía la calibración, se le iniciaba el asistente.
 - Primero se le presenta al usuario la imagen correspondiente a la cámara fija para que marque al menos 4 puntos de calibración.
 - Como segundo paso se le muestra al usuario la imagen correspondiente a la cámara móvil para que marque las correspondencias puntuales en el mismo orden.
 - Posteriormente se sitúa la cámara móvil en dichas posiciones mediante el sistema de control relativo (véase apartado 5.2), solicitando en el proceso las correspondientes coordenadas absolutas de la cámara mediante el servicio de la misma proporcionado a tal efecto (detallado en el manual de SONY en [17]).
 - Finalmente, con las coordenadas de imagen y con las coordenadas absolutas de la cámara, se calcula la homografía que pone en correspondencia ambas vistas.

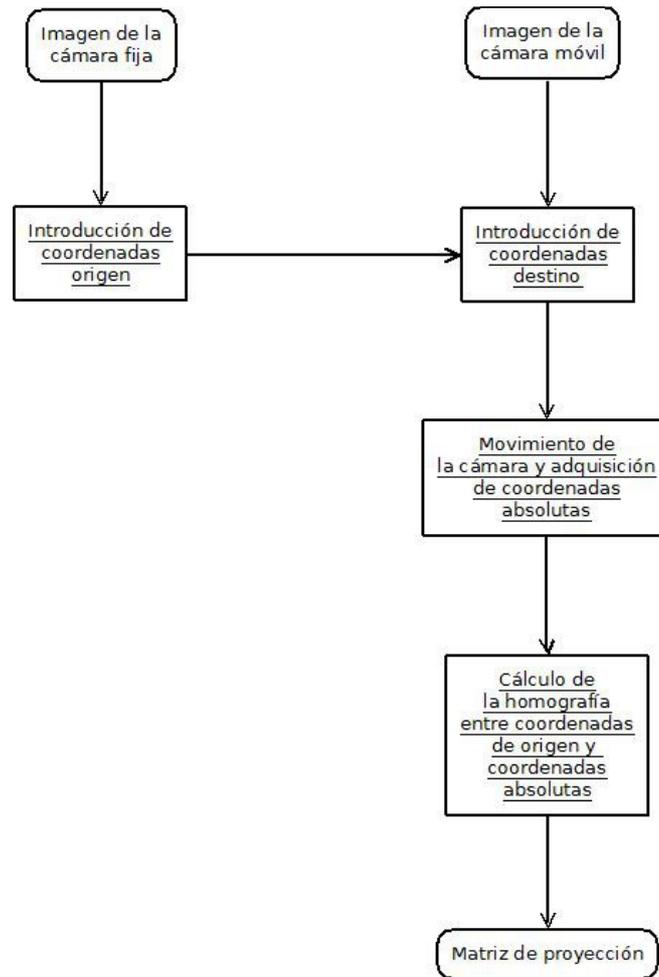


Figura 8.1: Diseño del sistema de calibración de la cámara estática.

Una vez ya con la matriz homográfica calculada, y bajo las mismas restricciones de movimiento que en el sistema de control de la cámara móvil (detallado en el apartado anterior), se hace la proyección de las coordenadas de imagen al espacio absoluto de la cámara PTZ, para finalmente moverse mediante el sistema de control absoluto proporcionado por la misma a través del módulo detallado en el apartado 5.1 de este mismo documento.

8.3 Reglas de producción

Como ya se comentó en la introducción de este punto, y a causa de falta de recursos para el desarrollo adecuado del sistema de producción, se tuvo que diseñar un sistema relativamente simple de producción automática basado en reglas estáticas para cubrir las necesidades mínimas necesarias para el presente proyecto.

Para ello se definieron principalmente tres reglas:

- En caso de permanecer el objeto quieto durante un intervalo de tiempo determinado,

entonces se realiza “Zoom In”. El “Zoom In” es realizado de forma proporcional a una distancia máxima de “Zoom In” definida en el fichero de configuración. De esta forma contra más lejos se estuviese del “Zoom In” máximo, más rápido se realizaría la ampliación, mientras que por otro lado, según se vaya decrementando dicha distancia, la velocidad del “Zoom In” irá disminuyendo.

- En caso de empezar a moverse el objeto un espacio mayor a un umbral configurado como parámetro en el fichero de configuración, entonces se realiza “Zoom Out” proporcionalmente a la cuantía de movimiento realizada por el objeto, y así hasta el “Zoom Out” mínimo.
- La tercera regla ya ha sido introducida en el apartado 8.1, y se basa en suavizar el movimiento de la cámara mediante la realización de cada traslación de forma proporcional al movimiento realizado por el objeto. De esta forma cuanto más lejos esté el objeto, más rápido se moverá la cámara, decrementando su velocidad según se vaya acercando al centro del mismo.

9 Evaluación y Resultados

En esta sección evaluaremos de forma objetiva los resultados de los sistemas de seguimiento implementados a lo largo del proyecto. No obstante, y a causa de la ausencia de bases de datos etiquetadas que evalúen los específicos requisitos de nuestra aplicación, se harán uso de las secuencias utilizadas en [47] para la evaluación de los distintos algoritmos de seguimiento de objetos en el estado del arte.

Hay que tener muy presente, que nuestros algoritmos de seguimiento fueron diseñados para un aprendizaje progresivo de los objetos, con el objetivo de poder realizar así seguimientos de los mismos a largo plazo y en escenas poco complejas. Por contra, las secuencias públicas utilizadas en este apartado fueron definidas para la evaluación de algoritmos de seguimiento de objetos, a muy “corto plazo” y según las distintas complejidades de escena, sin atender en ningún momento a sus capacidades adaptativas, ni a su rendimiento a largo plazo. Todo esto provoca que dichas secuencias sean extremadamente cortas, impidiendo en la mayoría de los casos cualquier tipo de aprendizaje, ya que si por ejemplo otro objeto distinto se cruza nada más iniciar la secuencia cuando el objeto a seguir todavía no ha podido ser aprendido, de nada le servirá al algoritmo su capacidad de aprendizaje.

Como métrica de evaluación se hará uso de la SFDA, tal y como se describió y utilizó en [47]. Dicha métrica es una medida de la proporción del solapamiento respecto a la exclusión, donde el solapamiento y la exclusión son definidos sobre los “bounding boxes” de la posición real y la estimación realizada del objeto a lo largo de toda la secuencia.

Según el análisis realizado en [47], valores superiores a 0.6 indican un seguimiento prácticamente perfecto. No obstante, en nuestros experimentos no se perdió nunca al objeto para valores superiores a 0.5, mientras que valores entre 0.4 y 0.5 fueron indicativo de un buen seguimiento pero con poca precisión, provocando en algunos casos una pérdida del objeto casi al final de la secuencia. En cuanto a valores por debajo de 0.3 indicaban claramente la pérdida absoluta del objeto en algún punto de la secuencia.

Las secuencias utilizadas para la evaluación fueron seleccionadas de los conjuntos de vídeos pertenecientes a escenas reales complejas, de los niveles 3 y 4 de las bases de datos utilizadas en [47]. En cuanto a los algoritmos evaluados han sido los 9 más representativos en cuanto a resultados, los cuales son descritos en la [Tabla 9.1].

Algoritmo	Descripción
KLT	Algoritmo de seguimiento KLT descrito en el apartado 7.1.
PF Motion	Algoritmo de seguimiento de filtros de partículas de movimiento descrito en el apartado 7.2.
KLT ND	Modificación del algoritmo KLT sin memoria, donde en cada intervalo de tiempo las características son eliminadas y vueltas a detectar de nuevo desde cero.
PF Color	Algoritmo de seguimiento de filtros de partículas sobre el histograma de color del objeto.
PKLT	Algoritmo PKLT descrito en el apartado 7.3.
PKLTGA	Primera versión del algoritmo PKLTGA donde para el aprendizaje del modelo se utilizaba la plantilla del objeto correspondiente al intervalo de tiempo anterior.
PKLTGA Learning	Algoritmo PKLTGA descrito en el apartado 7.4
PKLTGA Learning 2	Modificación del algoritmo PKLTGA donde se hace uso tanto de las partículas como de la plantilla del intervalo anterior para el aprendizaje.
PKLT Filter	Simplificación del modelo PKLTGA donde las partículas son detectadas de forma independiente respecto al tiempo (lo que es equivalente a un PKLTGA con umbral de filtrado al máximo). En este algoritmo se pretendía utilizar también las características correspondientes al fondo de la escena para pesar las partículas con respecto a su disimilitud con dicho modelo de fondo, pero desgraciadamente y por falta de tiempo dicha funcionalidad no pudo ser implementada.

Tabla 9.1: Descripción de los algoritmos evaluados.

En cuanto a los resultados obtenidos, tal y como se puede ver en las [Tablas 2-6], el algoritmo con mejores resultados a nivel global es el PKLT Filter, el cual es la evolución final y simplificada de los modelos PKLTGA, pero con independencia temporal a la hora de detectar las características. El único conjunto de secuencias donde el PKLT Filter no presenta la mejor opción es el de secuencias de casos reales de coches de nivel 4 ([Tabla 2]), siendo la razón de ello la enorme y abrupta variación en el histograma de los objetos a seguir durante la primera mitad de las secuencias, y es precisamente por ello que en dichos vídeos la mejor opción es el algoritmo de seguimiento por movimiento más evolucionado y que todavía no realiza el modelado de objeto a base del color, el algoritmo PKLT. No obstante, los aparentes peores resultados de los algoritmos PKLTGA con respecto al PKLT Filter, vienen dados principalmente por la falta de tiempo en estos primeros de recoger suficientes características a la hora de aprender el objeto (no olvidemos que son secuencias de alta complejidad y muy cortas), mientras que en el caso del PKLT Filter, la independencia temporal del mismo a la hora de detectar características le permite al algoritmo realizar un muestreo mucho más rápido (lo que no quiere decir necesariamente mejor), y por lo tanto aprender la información del objeto más rápidamente.

Secuencia	KLT	PF Motion	KLT ND	PF Color	PKLT	PKLTGA	PKLTGA Learning	PKLTGA Learning 2	PKLT Filter
mv2_020_red	.259	.294	.062	.107	.556	.335	.175	.179	.310
mv2_020_silver	.387	.214	.036	.022	.679	.391	.168	.466	.245
mv2_020_whiteban	.115	.352	.042	.439	.496	.440	.109	.478	.156

Tabla 9.2: Resultados sobre las secuencias de nivel 4 de: coches, casos reales, alta complejidad.

Secuencia	KLT	PF Motion	KLT ND	PF Color	PKLT	PKLTGA	PKLTGA Learning	PKLTGA Learning 2	PKLT Filter
PETS2009_S2_L1_view001_1	.179	.163	.096	.197	.527	.330	.390	.270	.452
PETS2009_S2_L2_view001_1	.075	.085	.031	.706	.073	.088	.068	.081	.516
PETS2009_S2_L3_view001_2	.055	.122	.077	.021	.276	.195	.327	.321	.530

Tabla 9.3: Resultados sobre las secuencias de nivel 4 de: personas, casos reales, alta complejidad.

Secuencia	KLT	PF Motion	KLT ND	PF Color	PKLT	PKLTGA	PKLTGA Learning	PKLTGA Learning 2	PKLT Filter
HEADTRACK_seq_ms	.430	.257	.194	.133	.373	.726	.200	.631	.662
HEADTRACK_seq_sb	.285	.242	.006	.075	.283	.603	.286	.497	.626
HEADTRACK_seq_villains2	.093	.211	.021	.178	.155	.359	.201	.466	.621

Tabla 9.4: Resultados sobre las secuencias de nivel 3 de: caras, alta complejidad.

Secuencia	KLT	PF Motion	KLT ND	PF Color	PKLT	PKLTGA	PKLTGA Learning	PKLTGA Learning 2	PKLT Filter
mv2_002_redcar	.320	.378	.330	.403	.531	.401	.610	.337	.467
mv2_005_silvercar	.313	.380	.646	.029	.530	.509	.512	.468	.517
mv2_003_blackcar	.227	.080	.670	.443	.396	.409	.349	.357	.446

Tabla 9.5: Resultados sobre secuencias de nivel 3 de: coches, oclusión, alta complejidad.

Secuencia	KLT	PF Motion	KLT ND	PF Color	PKLT	PKLTGA	PKLTGA Learning	PKLTGA Learning 2	PKLT Filter
l3_occlusion_people_1	.288	.377	.111	.303	.301	.353	.518	.482	.491
l3_occlusion_people_2	.058	.202	.079	.234	.198	.285	.198	.248	.619

Tabla 9.6: Resultados sobre secuencias de nivel 3 de: personas, oclusiones, alta complejidad.

Por otro lado, si comparamos los resultados obtenidos con los resultados mostrados en [47] sobre el estado del arte actual, en principio parece haber una clara y sustancial mejoría en la mayoría de las secuencias evaluadas. No obstante, aquí tan sólo se ha evaluado un pequeño conjunto de las mismas, y aunque las secuencias seleccionadas sean todas ellas de los niveles de mayor complejidad, sería todavía necesario un proceso de evaluación bastante más exhaustivo antes poder validar cualquier afirmación de ese estilo.

Independientemente de los prometedores resultados anteriores, tal y como se introdujo al principio de este apartado, las secuencias utilizadas sólo permiten la evaluación a corto plazo de los distintos algoritmos de seguimiento, lo que provoca que la mayoría de aquellos algoritmos que muestran buenos resultados en las mismas, fallen luego a la hora de seguir a un objeto durante un intervalo de tiempo ligeramente mayor. Por ello, y teniendo en cuenta el coste de etiquetar secuencias de pocas decenas de segundos, el etiquetar secuencias de decenas de minutos queda obviamente fuera del objetivo del proyecto. A causa de lo cual, y tan sólo a nivel de ejemplo cualitativo, se mostrará el comportamiento de los principales algoritmos implementados sobre una situación real dentro del campo de aplicación para el que fueron diseñados.

El vídeo utilizado pertenece a una grabación de una clase simulada por uno de los profesores del departamento, de 5 minutos de duración, y dentro del aula donde fue montado el piloto. En la [Tabla 9.7] se muestran los resultados de las simulaciones, donde se indica tanto el número de veces que se perdió al objetivo, como si fue posible la recuperación del mismo en todas ellas.

Algoritmo	Número de veces perdido el objeto	Recuperación del objeto todas las veces
KLT	4	No
PF Motion	5	Si
KLT ND	1	No
PF Color	1	No
PKLT	1	Si
PKLTGA	0	-
PKLTGA Learning	0	-
PKLTGA Learning 2	0	-
PKLT Filter	0	-

Tabla 9.7: Resultados del seguimiento sobre una secuencia real de 5 minutos de duración.

Tal y como se puede ver en los resultados de la [Tabla 9.7], los algoritmos diseñados se comportan a la perfección para la función para la que fueron diseñados, teniéndose resultados de seguimiento perfectos a partir del primer algoritmo donde se incluyó el modelado de objeto.

10 Conclusiones y Trabajo Futuro

En este trabajo se ha desarrollado un “framework” completo para la realización del seguimiento y producción automática de objetos mediante el uso de cámaras móviles en entornos de red, habiéndose aplicado con éxito al problema concreto del seguimiento automático de profesores durante la emisión virtual de clases presenciales.

No obstante, la principal aportación de esta investigación son los distintos métodos de seguimiento y aprendizaje automático de objetos a largo plazo que se han propuesto, desarrollado y evolucionado a lo largo de todo el trabajo. Por otro lado y gracias al sistema de estabilización de vídeo implementado, todos estos métodos de seguimiento desarrollados son capaces de funcionar también en cámaras móviles.

Como consecuencia de todo ello, no sólo se ha cumplido más allá de las expectativas con el principal objetivo de este trabajo (el de realizar el seguimiento y producción automática del profesor para la emisión virtual de las clases), sino que también y en base a los resultados obtenidos en el apartado 9, los algoritmos implementados parecen mostrar un rendimiento notoriamente superior al del actual estado del arte, con el añadido además de haber demostrado su más que notable capacidad de realizar seguimientos a largo plazo.

Desgraciadamente y a pesar de los buenos resultados obtenidos, a causa de una clara falta de recursos, el desarrollo del sistema de seguimiento se quedó parcialmente a medias con respecto a las intenciones iniciales, dejando todavía mucho trabajo futuro por hacer:

- Añadir información de forma al modelado de objeto.
- Tener en cuenta el fondo a la hora de realizar el seguimiento.
- Tener en cuenta los otros objetos en la escena también a la hora de realizar el seguimiento.
- Añadir a la métrica de evaluación de la similitud del objeto, la disimilitud tanto con el fondo como con el resto de objetos.
- Evaluar la opción de realizar un modelado por partes del objeto en vez de un modelado global, con todo lo que ello conlleva.
- Etc.

Referencias

1. P.D.Z. Varcheie, G.A. Bilodeau: Active people tracking by a PTZ camera in IP surveillance system. In proceeding of Robotic and Sensors Environments, pp. 98-103, 2009.
2. P.D.Z. Varcheie, G.A. Bilodeau: Human Tracking by IP PTZ Camera Control in the Context of Video Surveillance. In Image Analysis and Recognition, 6th International Conference, ICIAR 2009, Halifax, Canada, July 6-8, 2009.
3. P.D.Z. Varcheie, G.A. Bilodeau: Adaptive Fuzzy Particle Filter Tracker for a PTZ Camera in an IP Surveillance System. In IEEE Transactions on instrumentation and measurement, Vol. 60, N. 2, February 2011.
4. R. Cucchiara, A. Prati, R. Vezzani: Advanced video surveillance with pan tilt zoom cameras. In Proceedings of Workshop on Visual Surveillance (VS) at ECCV, 2006.
5. C. Guillot, M. Taron, P. Sayd, Q.C. Pham, C. Tilmant, J.M. Lavest: Background subtraction adapted to ptz cameras by keypoint density estimation. In BMVC, Vol. 34, pp. 1–10, 2010.
6. L. Robinault, S. Bres, S. Miguet: Real time foreground object detection using ptz camera. In VISSAPP Vol. 1, N. 9, pp. 609–614, 2009.
7. I.H. Chen, S.J. Wang: Efficient Vision-Based Calibration for Visual Surveillance Systems with Multiple PTZ Cameras. In IEEE International Conference on Computer Vision Systems, ICVS 06, 2006.
8. R. Yi Da Xu, J. Gao, M. Antolovich: Novel methods for high-resolution facial image capture using calibrated PTZ and static cameras. In Proceedings of the 2008 IEEE International Conference on Multimedia and Expo, ICME 2008.
9. J.M Gamo: Emision de clases presenciales. PFC in UAM, 2012.
10. Y. Xie, M. Pei, G. Yu, X. Song, Y. Jia: Tracking pedestrians with incremental learned intensity and contour templates for PTZ camera visual surveillance. In Multimedia and Expo (ICME), IEEE International Conference, 2011.
11. Z. Wu, R. J. Radke: Using scene features to improve wide-area video surveillance. In CVPR Workshops, pp 50-57, 2012.
12. F. Chang, G. Zhang, X. Wang, Z. Chen: PTZ Camera Target Tracking in Large Complex Scenes. In Proceedings of the 8th World Congress on Intelligent Control and Automation, Jinan, China, July 2010.
13. T. Dinh, Y. Qian, G. Medioni: Real time tracking using an active pan-tilt-zoom network camera. In Intelligent Robots and Systems, 2009.
14. C.S. Yang, R.H. Chen, C.Y. Lee, S.J. Lin: PTZ camera based position tracking in IP surveillance system. In: Inelligence. Conference on Sensing Technology, pp. 142–146 (2008).
15. Y. Xie, L. Lin, Y. Jia: Tracking Objects with Adaptive Feature Patches for PTZ Camera Visual Surveillance. In Proceeding of 20th International Conference on Pattern Recognition, pp.1739-1742, ICPR 2010, Istanbul, Turkey, 23-26 August 2010.
16. S. Fazli, H. Moradipour, H. Bouzari: Particle Filter based object tracking with sift and color feature. In ICMV Conference, UAE- Dubai, 28 - 30 December 2009.
17. Sony: Sony Video Network Station SNT-EP104/EP154/EX101E/EX104/EX154 CGI Command Manual. 5th Generation, v 2.03, January 2012.
18. C. Harris and M. Stephens: A Combined Corner and Edge Detector. In Proceeding Alvey Vision Conference, Univ. Manchester, pp. 147-151, 1988.
19. J. Shi and C. Tomasi: Good Features to Track. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 593-600, June 1994.

20. E. Rosten, T. Drummond: Machine learning for high-speed corner detection. In European Conference on Computer Vision, pp 430-443, May 2006.
21. E. Rublee, V. Rabaud, K. Konolige, G. R. Bradski: "Orb: An efficient alternative to sift or surf". In ICCV, 2011.
22. B. D. Lucas, T. Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision. In International point Conference on Artificial Intelligence, pages 674-679, 1981.
23. C. Tomasi, T. Kanade: Detection and Tracking of Point Features. In Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.
24. J. Bouguet: Pyramidal implementation of the lucas kanade feature tracker. In Intel Corporation Microprocessor Research Labs, 2000.
25. K. Ki-Sang, J. Dae-Sik: Real time face tracking with pyramidal lucas-kanade feature tracker. In Computational science and its applications, pp 1074-1082, ICCSA, 2007.
26. A. S. Mian: Realtime Visual Tracking of Aircrafts. In Proceedings of the 2008 Digital Image Computing: Techniques and Applications, pp 351-356, 2008.
27. V. Chandola, A. Banerjee, V. Kumar: "Anomaly detection: A survey". In ACM Computing Surveys, vol 41, Iss 3, July 2009.
28. M.S. Arulampalam, S. Maskell, N. Gordon, T. Clapp: A tutorial on particle filters for online non-linear/non-Gaussian Bayesian tracking. In IEEE Transactions on Signal Processing, vol 50, pp 174-188, 2002.
29. M. Isard, A. Blake: "Condensation: conditional density propagation for visual tracking". In International Journal of Computer Vision, Vol 29, pp 5-28, 1998.
30. J. Wang, W. Wei, Y. Zhou: Fractional finite time delay evolution systems and optimal controls in infinite-dimensional spaces. In Journal of Dynamical and Control Systems, Vol 17, Issue 4, pp 515-535, 2011.
31. Y. Wang, Y. LI: Multi-Agent Co-Evolutionary Particle Filter for Robust Moving Object Tracking. In Journal of Computational Information Systems, pp 2511-2519, 2010.
32. J. Li, X. Lu, L. Ding and H. Lu: Moving Target Tracking via Particle Filter Based on Colour and Contour Features. In Proceedings of 2nd International Conference on Information Engineering and Computer Science, pp 1-4, 2010.
33. J. Czyz, B. Ristic and B. Macq: A Color-based Particle Filter for Joint Detection and Tracking of Multiple Objects. In Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing. pp 217-220, 2005.
34. J. Sherrah, B. Ristic, and N. Redding: Particle filter to track multiple people for visual surveillance. In IET Computer Vision, Vol 5, n° 4, pp 192-200, 2011.
35. A. Beaugendre, H. Miyano, E. Ishidera, S. Goto: Human Tracking System for Automatic Video Surveillance with Particle Filters. In IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2010.
36. Y.J. Chai, J.Y. Park, K.J. Yoon, T.Y. Kim: Multi target tracking using multiple independent particle filters for video surveillance. In IEEE International Conference on Consumer Electronics, 2011.
37. D. Comaniciu, V. Ramesh, P. Meer: Real-time tracking of non-rigid objects using mean shift. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), Vol. 2, pp 142-149, 2000.
38. D. Comaniciu, P. Meer: "Mean shift: A robust approach toward feature space analysis". In PAMI, vol 24, pp 603-619, 2002.
39. D. Comaniciu, V. Ramesh, and P. Meer: Kernel based object tracking. PAMI, vol 25, pp 564-575, 2003.

40. J. G. Allen, R. Y. D. Xu., J. S. Jin: Object tracking using camshift algorithm and multiple quantized feature spaces. In Proceeding Pan-Sydney area workshop on visual information processing, pp 3–7, 2004.
41. R. Han, Z. Jing, G. Xiao: Probabilistic motion switch tracking method based on mean shift and double model filters. In Proc. International Symposium on Neural Networks, vol 2, pp 705–714, Nanjing, China, June 2007.
42. J. Ning, L. Zhang, D. Zhang, C. Wu: Scale and orientation adaptive mean shift tracking. Computer Vision. In IET, vol 6, pp 52–61, 2012.
43. T. Vojir, J. Noskova, J. Matas: Robust Scale-Adaptive Mean-Shift for Tracking. In Computer Science, Volume 7944, pp 652-663, 2013.
44. Y. Cheng: Mean Shift, Mode Seeking, and Clustering. In IEEE Transactions on Pattern Analysis and Machine Intelligence archive, vol 17, Iss 8, pp 790-799, 1995.
45. J. Garcia: Autocalibración y sincronización de múltiples cámaras PTZ. PFC in UAM, 2007.
46. E. Dubrofsky: Homography Estimation. Master's Essay, The University Of British Columbia, Vancouver, March, 2009.
47. M. Lozano: Evaluación de Algoritmos de Seguimiento de Objetos. PFC in UAM, 2012.