



EMPRESA DAVANTIS: SEGUIMIENTO DE PERSONAS Y VEHÍCULOS POR VIDEOVIGILANCIA INTELIGENTE

Memoria del proyecto de final de carrera correspondiente a los estudios de Ingeniería Superior de Informática presentado por Javier Jubany Garrido y dirigido por Joan Serrat Gual

Bellaterra, 16 de Junio de 2008

El firmante, Joan Serrat Gual , profesor de la Escuela
Técnica Superior de Ingeniería de la Universidad Autónoma
de Barcelona

CERTIFICA:

Que la presente memoria ha sido realizada sobre su dirección
por Javier Jubany Garrido

Bellaterra, 16 Junio de 2008

Firmado: Joan Serrat Gual

El firmante, Martí Balcells de la empresa Davantis

CERTIFICA:

Que la presente memoria ha sido realizada en la empresa sobre su supervisión mediante convenio firmado con la Universidad Autónoma de Barcelona. Así mismo, la empresa tiene conocimiento y da el visto bueno al contenido que se detalla en esta memoria.

Bellaterra, 16 Junio de 2008

Firmado: Martí Balcells

A mi familia y amigos.

Agradecimientos

Este proyecto ha sido posible, gracias a la ayuda y soporte de muchas personas: Me gustaría agradecerse, a mi familia y amigos, por su apoyo moral en los momentos más difíciles. También a la empresa Davantis, por ofrecerme la oportunidad de desarrollar el proyecto con ellos. En especial a mi tutor, Martí Balcells, por su esfuerzo y dedicación, gracias a los cuales este proyecto ha llegado a buen puerto. Agradezco también la ayuda de mi director de proyecto, Joan Serrat, por sus consejos y correcciones.

A todos ellos, muchas gracias.

Índice general

1. Introducción	1
1.1. Motivaciones	2
1.2. Objetivos	3
1.3. Estructura de la memoria	4
2. Estado del arte	5
2.1. Seguimiento de objetos	5
2.2. Modelos para la representación de objetos	6
2.3. Detección de objetos en escena	8
2.4. Modelos de tracking	10
2.5. Discusión	12
3. Análisis	15
3.1. Requisitos	15
3.2. Estudio de viabilidad	16
3.3. Planificación temporal	16
4. Método e Implementación	19
4.1. Mean Shift	19
4.2. Artículos escogidos	20
4.3. Mean Shift mediante Blobs	22
4.3.1. Mean Shift multinúcleo	22
4.3.2. Pseudocódigo	23
4.3.3. Modelo de objeto	24

4.3.4.	Vector de Mean Shift con información del background . . .	25
4.4.	Mean Shift complementado	28
4.4.1.	Rango de la solución	28
4.5.	Mean Shift mononúcleo	29
4.5.1.	Adaptación de escala	30
4.5.2.	Background	32
5.	Metodología de evaluación	33
5.1.	Problemas al evaluar un tracker	33
5.2.	Sistema de evaluación del sistema de tracking	34
5.2.1.	Evaluación manual	35
5.2.2.	Test automático de tracking	37
5.2.3.	Tests automático global	38
5.3.	Sistema de evaluación para objetos estáticos	39
6.	Resultados de evaluación	41
6.1.	Modelo de Tracking de Davantis	41
6.2.	Mean Shift multinúcleo	44
6.3.	Mean Shift complementado	46
6.4.	Comparativa	49
6.5.	Objetos estáticos	51
6.5.1.	Comparativa	52
7.	Conclusiones	55
7.1.	Valoración del método	55
7.2.	Valoración de la implementación	56
7.3.	Valoración de la planificación inicial	57
	Bibliografía	59

Índice de figuras

2.1. Modelos basados en forma	6
2.2. Detección de objetos en escena	9
2.3. Elección de correspondencias, Tracking por puntos	10
4.1. Tres iteraciones del Mean Shift	20
4.2. Frame mostrando los múltiples núcleos	22
4.3. Muestra esquemática de la búsqueda multinúcleo	23
4.4. Regiones para Histogramas	25
4.5. Problemas con la escala del objeto	31
4.6. Actualización del modelo de background	31
5.1. Salida para el test manual	35
5.2. Iluminación: Cambio brusco y escena con sombras	37
5.3. Posición: Objeto lejano, cruce y oclusión parcial	37
5.4. Objeto en posición estática	39
6.1. Porcentaje de tracking y N_o medio de id's, Daview	43
6.2. Porcentaje de tracking y N_o medio de id's, Mean Shift multinúcleo	45
6.3. Porcentaje de tracking y N_o medio de id's, Mean Shift comple-	
mentado	47
6.4. Porcentaje de tracking y N_o medio de id's, Comparativa	49
6.5. Tiempo de tracking, Comparativa	53

Índice de cuadros

5.1. Conjunto de test, manual	36
6.1. Resultados manuales, Daview	42
6.2. Medias de métricas manuales, Daview	42
6.3. Resultados manuales, Mean Shift multinúcleo	45
6.4. Medias de métricas manuales, Mean Shift multinúcleo	46
6.5. Resultados manuales, Mean Shift complementado	47
6.6. Medias de métricas manuales, Mean Shift complementado	48
6.7. Resultados de los tests automáticos	49
6.8. Resultados de los tests automáticos	51
6.9. Tiempo de tracking, Mean Shift mononúcleo	52
6.10. Tiempo de tracking, Tracking con Correlación	52

Capítulo 1

Introducción

Hoy en día, el aumento de la criminalidad y de los robos, ha provocado que un gran número de empresas y particulares demanden servicios de videovigilancia. Por lo tanto la seguridad es un tema en plena expansión. Esto hace interesante el hecho de iniciar nuevas vías de investigación sobre la materia.

Este proyecto, consiste en la realización de un estudio, sobre una técnica aplicable al seguimiento de personas y vehículos para un sistema de videovigilancia inteligente.

El concepto de videovigilancia inteligente, define el intento de dotar de inteligencia a las cámaras, es decir, darles capacidad de filtrar falsas alarmas, y detectar intrusiones mediante la utilización de algoritmos de visión por computador. Gracias a estas técnicas, se consigue mayor seguridad, debido a que se eliminan posibles fallos humanos, que se producen en un sistema de videovigilancia normal. Estos fallos vienen producidos, por la dificultad que conlleva el hecho de que un trabajador esté atento a unas cámaras en todo momento.

Éste es un proyecto propuesto por Davantis [Dav], empresa localizada en el Centro de Visión por Computador [Cvc] de la Universidad Autónoma de Barcelona [Uab]. Esta empresa se dedica a la creación de innovadores sistemas de análisis de vídeo, que analizan las imágenes de cámaras de seguridad, y producen alarmas cuando se detecta riesgo, con el objetivo de automatizar el proceso de videovigilancia. Su producto principal es el Daview, sistema de videovigilancia inteligente

que se compone de tres ramas. Daview perímetros, sistema perimetral que permite la detección de intrusos o individuos sospechosos, con un previo filtrado de cambios de luz, lluvia o árboles que se mueven con el viento. Daview contador, sistema de conteo que permite controlar el flujo de personas en un recinto o local. Daview CRA, sistema avanzado de videovigilancia, que ofrece la posibilidad de ser conectado a una central de alarmas.

Actualmente existen multitud de técnicas relacionadas con la visión por computador, aplicables a la videovigilancia inteligente. De este hecho nace el interés de estudiar dichas técnicas, para poder buscar posibles mejoras del producto actual, el Daview. Pero un sistema de videovigilancia como el Daview, se compone de diversos módulos, desde la detección de un elemento en escena, hasta su reconocimiento final, para determinar, por ejemplo, si ese elemento se trata de una persona o un vehículo. Este proyecto se basa únicamente en el proceso de seguimiento de estos elementos (independientemente de que sean vehículos o personas), una vez ya han sido detectados por otro módulo. A este proceso comúnmente se le llama Tracking.

Durante el desarrollo de esta memoria, se explica todo el proceso llevado a cabo para realizar el estudio de una nueva técnica de tracking, para el Daview y determinar si dicha técnica puede ofrecer mejoras de algún tipo a su actual sistema de tracking.

1.1. Motivaciones

Este proyecto, se fundamenta en la necesidad de elaborar un sistema de tracking estable ante cualquier tipo de situación, como parte vital de un sistema de videovigilancia inteligente. Gran parte de los errores que puede cometer un sistema de este tipo, se deben a problemas con el sistema de tracking, ya que si el seguimiento de un objeto falla, difícilmente se efectuará una clasificación correcta.

El seguimiento de un objeto en escena, resulta un problema complejo. Además durante este seguimiento pueden influir diversos aspectos que dificulten este

proceso. Un elemento puede cruzarse con otro, puede ocultarse temporalmente, puede mantenerse quieto en la escena, o moverse en diferentes condiciones de iluminación. Todos estos factores provocan pérdidas de tracking, problema que se propaga fácilmente, produciendo importantes errores en la salida final del sistema. Por lo tanto, mediante la correcta implementación del método adecuado en el módulo de tracking, se puede mejorar considerablemente la efectividad de estos sistemas.

1.2. Objetivos

El primer objetivo, es el de estudiar una técnica que pueda mejorar en algún sentido al sistema de tracking del Daview, es decir, que pueda mejorar su funcionamiento, manteniendo mejor el seguimiento de cada elemento o respondiendo mejor ante algún tipo concreto de contratiempo, como pueden ser cruces, oclusiones, o cualquier otro tipo de situación conflictiva. Para realizar este propósito se deberán implementar una serie de prototipos. Estas implementaciones, deberán ser correctamente evaluadas, para ello también se deberá desarrollar una metodología de evaluación apropiada, para concretar si un sistema de tracking mejora respecto a otro. Ésto no resulta un tema trivial, como podría ser la evaluación de un programa más funcional, sino que influyen un gran número de características que han de ser contempladas y correctamente seleccionadas al realizar los tests. Entre estas características, se encuentran el tiempo que el objeto esta siendo seguido, posición del objeto, identificadores asociados, etc. Finalmente, una vez obtenidos los resultados, el objetivo es extraer una serie de conclusiones para determinar si el método mejora o podría mejorar algún aspecto del tracking del Daview. Con estos datos, se estudiará la viabilidad de implementación del algoritmo en un producto comercial.

1.3. Estructura de la memoria

La estructura de la memoria es la siguiente. En el **capítulo 2**, se explica el estado actual de las técnicas de tracking y se justifica la elección del método escogido. En el **capítulo 3**, se exponen los requerimientos del proyecto, así como la viabilidad de éste. En el **capítulo 4**, se explica el método y todas las implementaciones que se han realizado para su estudio. En el **capítulo 5**, se presenta toda la metodología de evaluación que se seguirá durante el periodo de test. En el **capítulo 6**, se exponen todos los resultados para cada una de las implementaciones, y posteriormente son comparados entre sí. Finalmente, en el capítulo de conclusiones se valora el método de tracking, el método de evaluación y la viabilidad de las implementaciones desarrolladas.

Capítulo 2

Estado del arte

En este capítulo se dará una visión global sobre el estado actual del seguimiento de objetos mediante el procesamiento de imágenes, así como los procesos, pasos previos y algoritmos más utilizados para esta labor. Finalmente se introduce la línea general por la cual se realizarán las implementaciones y el estudio en el que se basa este proyecto.

2.1. Seguimiento de objetos

El seguimiento de objetos mediante videocámaras es un tema muy explorado en el campo de la visión por computador debido a su gran utilidad. Esta clase de seguimiento es utilizada en sistemas de reconocimiento basados en movimiento, sistemas de videovigilancia, adquisición de metadata para bases de datos de vídeo, interacción hombre-máquina, etc.

En el tema de la videovigilancia, es un hecho que el análisis de vídeo requerido para un sistema de este tipo se divide en tres principales etapas, las cuales son: Detección de objetos interesantes en escena, seguimiento de éstos y posteriormente su reconocimiento. Teniendo en cuenta el objetivo de este proyecto, únicamente se expondrán las dos primeras etapas de este proceso, las cuales están directamente relacionadas con el tema tratado y se dejará de banda todo lo referente al reconocimiento final de dichos objetos.

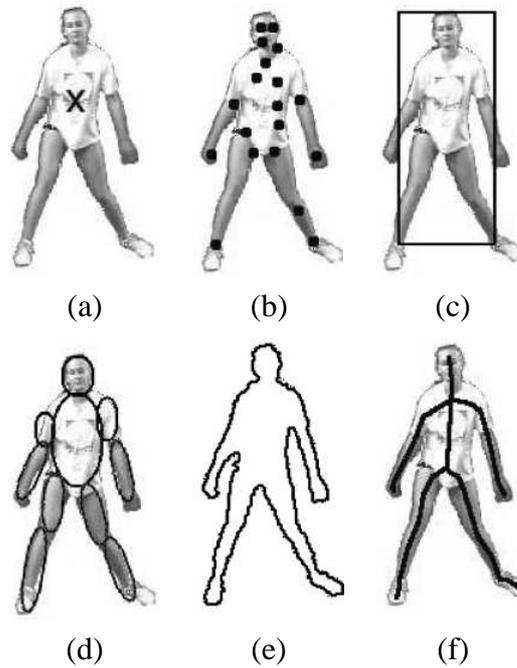


Figura 2.1: Modelos basados en forma

2.2. Modelos para la representación de objetos

El primer punto es la detección de objetos en escena, pero para ello antes hay que definir una representación para estos objetos. Para llevar a cabo esta tarea hay diversos modelos que se pueden dividir en dos grandes grupos: modelos basados en forma y modelos basados en apariencia [Yil06].

El primer grupo corresponde a los **Modelos basados en forma**. Éstos utilizan determinadas características referentes a la forma del objeto para hacer el seguimiento. Este tipo de modelos suelen ser los más utilizados para la mayoría de problemas de tracking y existen diversas versiones:

- **Modelo de puntos:** El objeto está representado por un solo punto (su centro) [Vee01] (ver Figura 2.1, a), o por un conjunto de puntos [Ser04] (ver Figura 2.1, b).
- **Modelo de formas geométricas primitivas:** El objeto está representado

por un elipse, rectángulo o alguna otra forma geométrica primitiva [Com03] (ver Figura 2.1, c). Esta forma es idónea para representar objetos rígidos pero a la práctica es muy utilizada también para cualquier otro tipo de objetos no-rígidos.

- **Modelo de formas articuladas:** Este método es una evolución del método anterior, ya que se basa en la utilización de formas geométricas primitivas, pero en este caso para representar cada una de las partes rígidas de un determinado objeto, separadas por una articulación (ver Figura 2.1, d).
- **Modelo de silueta y contorno:** Se basa en utilizar el contorno del objeto para definir sus fronteras y así poder obtener la parte interior, llamada silueta [Yil04] (ver Figura 2.1, e). Este método resulta idóneo para representar objetos no-rígidos complejos pero es mucho más costoso que los anteriores.
- **Modelo de esqueleto:** Consiste en la extracción del esqueleto del objeto mediante el uso previo de su silueta [Bal82] (ver Figura 2.1, f). Este método es muy utilizado cuando se tiene previsto realizar el paso de reconocimiento de objetos.

El segundo tipo son los **Modelos basados en apariencia**. Este tipo de modelos utilizan la información cromática o de contornos que ofrece cada uno sus píxeles. Es bastante frecuente ver sistemas que utilizan una representación basada en forma combinada con su apariencia. De esta forma obtienen más información del objeto [Coo01], ya que la utilización del modelo de apariencia ofrece una gran información complementaria, pero a menudo resulta insuficiente si no viene acompañada de algún componente espacial. Existen diversas representaciones de este tipo:

- **Modelo de densidades de probabilidad:** Define un objeto únicamente por las estimaciones de las funciones de densidades de probabilidad. Para extraer esta medida existen diversos métodos, ya sea mediante Gaussianas [Zhu96], ventanas Parzen, [Elg02] o histogramas [Com03].

- **Patrones:** Este modelo se basa en formas geométricas o siluetas predefinidas para identificar objetos en escena [Fie97]. Cada una de estas siluetas contendrá información cromática del objeto. Es un método sencillo pero solo apto para objetos que no varían excesivamente en el tiempo.
- **Modelo de apariencia activo:** Este modelo se basa en generar información referente a la forma y apariencia del objeto. La forma viene definida por una serie de marcas basándose en un sistema similar a la detección del contorno, y cada una de éstas marcas contiene un vector de apariencia con diversas características como el color, textura o la magnitud del gradiente [Edw98].
- **Modelo de apariencia multivista:** Este modelo se basa en guardar la información de diferentes vistas de un mismo objeto, para así ser menos susceptible a cambios de forma o apariencia (por ejemplo, personas que giran sobre sí mismas). Para almacenar estas vistas utiliza una codificación mediante subespacios mediante métodos como el PCA (Principal Component Analysis) [Shl05] y el ICA (Independent Component Analysis) [Mug97]

2.3. Detección de objetos en escena

Una vez escogido el sistema de representación de objetos, el siguiente punto lógico es decidir un sistema para detectar dichos objetos en función a la representación escogida. Para ello existen diversos métodos:

- **Detectores de puntos:** Este tipo de detectores se basan en encontrar puntos característicos en una imagen utilizando su textura (ver Figura 2.2, a). Hay diversos algoritmos para realizar esta tarea, como por ejemplo “Moravec’s interest operator” [Mor79], “Harris interest point detector” [Har88], “KLT detector” [Shi94] y “SIFT detector” [Low04].
- **Background subtraction:** Este método se basa en la adquisición de un modelo de background, el cual se definirá a partir de una secuencia de frames y se irá actualizando a lo largo del análisis del vídeo. En la figura 2.2 (c)

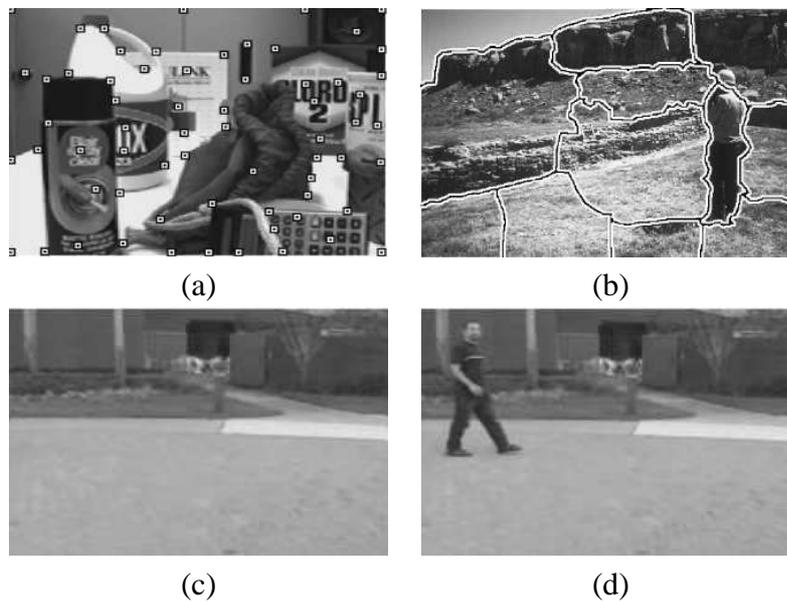


Figura 2.2: Detección de objetos en escena

podemos ver el modelo de background para el frame de la figura 2.2,(d). El hecho de tener este modelo nos ayuda mediante una simple resta, o mediante algoritmos más eficientes [Jai79] [Sta00], a detectar nuevos objetos en escena.

- **Segmentación:** Este método segmenta la imagen en regiones similares (ver Figura 2.2, b). Para realizar esta tarea se plantean dos problemas, primero buscar un buen criterio a la hora de escoger regiones similares, y posteriormente buscar una manera de almacenar esta segmentación. Para solucionar estos problemas existen diversos algoritmos como por ejemplo “Mean-Shift Clustering” [Com02], “Image Segmentation Using Graph-Cuts” [Shi00] o “Active Contours” [Kas88].
- **Entrenamiento supervisado:** Consiste en aplicar alguno de los múltiples métodos de clasificación existentes para determinar si aparece algún elemento nuevo en escena. Algunos de estos métodos podrían ser Redes neuronales [Row00], “Adaptative Boosting” [Vio03] o “Support Vector Machines” [Pap98].

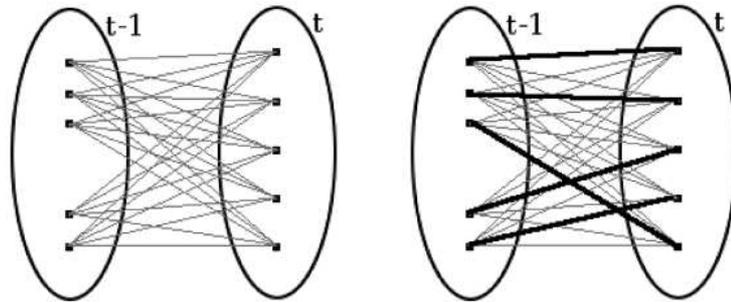


Figura 2.3: Elección de correspondencias, Tracking por puntos

2.4. Modelos de tracking

Una vez escogido el método para detectar los objetos en escena, el siguiente paso es escoger un modelo para realizar el propio tracking del objeto. Para ello, al igual que en los pasos anteriores disponemos de diversas posibilidades que se pueden dividir en tres grandes grupos: Tracking por puntos, núcleo y silueta [Yil06].

El **Tracking por puntos** se usa en el caso de tener un modelo de objeto basado en puntos. El objetivo es relacionar los puntos en un determinado frame con los mismos puntos en el frame anterior. Los algoritmos para establecer estas correspondencias se dividen en dos grandes grupos:

- **Métodos deterministas de correspondencia:** En este tipo de métodos se relacionan los puntos de un frame en un instante de tiempo $t-1$ con los mismos puntos en t . Para ello se utilizan todas las combinaciones posibles a la hora de asociar dichos puntos y posteriormente se escogen las correspondencias correctas (ver Figura 2.3) mediante métodos de asignación óptima como por ejemplo el “Hungarian algorithm” [Kuh55].
- **Métodos estadísticos de correspondencia:** En este caso se trata, al igual que en el caso anterior, de encontrar relaciones entre puntos. Pero esta vez, teniendo en cuenta una cierta incertidumbre (cuantificada en forma de error) en las medidas [Isa98].

El **Tracking de Núcleo** (Kernel Tracking), a diferencia del anterior, basado en puntos muy concretos de la imagen, se basa en una región primitiva de donde se extraen una serie de parámetros para realizar el seguimiento de cada objeto. Se puede dividir este tipo de tracking en tres métodos:

- **Tracking basado en patrones:** Se basa en la utilización de un patrón para buscar el objeto a través de la imagen, normalmente mediante correlaciones [Sch02] o algún método similar. Este método tiene bastantes pérdidas cuando la apariencia del objeto varía mucho durante la escena. Sin embargo, funciona muy bien para el tracking de objetos que no modifican su forma.
- **Tracking basado en modelos de apariencia:** Este modelo se basa en la búsqueda de ciertas características del objeto, como por ejemplo podrían ser histogramas de color, gradientes u otros modelos de obtención de densidad, mediante los cuales prevé la dirección en la que se desplaza el objeto sin necesidad de recorrer toda la imagen. Uno de estos métodos es el “Mean Shift”. [Com03].
- **Tracking mediante modelos de apariencia multivista:** Consisten en aumentar el aprendizaje de un objeto basándose en múltiples vistas generadas de forma offline. Es decir, con un procesamiento previo a la exposición del sistema a un tracking a tiempo real, con el objetivo de hacer al tracking menos susceptible a cambios drásticos de la forma del objeto. Para ello existe un método basado en subespacios llamado “Eigenspace” [Bla98], basado en el “Principal Component Analysis” (PCA) [Shl05].

Finalmente, el **Tracking Basado en Siluetas** esta orientado al tracking de objetos complejos, para los cuales no se puede usar una simple forma geométrica. Este tipo de Tracking se puede dividir en dos grandes grupos:

- **Tracking de forma:** Este método es similar al tracking mediante patrones, ya mencionado anteriormente, pero utilizando siluetas complejas como patrón de búsqueda [Cole04].

- **Tracking de contorno:** Este método no considera una silueta fija, sino que intenta seguir frame a frame la evolución del contorno de cada objeto para obtener así un tracking más preciso en objetos que tienden a cambiar de forma durante la escena [Isa98].

2.5. Discusión

Tal y como se puede ver, existen un gran número de métodos para realizar un modelo de tracking, cada uno con sus ventajas e inconvenientes respecto a situaciones determinadas. Por lo que respecta a este proyecto, y considerando el actual sistema de tracking del Daview, se enfocó la investigación en un sistema basado en modelo de apariencia, el cual puede mantener todo su potencial independientemente de la dirección o velocidad del objeto, y no debe perder el tracking cuando se deja de tener información espacial (cuando el objeto detiene su movimiento).

Esta decisión fue fundada al analizar dicho sistema de tracking y observar que precisamente fallaba en aquellos puntos donde la información espacial era nula (una persona que de repente se para y luego continúa su trayecto) o bien confusa (cambios de dirección bruscos, o cruces). De entre todos los modelos de objetos basados en apariencia, enseguida se descartó el basado en patrones [Fie97], debido a su poca flexibilidad frente a objetos que varían su forma durante la escena. El sistema de tracking necesario para este estudio, debía ser capaz de seguir personas, las cuales pueden moverse mucho a lo largo de la escena y por lo tanto cambiar de forma y apariencia. Además también provocaría problemas con objetos lejanos acercándose, o viceversa. En cuanto a los modelos de apariencia activos [Edw98], parecen ofrecer buenos resultados. Sin embargo, son modelos muy costosos computacionalmente, además de suponer un gran coste de memoria (lo cual podría suponer un problema en escenas con gran cantidad de elementos). Otro método interesante son los modelos de apariencia multivista, los cuales utilizan PCA [Sh105] para recoger múltiples vistas de un objeto mediante el uso de subespacios. Pero este método, a parte de ser inviable computacionalmente para un sistema a tiempo real, donde prima la velocidad de procesamiento, necesita de

un procesamiento previo, conociendo los objetos que van a aparecer en escena. Por lo cual queda totalmente descartado en un sistema de vigilancia. Finalmente se optó por adoptar un modelo de densidades de probabilidad, el cual proporciona la información cromática del objeto utilizando poca memoria. Una vez escogido el modelo de objeto, evidentemente el sistema debería basarse en un tracking de núcleo (Kernel Tracking). En este tipo de tracking destaca un método llamado Mean Shift [Com03]. Este método utiliza un modelo basado en histogramas, relativamente ligero de memoria y que se basa en la búsqueda del punto más cercano a una función de densidad que define al objeto. Este método será explicado con más detalle en capítulos posteriores.

Capítulo 3

Análisis

En este capítulo se hará un análisis de la situación inicial del proyecto. Para ello se destacarán los requisitos principales que deberán solventar las implementaciones, se justificará la viabilidad del proyecto y se planteará la distribución temporal de cada una de las tareas de las que se compone.

3.1. Requisitos

Las implementaciones que se llevarán a cabo se basan en realizar un estudio sobre la utilización de un determinado tipo de técnica, el Mean Shift. El principal requisito que deberán cumplir todas las implementaciones es que dado un determinado vídeo o secuencia, el programa devuelva a tiempo real la posición de cada uno de los elementos a los cuales se está siguiendo. Por ello se considera importante un tiempo de ejecución razonable del algoritmo para poder adaptarse a la propia funcionalidad del programa. Además, para finalmente ser introducido en el sistema, deberá obtener unos resultados de los tests automáticos (los cuales serán explicados posteriormente), superiores a los del Daview actual.

En cuanto a la implementación de cada una de las soluciones está pensada para ser utilizada en el sistema implementado por Davantis. Eso conlleva una serie de requisitos. En cuanto al lenguaje de programación deberá ser C++ utilizando las funcionalidades de las librerías OpenCv [OpCv]. Otro requisito será la estructura

del programa, la cual deberá respetar la estructura general del Daview para su adecuada cohesión.

3.2. Estudio de viabilidad

Este estudio se basa en la implementación y análisis de un método de tracking utilizando el Mean Shift. Esta técnica, dispone de documentación a cerca de diversas implementaciones en tiempo real (incluso a frame-rates de hasta 1 fps), usadas para videovigilancia. Por lo tanto, al tratarse de un método muy documentado, su implementación no debería suponer un problema. Para llevar a cabo el periodo de evaluación, punto decisivo del estudio, se necesitará una cierta infraestructura, es decir:

- Software específico para el desarrollo y testeo de la aplicación.
- Videocámaras para la obtención del material necesario para la fase de test.
- Servidores para el procesamiento de los tests. Esto es necesario para la evaluación del sistema frente a un gran número de videos y la imposibilidad de llevarlo a cabo en un tiempo razonable sin el uso de servidores.

Todos estos elementos son proporcionados por Davantis. Así como soporte por parte de un tutor.

3.3. Planificación temporal

En este apartado se presenta la planificación temporal ideada para el desarrollo del proyecto.

1. Octubre - Noviembre

- **Introducción al sistema:** Tarea para la cual se realizará un acercamiento a la plataforma utilizada por Davantis, y al OpenCv (mediante una serie de tutoriales).

- **Implementación de un sistema de Tracking Sencillo:** Basado únicamente en la búsqueda y asignación de la región de movimiento más cercana del objeto. El objetivo de esta implementación únicamente es el de familiarizarse con la estructura general del sistema.

2. Noviembre - Enero

- **Decisión del método a utilizar:** Lo cual implicaría la búsqueda de los métodos existentes en el campo y la decisión del más adecuado para el estudio que se va llevar a cabo.
- **Elección de la metodología de evaluación:** Lo cual incluiría buscar los métodos para evaluar el proyecto y el conjunto de vídeos de test apropiado para ellos.
- **Evaluación del Tracker de Davantis:** Evaluación manual y automática siguiendo la metodología ya escogida para el sistema de tracking de Davantis.

3. Febrero - Abril

- **Implementación:** De cada una de las soluciones que se planteen durante este periodo de tiempo.

4. Abril - Mayo

- **Periodo de Pruebas:** Donde se realizarán todos los tests siguiendo la metodología ya escogida anteriormente.
- **Mejoras de implementación:** De ser necesario, incorporación de ciertas mejoras para las implementaciones.

5. Mayo-Junio

- **Redacción de la Memoria:** Se dedicará este último periodo a la redacción de la memoria del proyecto y posiblemente a la finalización de algunos tests.

Capítulo 4

Método e Implementación

En este capítulo se explicará en que se basa el método utilizado, el Mean Shift. Posteriormente se expondrá con detalle cada una de las tres implementaciones que se han realizado para el correcto estudio del método, así como las fuentes de información en las que se han basado.

4.1. Mean Shift

Antes de explicar propiamente todas las versiones implementadas se expone una breve explicación sobre el concepto de Mean Shift. El Mean Shift es una técnica no paramétrica que analiza un espacio de densidades que puede ser más o menos complejo. La técnica se basa en la aplicación recursiva de un método para encontrar el punto estable más cercano a la función de densidad.

Así pues, considerando un vector x , el método consiste en iterar. En cada una de estas iteraciones deberemos encontrar el nuevo vector x' . Si consideramos esto como un incremento, $x' = x + \Delta_x$, cada iteración se basará únicamente en la búsqueda de este incremento.

La búsqueda de este valor se basa en el acercamiento al modelo de densidad que define al propio objeto, mediante la siguiente formula:

$$\Delta_x = \frac{\sum_a K(a-x)w(a)a}{\sum_a K(a-x)w(a)} - x = \frac{\sum_a K(a-x)w(a)(a-x)}{\sum_a K(a-x)w(a)}$$

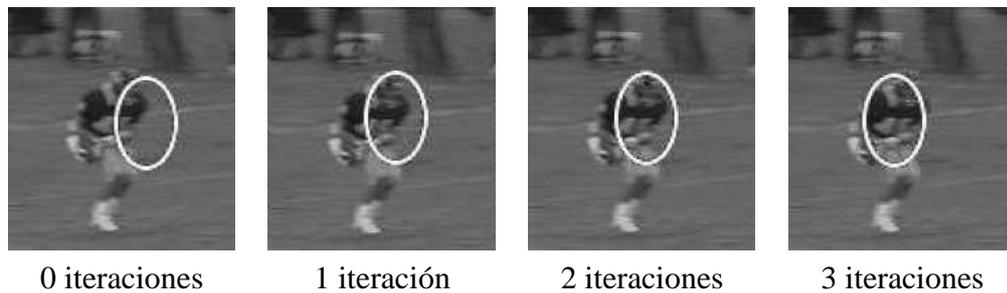


Figura 4.1: Tres iteraciones del Mean Shift

Donde a son los píxeles correspondientes al área de búsqueda del Mean Shift (la cual suele ser del mismo tamaño que el modelo del objeto), K es una función que nos ofrece la información del núcleo, normalmente una gaussiana, y $w(a)$ un peso asignado en función de la información cromática del píxel.

Deberemos pues iterar en este método repetidas veces, efectuando la actualización pertinente $x' = x + \Delta_x$ a cada iteración. Los resultados obtenidos al realizar ésto, serán similares a los que muestra la figura. 4.1, para cada iteración del método. En dicha figura podemos observar como cada iteración el modelo se aproxima más a la solución correcta, llegando finalmente a ella.

El método es robusto pero tiene sus limitaciones. El hecho de tener que trabajar en un área pequeña de búsqueda hace que únicamente nos sirva en aquellas ocasiones en las que los objetos se solapan de un frame a otro. De no ser así, como ocurre a frame-rates bajos, se pierde el tracking. Pero ésto, como se ve en el siguiente apartado, tiene solución. Existen métodos que extienden el uso del Mean Shift mejorandolo para este tipo de situaciones.

4.2. Artículos escogidos

Una vez decidido el método a utilizar se decidió buscar más información acerca de él y de sus posibles variantes. Existen diversos artículos relacionados con este método, muchos de los cuales presentan posibles soluciones a un sistema de tracking. Finalmente este proyecto se basó en dos de ellos.

El primero de ellos es “Kernel-Based Object Tracking” [Com03]. Éste es un artículo muy conocido e innovador en su campo y explica como diseñar un sistema de tracking basado en Mean Shift. Este artículo fue de gran utilidad para asentar conceptos sobre el método y como referencia a la que recurrir en diversas ocasiones para aclarar dudas que surgían con otros artículos que tomaban a éste como su punto de partida.

El segundo de ellos es “Object Tracking in Low-Frame-Rate Video” [Por05], este artículo se ve englobado dentro de los que mencionaba anteriormente, los cuales toman como punto de partida el artículo de Comaniciu [Com03], pero intenta introducir una serie de modificaciones con el objetivo de encontrar una solución más robusta para el sistema de tracking y así poder llegar a trabajar a un frame-rate muy reducido, característica muy importante para el tracking a tiempo real.

La mejora más destacable que presenta el artículo es extender la amplitud de la búsqueda del Mean Shift tradicional, basada en una búsqueda lineal a partir de la última localización del objeto, a una búsqueda multinúcleo en donde cada uno de los núcleos sería una región de la imagen donde se haya detectado algún movimiento dentro de un radio de acción determinado. Otra característica importante, que también se plantea es el uso de la información del fondo de la escena (background) para el calculo del Mean Shift. Ésto tiene como objetivo el reducir la distorsión que provoca la porción de background dentro de la región que define el objeto, la cual no aporta nada de información referente a éste.

Tomando como punto de partida estos dos artículos se llegó a la implementación de tres sistemas diferentes, basados en unos mismos conceptos iniciales.

La primera implementación, **Mean Shift mediante Blobs**, se basa en el segundo de los artículos mencionados y tiene como objetivo el servir como sistema de tracking independiente. La segunda implementación, **Mean Shift complementado**, esta ideada como técnica complementaria a un sistema de tracking con diversas características. En cuanto a su base es similar a la del primer método. La tercera y última implementación, **Mean Shift Mononúcleo**, se basa en el Mean Shift más clásico y tiene como objetivo aguantar el tracking en objetos que se detienen en la escena pero continúan efectuando ligeros movimientos (girar sobre sí

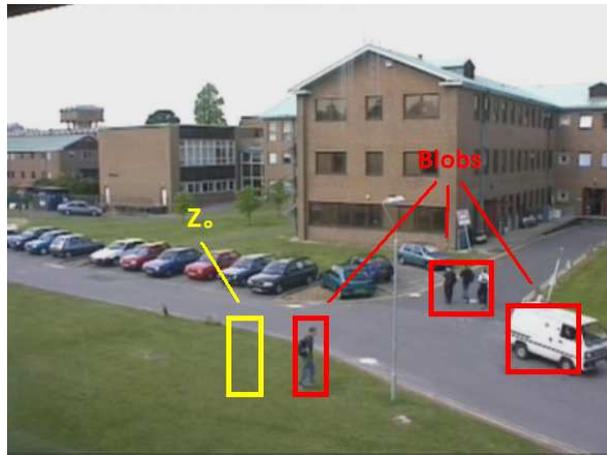


Figura 4.2: Frame mostrando los múltiples núcleos

mismo, desplazarse ligeramente, etc.).

4.3. Mean Shift mediante Blobs

En esta primera implementación se desarrolló un sistema muy parecido al del artículo de Porikli [Por05], es decir, la implementación se centró en una búsqueda multinúcleo. Para ello en cada frame se recoge la información referente a las regiones donde se haya producido movimiento respecto al frame anterior, a estas regiones se las denomina Blobs. También se utilizó información del background, eliminando así la información no referente al objeto y que únicamente distorsiona el resultado. A continuación se explica la implementación con más detalle.

4.3.1. Mean Shift multinúcleo

La idea de esta búsqueda multinúcleo es la de fijar un cierto radio de acción y seleccionar todos los Blobs dentro de este radio (ver Figura. 4.2), donde Z_0 sería la posición en el frame anterior. Una vez se tiene la lista de Blobs con sus correspondientes centros se procederá a realizar la búsqueda mediante Mean Shift de forma iterativa para cada uno de estos puntos (ver Figura. 4.3) y finalmente se

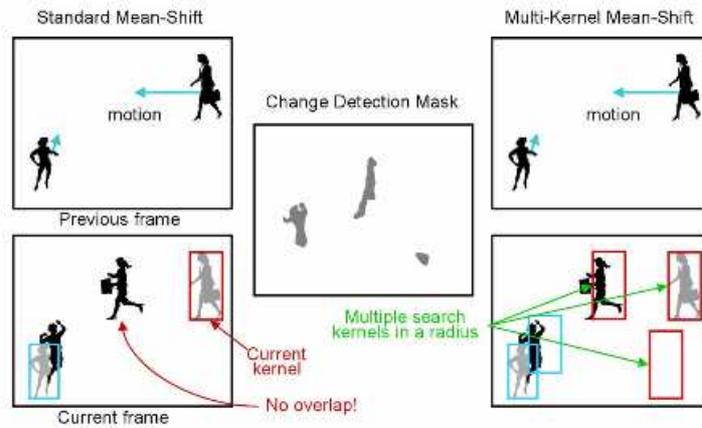


Figura 4.3: Muestra esquemática de la búsqueda multinúcleo

escogerá al Blob donde el Mean Shift haya devuelto una medida de similitud más elevada. Esta similitud se decide mediante el coeficiente de Bhattacharya que será explicado con más detalle posteriormente. Gracias a este método se soluciona un gran problema que surge con el sistema de Mean Shift clásico, al procesar una escena a un bajo frame-rate. Este problema se basa en que si un objeto en un determinado frame no está en una posición solapada con la del frame anterior, como puede ocurrir fácilmente en objetos a gran velocidad y bajo frame-rate, el Mean Shift pierde el seguimiento del objeto, en cambio con esta solución la nueva posición del objeto será detectada como un nuevo Blob y se realizará una búsqueda a partir de ella, de forma que no se perderá a dicho objeto.

4.3.2. Pseudocódigo

Primero se dará una visión global del algoritmo mediante la ayuda de su pseudocódigo.

Entrada: Localización inicial z_0 y serie de Blobs dentro del radio de acción $\{c_i\}_{i=1..l}$

1. $B_{max} \leftarrow 0, i \leftarrow 0$
2. Inicializar el Mean Shift en la última localización del objeto

3. **while** $i \leq l$
4. Buscar vector Mean Shift $m(z_0)$
5. **while** $\eta(z_0) < \eta(z_0 + m(z_0))$
6. $z_0 \leftarrow z_0 + m(z_0)$
7. Buscar vector Mean Shift $m(z_0)$
8. **if** $B_{max} < \eta(z_0)$
9. **then** $B_{max} \leftarrow \eta(z_0), z_1 \leftarrow z_0$
10. Inicializar el Mean Shift para $z_0 = c_i$
11. $i \leftarrow i + 1$

El algoritmo consta de un bucle exterior que recorrerá todos los Blobs contenidos dentro del radio de acción prefijado. Para cada uno de ellos lo primero será iterar en el cálculo del Mean Shift y desplazamiento del núcleo en base a este vector. Esto se realizará mientras se aprecie mejora en la nueva posición del núcleo (mediante un coeficiente de similitud que será explicado posteriormente). Una vez acabada esta iteración se compara esta similitud para ver si es superior a la de alguna de los núcleos que hayan sido calculados. De ser así, se asigna este núcleo (sin tener en cuenta los desplazamientos por el Mean Shift) como candidato. Finalmente el último candidato al terminar todas las iteraciones será la posición a la que se traslade el modelo de objeto.

4.3.3. Modelo de objeto

El modelo de objeto escogido se compone principalmente de dos partes, una con la información espacial del objeto dentro de la imagen y otra con el modelo de apariencia. La parte de información espacial se compone de un modelo basado en formas geométricas (un rectángulo), en el cual guardamos de las coordenadas correspondientes al centro del objeto (x_i) y a los límites que fijan su caja mínima.



Figura 4.4: Regiones para Histogramas

También guardaremos la información relativa al vector de Mean Shift ($m(z_0)$) y al área candidata de búsqueda (y_j) basándonos en dicho vector. Por otro lado en cuanto a información sobre el modelo de apariencia, cada objeto posee tres histogramas. El primero utilizado como historial contiene la información del objeto actualizada frame a frame (q_s). El segundo hace referencia a la información del área candidata (p_s). Por último, el tercero contiene el modelo de background (b_s). En la figura. 4.4 se pueden ver las regiones de las cuales se realizan los histogramas para un determinado instante T, donde H1 es el histograma historial, H2 el área candidata y H3 el background. Cabe decir que los histogramas, a excepción del primero, el cual será actualizado constantemente, deberán ser sustituidos a cada frame.

4.3.4. Vector de Mean Shift con información del background

El hecho de adaptar el cálculo del Mean Shift mediante el uso de la información del background tiene como objetivo eliminar dicha información del modelo de objeto, para así dejar únicamente (o principalmente) la información referente al propio objeto. Es decir, cuando se obtiene información sobre la apariencia de un objeto, realmente se está obteniendo la información de toda su caja mínima. Ésto incluye los píxeles que están dentro de ella pero no pertenecen al objeto. Al eliminar información de los colores referentes al background en esta zona, se está penalizando toda la información del modelo que se parezca a la del fondo, y así se consigue la información referente solo al objeto.

En cuanto al cálculo del Mean Shift, para llevarlo a cabo se deben realizar una

serie de procedimientos previos que se explican a continuación.

Histogramas

El primer paso es el de crear los histogramas necesarios para el cálculo del Mean Shift. Es decir, cada uno de los tres de los que se compone el modelo de objeto explicado anteriormente. A continuación se muestra la forma general para crear dicho histograma al cual llamaremos $\{q_s\}_{s=1..m}$, donde m es el número de columnas deseado, el cual vendrá determinado por el número de regiones en el que se desee dividir el espacio cromático (512 en este caso):

$$q_s = Q_1 \sum_{j=1}^M \left(\left\| \frac{y_j}{h_t} \right\|^2 \right) \delta (\hat{m}(v_j) - s)$$

Donde Q_1 es una constante que utilizamos para normalizar la expresión, la cual satisface que $\sum_{s=1}^m q_s = 1$, \hat{m} es la función que mapea el RGB del píxel y nos devuelve la columna del histograma que le corresponde, h_t es el ancho de banda correspondiente a la mitad del tamaño del objeto (diferente para cada dimensión), δ es la delta de Kronecker (la cual consiste en devolvernos un 1 en caso de que su parametro sea 0, y un 0 en caso contrario) y K_N es el perfil del núcleo, el cual corresponde a un tipo de Gaussiana multidimensional que nos ayudará a dar un peso a cada píxel en función de su distancia con el centro:

$$K_N(x^*) = (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2}x^*}$$

Donde d es el número de dimensiones utilizado, para este caso $d = 2$.

Dicho ésto, se procede a calcular el histograma del candidato $p(z)$ y el del background $b(z)$ en cada iteración para ser substituidos por el histograma anterior. Estos histogramas se utilizarán posteriormente para calcular el vector de Mean Shift. Después calcularemos el histograma del objeto en cuestión a cada frame, pero éste no será substituido directamente por el anterior sino que se realizará una media ponderada por un cierto factor. El histograma resultante de esta media será el nuevo histograma que se tomará como historial.

Coefficiente de Bhattacharya

Una vez calculados los histogramas, se debe decidir el peso que se le dará a los píxeles cuando se calcule el vector de Mean Shift. Para ello se utilizará el coeficiente de Bhattacharya.

Este coeficiente es una fórmula estadística que mide la similitud entre dos distribuciones probabilísticas y es normalmente utilizado en problemas de separación de clases. La forma normal de este coeficiente es la siguiente: $\eta(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)}$ para las distribuciones p y q , pero en nuestro caso consideraremos este coeficiente en un determinado punto z como:

$$\eta(z) = \sum_{s=1}^m \sqrt{p_s(z)} \left(\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(z)} \right)$$

Donde α_f y α_b corresponden al peso que se le quiera dar a cada una de las dos distribuciones para así penalizar mas o menos el parecido con el modelo de background.

Esta fórmula posteriormente es sujeto de una serie de reducciones tomando z_0 como localización inicial y aplicando una expansión de Taylor en los valores $p_s(z_0)$ y $b_s(z_0)$. Posteriormente se utiliza la definición de $p(z)$ y $b(z)$, para llegar a la siguiente simplificación:

$$\eta(z) \approx Q_2 + Q_3 \sum_{i=1}^N w_i k_N \left(\left\| \frac{z - x_i}{h_t} \right\|^2 \right)$$

Donde Q_2 y Q_3 son constantes y w_i es el peso que buscábamos, el cual viene determinado por:

$$w_i = \sum_{s=1}^m \frac{\alpha_f \sqrt{q_s} - \alpha_b \sqrt{b_s(z_0)}}{2\sqrt{p_s(z_0)}} \delta[\hat{m}(x_i) - s] - \sum_{s=1}^m \frac{\alpha_b \sqrt{p_s(z_0)}}{2\sqrt{b_s(z_0)}} \delta[\hat{m}(x_i) - s]$$

Vector del Mean Shift

Una vez obtenido este peso ya es posible calcular el vector del Mean Shift, el cual determinará la dirección en la cual proseguir la búsqueda y que presenta la

siguiente forma:

$$m(z_0) = \frac{\sum_{i=1}^n (x_i - z_0) w_i g_N \left(\left\| \frac{z_0 - x_i}{h} \right\|^2 \right)}{\sum_{i=1}^n w_i g_N \left(\left\| \frac{z_0 - x_i}{h} \right\|^2 \right)}$$

Donde $g_N(x^*) = -k'_N(x^*)$

4.4. Mean Shift complementado

La idea de esta segunda implementación es la de integrar el sistema de Mean Shift que se acaba de exponer en una versión del tracking del Daview. Esto se hace con el objetivo de combinar la información del modelo de apariencia de este método con diversas características (velocidad, trayectoria, distancia, etc.) que usa el Daview en su sistema de tracking. Cada una de estas medidas están integradas como módulos independientes y cada uno de estos módulos retorna un cierto valor dentro de su rango. En función de estos valores, mediante unas funciones heurísticas se escoge la información de los módulos que ofrecen mejores resultados. La nueva posición del modelo vendrá determinada por la que marquen estos módulos.

4.4.1. Rango de la solución

El principal problema para integrar el código a esta plataforma fue decidir el valor del rango de soluciones. Para esto se desarrollaron dos subsistemas independientes.

El primer sistema se basa en la utilización del coeficiente de Bhattacharya. Debido al reducido rango de valores que ofrece este sistema, se multiplica el resultado por un cierto factor k . Ésto normaliza el resultado en un rango aproximadamente entre 0 y 100.

El segundo sistema se basa en una cierta medida de probabilidad mencionada también en el artículo de Porikli [Por05], basada en un predictor de ventanas Parzen con la siguiente forma:

$$l_j(u_i) = \frac{1}{Dh_c^3} \sum_{k=1}^D k_N \left(\left\| \frac{u_i - v_{jk}}{h_c} \right\|^2 \right)$$

Donde h_c es el ancho de banda del color 3D para el cual se recomienda dar un valor de $h_c = 16$, u_i el RGB de un determinado píxel y v_{jk} el mismo píxel del objeto para frames anteriores de un historial de tamaño D .

Gracias a esta medida y basándose también en la distribución de los píxeles para darles un peso determinado, es posible calcular dicha probabilidad de la siguiente forma:

$$L(z) = \frac{1}{N} \sum_{i=1}^N l_j(u_i) k_N \left(\left\| \frac{x_i - z}{h} \right\|^2 \right)$$

Para llevar a cabo este método se necesita una ampliación del modelo de objeto con tal de incluir el historial de tamaño D . En este caso se implementó un historial de cinco regiones de píxeles que se actualizan frame a frame de forma cíclica.

4.5. Mean Shift mononúcleo

La idea de esta tercera y última implementación es la de utilizar el algoritmo del Mean Shift pero sin basarse en la información de los Blobs cercanos, sino únicamente en la propia información que ofrece el vector de Mean Shift. Por lo tanto se basará en un sistema de Mean Shift más clásico [Com03]. Eso sí, se mantiene el modelo de background al igual que en los dos métodos anteriores. Para ello se reutilizan funciones de dichos métodos, pero esta vez únicamente iterando en el calculo del vector de Mean Shift y su comparación mediante Bhattacharya. Así a cada frame se obtiene un único candidato. El modelo se desplazará al punto final de este candidato (no existirán puntos fijos como los de los Blobs de los otros métodos). A continuación se presenta el pseudocódigo.

Entrada: Localización inicial z_0

1. Inicializar el Mean Shift en la última localización del objeto
2. Buscar vector Mean Shift $m(z_0)$

3. **while** $\eta(z_0) < \eta(z_0 + m(z_0))$
4. $z_0 \leftarrow z_0 + m(z_0)$
5. Buscar vector Mean Shift $m(z_0)$
6. $z_1 \leftarrow z_0$
7. Realizar Adaptación de Escala (z_1)

Esta implementación es algo diferente a las anteriores. Gracias a la adaptación de escala (que será explicada en al siguiente sección) se considera viable su uso como sistema de tracking independiente. Sin embargo presenta muchos problemas de pérdidas de objetos a un frame-rate reducido o con objetos demasiado rápidos. Por lo tanto este sistema no se considera comparable en este punto a las dos implementaciones anteriores y por lo tanto no será evaluado como tal. El punto fuerte de esta implementación es el hecho de poder mantener un objeto que se detiene en una posición concreta pero no se mantiene estático (gira sobre sí mismo, se desplaza ligeramente, etc.), sin llegar a perderlo. Así pues, esta implementación estaría destinada a ser complemento de un sistema de tracking, y actuaría cuando se detectase que un objeto se detiene.

4.5.1. Adaptación de escala

Un problema que surge al implementar este algoritmo es que a diferencia de los anteriores, en éste sera un mismo modelo el que ira recorriendo toda la escena. En los otros dos métodos se obtenía el núcleo candidato final y éste siempre venia asociado a un Blob con su correspondiente información espacial. Ahora, al no tener esta información, los objetos que modifiquen mucho su tamaño a lo largo de la escena, como por ejemplo objetos lejanos que se acercan o viceversa podrán provocar problemas (Ver Figura. 4.5) .

Para solucionar ésto se debe realizar una adaptación del tamaño del modelo a cada frame. Para llevar a cabo esta tarea el articulo de Porikli [Por05] ofrece



Figura 4.5: Problemas con la escala del objeto



Figura 4.6: Actualización del modelo de background

una solución en la cual intervienen los bits de la caja mínima del objeto diferenciándolos según si están dentro de la máscara del objeto (es decir, pertenecen al objeto) o no (es decir, pertenecen al background). A partir de esta información hace comparaciones con pequeños incrementos de la caja mínima en cada una de las posibles direcciones y determina si dichos incrementos son adecuados. El problema de este método es que al no utilizar información de Blobs no se dispone tampoco de la información referente a la máscara del objeto. La solución adoptada fue la de ampliar/reducir la caja mínima un cierto porcentaje en cada una de las cuatro direcciones, tal y como decía el artículo, pero en vez de utilizar la información referente a la máscara del objeto, utilizar el coeficiente de Bhattacharya. Para su correcto funcionamiento se tiene que ajustar la comparación de forma de que el nuevo coeficiente sea mayor al anterior multiplicado por un cierto factor k , ($\eta(z_0) * k < \eta(z_0 + m(z_0))$), para decidir si el aumento/reducción se debe efectuar.

4.5.2. Background

Un punto que en un inicio se presentó problemático para este método fue el hecho de que el algoritmo utilizara la información del background como parte fundamental de su funcionamiento. En cambio el sistema a donde va destinado este método utiliza una actualización del modelo de background (ver Figura. 4.6). Este hecho hace que la información del background se acabe perdiendo cuando el objeto permanece estático, por lo tanto se debe a ampliar el modelo de objeto y guardar una imagen con el background de la escena en el momento que el objeto es detectado. Durante todos los calculos que impliquen utilizar el background se utilizará esta imagen.

Capítulo 5

Metodología de evaluación

5.1. Problemas al evaluar un tracker

Evaluar un sistema de tracking no es trivial ya que en su evaluación influyen muchos factores. No es fácil definir lo que sería un buen sistema de tracking. Quizás un buen tracking debería seguir al objeto en su posición exacta, o quizás será mas importante mantener el objeto más tiempo aunque la posición detectada no sea tan precisa. Es decir, en la mayoría de casos no existe una forma clara de decidir con total certeza si un sistema mejora o no al anterior, ya que además estos resultados variarán mucho según la situación y condiciones de la escena. De hecho existen diversas líneas de investigación únicamente centradas en este tema. Como por ejemplo un sistema de generación de secuencias de pseudo-sintéticas de vídeo controladas [Bla03], métodos sin usar “ground-truth” (Concepto que describe el número de elementos reales de los vídeos de test para así poder comparar con los resultados) [Wu04] o métodos donde el “ground-truth” se genera de forma manual [Doe00].

Ésto supone un problema inicial de evaluación por lo que será muy importante el escoger unas características que nos permitan ver de una forma clara si el sistema evaluado mejora o no respecto a otro. Además deberán facilitar el hecho de determinar los errores específicos del método.

5.2. Sistema de evaluación del sistema de tracking

Escoger el sistema de evaluación fue uno de los primeros puntos que se realizaron en el proyecto, incluso antes de la implementación del propio algoritmo. Ésto se consideró de vital importancia para así asegurar que la propia concepción que se pudiera tener del algoritmo no influyera en la selección del método de evaluación o del propio conjunto de test, y así hacer todo el proceso de evaluación mucho más fiable. Cabe decir que este sistema es utilizado únicamente en las dos primeras implementaciones y posteriormente se explicará el sistema utilizado para el testeo del tercer sistema, el cual se centra en aguantar el tracking sobre un objeto en movimiento que en un determinado momento se detiene en un punto, a pesar de continuar generando movimiento en una posición estática (girar sobre si mismo, hacer pequeños desplazamientos o gestos).

Para esta evaluación se han utilizado tres procedimientos diferentes. El primer procedimiento, totalmente manual, ha sido evaluado con un pequeño conjunto de vídeos. Este sistema dará muestras muy representativas del funcionamiento del tracking las cuales serían suficientes para el estudio que atañe a este proyecto. Sin embargo, su realización en empresa obliga a la utilización de métodos de evaluación automática. Esto se debe al elevado coste temporal que supone una evaluación manual y la inviabilidad por parte de la empresa de utilizar este sistema de evaluación para sus implementaciones. Por lo tanto, el segundo procedimiento será un método de evaluación automática en el que se evalúa el correcto funcionamiento del tracker. Finalmente se utilizará un último test global automático. Este último test no evalúa directamente el tracker, sino todo el sistema. Sin embargo, cambios en el modelo de tracker implican repercusiones en todo el sistema, por lo cual nos ofrece una información interesante aunque no decisiva.

Un objetivo importante de la utilización de tres tipos de técnicas de evaluación es compararlas entre sí y ver si nos ofrecen la misma información. De no ser así estudiar cual serían los motivos. Este estudio viene de la necesidad de saber con certeza si el sistema automático podría llegar a sustituir al manual.

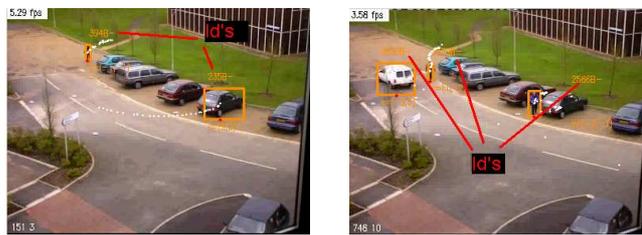


Figura 5.1: Salida para el test manual

5.2.1. Evaluación manual

Para llevar a cabo la evaluación manual se grabaron una serie de vídeos una vez pasados por el sistema de tracking, marcando cada objeto con el área que lo delimita y su identificador en el momento en el cual está siendo seguido (ver Figura. 5.1). Con este material se extraen principalmente dos medidas.

La primera es el tiempo medio de tracking de un objeto en escena, es decir, el número de frames en el cual este objeto esta siendo seguido, dividido por el número de frames que aparece en escena. Esta medida ayuda principalmente a ver la fiabilidad de el seguimiento del objeto y ver si hay situaciones en las que por alguna razón se pierde este tracking.

La segunda medida es el número de identificadores que recibe un mismo objeto durante el transcurso del vídeo. Esta medida permite apreciar si el seguimiento de un objeto en concreto es muy susceptible a pequeñas pérdidas de tracking. Por ejemplo un sistema con un alto porcentaje de tracking pero con un número elevado de identificadores podría suponer un problema para un posterior reconocimiento de ese objeto, ya que el sistema lo identificará como diferentes elementos durante el transcurso de la escena.

En cuanto al conjunto de test que se escogió para esta evaluación se compone de 17 vídeos (ver Cuadro. 5.1) con una duración total de aproximadamente nueve minutos, en los cuales aparecen un total de 49 elementos. Los vídeos son de baja calidad, es decir, con un nivel de compresión similar al que nos proporcionaría una cámara de vigilancia. Ésto se debe a que va a ser en estas condiciones, en las que se va a trabajar en un sistema real. Este conjunto esta escogido para representar

Nº	Vehículos	Personas	Duración	Cruces	Oclusiones	Iluminación	Origen	Frame
1	0	2	39"	Sí	No	Baja	Propio	
2	0	2	51"	Sí	No	Baja	Propio	
3	0	2	40"	Sí	No	Baja	Propio	
4	0	2	58"	Sí	No	Baja	Propio	
5	0	1	54"	No	No	Media	Davantis	
6	0	2	1' 50"	Sí	No	Media	Davantis	
7	1	0	58"	No	Parcial	Alta	Davantis	
8	6	0	1' 8"	Sí	Parcial	Media	Davantis	
9	2	0	50"	Sí	No	Baja	Davantis	
10	2	0	36"	Sí	Parcial	Media	Davantis	
11	2	0	20"	Sí	Parcial	Media	Davantis	
12	0	2	26"	Sí	Parcial	Alta	Davantis	
13	0	1	3' 16"	No	Sí	Alta	Davantis	
14	0	3	2'	Sí	Parcial	Media	Davantis	
15	0	1	45"	No	No	Media	Davantis	
16	3	3	1'	Sí	No	Alta	PETS	
17	3	9	1' 52"	Sí	No	Alta	PETS	

Cuadro 5.1: Conjunto de test, manual



Figura 5.2: Iluminación: Cambio brusco y escena con sombras



Figura 5.3: Posición: Objeto lejano, cruce y oclusión parcial

diversas situaciones adversas como lo son cambios de iluminación bruscos, sombras (ver Figura. 5.2), objetos lejanos, objetos pasando a gran velocidad, cruces y oclusiones (ver Figura. 5.3). La mayoría de éstos están compuestos por más de un elemento y son situaciones no controladas a excepción de cuatro de ellos grabados por mí, representando diferentes tipos de cruces. El resto de los vídeos fue obtenido de la base de datos PETS [Pets] o de la propia base de datos de Davantis.

5.2.2. Test automático de tracking

Este sistema utiliza un conjunto de vídeos donde intervienen un total de 108 elementos, 29 de los cuales pertenecen a vídeos sintéticos y el resto a vídeos reales. Estos vídeos están escogidos de manera que no haya elementos que puedan confundir al sistema de tracking, como son arboles moviéndose por el viento, sombras, etc. El sistema analizará los vídeos y hará un recuento del número de elementos detectados en cada vídeo, considerando cada nuevo identificador como un nuevo objeto. Finalmente el sistema retorna el número total de elementos detectados. Sabiendo el número de objetos iniciales, se puede extraer el número de

elementos detectados de más, a lo cual se le denomina “falsas alarmas”. Con estos datos se puede ver si el tracking es estable o si por lo contrario se pierde numerosas veces.

5.2.3. Tests automático global

Este sistema esta ideado como evaluación de un sistema de videovigilancia completo, y no de un sistema de tracking. Sin embargo, si el tracking de un sistema de este tipo mejora, también deberían verse repercusiones sobre la salida final del sistema (más detecciones correctas y reconocimientos). Es por esto que el uso de este sistema puede ofrecernos datos interesantes. Sin embargo, nunca los podríamos considerar como una medida decisiva para determinar las mejoras en un sistema de tracking independiente. Una importante motivación al usar este método es de cara a la empresa Davantis. Para que un módulo pueda integrarse al Daview, el sistema en el cual se integre dicho módulo debería ofrecer unos buenos resultados en este test.

Este sistema utiliza un conjunto de 6767 vídeos de entre uno y tres minutos de duración. Estos vídeos se etiquetan debidamente con los frames donde empiezan y acaban situaciones concretas, y el número y tipo de elementos que deberían ser detectados en cada una (ground-truth). El sistema se encargará de decir cuantos objetos detecta finalmente en cada situación, pero aquí a diferencia del anterior no tomará como objeto cada identificador, sino que contará los objetos realmente detectados como tales (evidentemente el número de identificadores influirá en esta detección en módulos posteriores).

Este sistema devuelve dos medidas: La “Precision”, que consiste en el porcentaje de objetos que han sido correctamente detectados entre todos los detectados, y el “Recall”, que se define como el porcentaje de objetos bien detectados respecto a todos los que deberían haber sido detectados.



Figura 5.4: Objeto en posición estática

5.3. Sistema de evaluación para objetos estáticos

Este método está ideado para la evaluación del último sistema de tracking, en el cual únicamente importa la eficacia de éste frente al problema de los objetos que detienen su movimiento durante el transcurso de la escena.

El método de evaluación es sencillo, y consiste en un sistema totalmente manual, en el cual se cronometra el tiempo que el sistema de tracking aguanta el seguimiento del objeto desde que éste se detiene. Para ello se realizaron grabaciones de un conjunto de cinco vídeos en diferentes situaciones pero con el factor común de intervenir una única persona que detiene su movimiento durante un determinado tiempo. Durante el transcurso de ese tiempo la persona no permanece inmóvil sino que se mueve ligeramente, rota sobre sí misma y realiza ligeros desplazamientos (ver Figura. 5.4). El conjunto de vídeos consta de una duración total de aproximadamente 19 minutos.

Otro punto importante para analizar este sistema es escoger algo con lo que contrastarlo, para ello se evaluó también una implementación del Daview basada en un método de correlación [Cro95] para situaciones en las que se detecta un objeto estático.

Capítulo 6

Resultados de evaluación

En este capítulo se presentan todos los resultados obtenidos durante el periodo de evaluación tanto de forma aislada para cada una de las implementaciones realizadas, como de forma comparativa entre ellas. También se exponen los resultados obtenidos al evaluar el propio sistema de tracking utilizado por Davantis. Por lo tanto se comenzará por presentar los resultados manuales y automáticos del tracking de Davantis y de cada una de las dos primeras implementaciones de tracking, así como una pequeña discusión de cada método utilizando ambos modelos de test. Posteriormente se expondrán de forma comparativa los tres métodos analizados. Finalmente, la última sección está dedicada a la evaluación del tracking de objetos estáticos para los sistemas mononúcleo y de correlación.

6.1. Modelo de Tracking de Davantis

En esta sección se presentan los resultados obtenidos con el sistema de tracking utilizado por Davantis en su programa, el Daview, los cuales fueron obtenidos al inicio del proyecto.

Para la **evaluación manual** se expone una tabla de resultados que incluye el porcentaje de tracking y el número de identificadores medio para cada uno de los 17 vídeos sobre los que se ha realizado el test, así como una serie de atributos que definen el estado del vídeo entre los que se incluyen características espaciales y

N°	Objetos	% Tracking	Id's	Cruces	Oclusiones	Iluminación
1	2	92	2.5	Sí	No	Baja
2	2	92	2	Sí	No	Baja
3	2	93	1	Sí	No	Baja
4	2	83	1.5	Sí	No	Baja
5	1	88	2	No	No	Media
6	2	87	2.5	Sí	No	Media
7	1	69	3	No	Parcial	Alta
8	6	88	1	Sí	Parcial	Media
9	2	76	2	Sí	No	Baja
10	2	80	1,5	Sí	Parcial	Media
11	2	70	2	Sí	Parcial	Media
12	2	81	1.5	Sí	Parcial	Alta
13	1	87	2	No	Sí	Alta
14	3	81	1.3	Sí	Parcial	Media
15	1	38	3	No	No	Media
16	6	86	1.2	Sí	No	Alta
17	12	94	1.4	Sí	No	Alta

Cuadro 6.1: Resultados manuales, Daview

Métrica	Valor Medio
% de Tracking	81,47
N° de Identificadores	1.8

Cuadro 6.2: Medias de métricas manuales, Daview

fotométricas (ver Cuadro. 6.1). Esta información también se puede ver de forma más visual mediante las gráficas adjuntas para el porcentaje de tracking y el número medio de identificadores (ver Figura. 6.1), donde cada columna es el valor para cada uno de los vídeos tratados.

Extrayendo métricas generales de los resultados (ver Cuadro. 6.2) se puede observar que el porcentaje de tracking se mantiene superior al 80%. Considerando las diferentes situaciones adversas de los videos se puede considerar este valor muy elevado. Este valor se complementa con un número medio de identificadores por objeto inferior a dos. Estos datos indican que es un modelo de tracking muy robusto el cual sigue al objeto durante la mayor parte del vídeo manteniendo

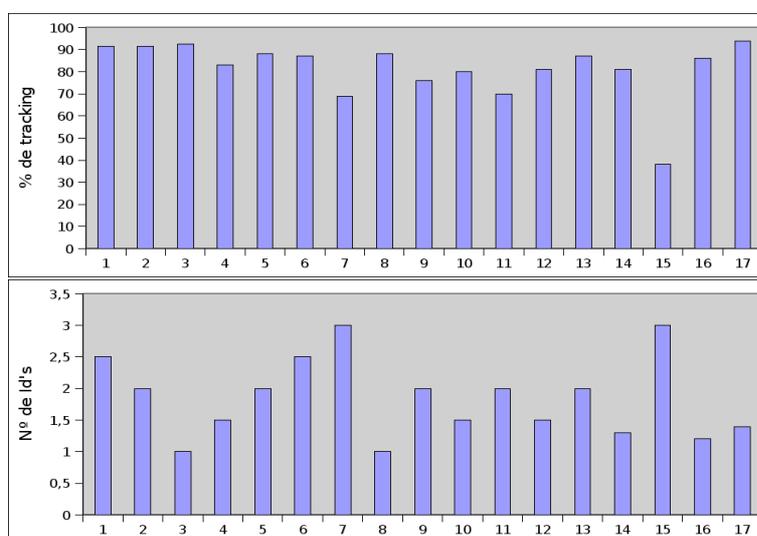


Figura 6.1: Porcentaje de tracking y N_o medio de id's, Daview

normalmente su identificador, salvo en algunas ocasiones de pérdida de tracking, cruces u oclusiones.

En cuanto a la **Evaluación automática**, el primer resultado que obtenemos es el del **Test de Tracking**. Éste se compone de pocos vídeos y únicamente nos da el número de falsas alarmas que tiene nuestro sistema. Este valor por si mismo no nos ofrece mucha información pero nos será útil durante la comparación de los tres sistemas.

- Número total de Falsas Alarmas: 283

En cuanto al segundo tipo de test, el **Test Global**, viene compuesto principalmente de dos medidas. La “precision” que ofrece el porcentaje de objetos bien detectados dentro del total de objetos detectados y el “recall” que ofrece el porcentaje de objetos bien detectados respecto a todos los que deberían haber sido detectados. Además también se exponen medidas del número de falsas alarmas y del número de objetos no detectados. Hay que tener en cuenta que este test evalúa el sistema de videovigilancia completo, no únicamente el tracking.

- Precision: 70,22 %

- Recall: 89,71 %
- Número de objetos no detectados: 483
- Número de falsas alarmas: 985

Interpretando estos resultados, los cuales tomaremos como punto de referencia durante esta evaluación, podríamos decir que el único punto donde el sistema muestra una cierta debilidad es en el número de falsas alarmas, provocado posiblemente por la asignación de múltiples identificadores a un mismo objeto. Tal y como se ha visto en la evaluación manual este valor no es muy elevado, pero si superior al ideal. Esto se ve reflejado en el test global, donde se puede apreciar un valor de precision bastante bajo, posiblemente debido al número de falsas alarmas provocado por el módulo de tracking.

6.2. Mean Shift multinúcleo

A continuación se presentan los resultados de la primera implementación realizada, es decir un sistema de Mean Shift multinúcleo, evaluada con la metodología escogida para un sistema de tracking independiente, explicada en el capítulo anterior.

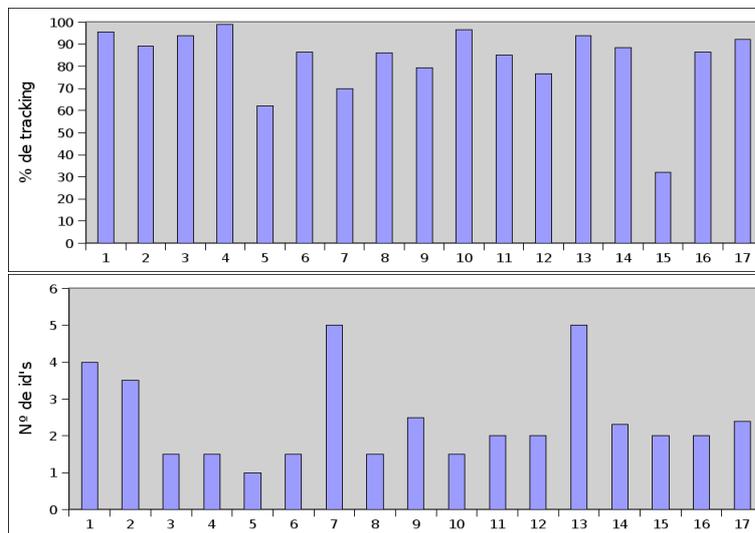
Para la **evaluación manual** se presentan el mismo tipo de resultados que en el método anterior. Lo cual incluye la tabla de resultados (ver Cuadro. 6.3), y sus gráficas (ver Figura. 6.2).

Mirando las métricas generales (ver Cuadro. 6.4) podemos ver que mantiene un porcentaje de tracking elevado, similar al del modelo de Davantis. Sin embargo el número medio de identificadores aumenta considerablemente, esto se debe en muchas ocasiones a pequeños cambios de identificador durante poco tiempo (2 o 3 segundos) para volver al original, pero cabe destacar que en un gran número de casos si se ignoraran estas pequeñas pérdidas, el objeto mantendría un mismo identificador durante la mayor parte de la escena, por lo tanto no se tomará este dato como algo excesivamente negativo.

En cuanto a la **evaluación automática** obtenemos los siguientes resultados:

N^o	Objetos	% Tracking	Id's	Cruces	Oclusiones	Iluminación
1	2	94	4	Sí	No	Baja
2	2	89	3.5	Sí	No	Baja
3	2	94	1.5	Sí	No	Baja
4	2	99	1.5	Sí	No	Baja
5	1	62	1	No	No	Media
6	2	86	1.5	Sí	No	Media
7	1	70	5	No	Parcial	Alta
8	6	86	1.5	Sí	Parcial	Media
9	2	79	2.5	Sí	No	Baja
10	2	96	1.5	Sí	Parcial	Media
11	2	85	2	Sí	Parcial	Media
12	2	76	2	Sí	Parcial	Alta
13	1	94	5	No	Sí	Alta
14	3	88	2.3	Sí	Parcial	Media
15	1	32	2	No	No	Media
16	6	86	2	Sí	No	Alta
17	12	92	2.4	Sí	No	Alta

Cuadro 6.3: Resultados manuales, Mean Shift multinúcleo

Figura 6.2: Porcentaje de tracking y N_o medio de id's, Mean Shift multinúcleo

Métrica	Valor Medio
% de Tracking	82.82
Nº de Identificadores	2.42

Cuadro 6.4: Medias de métricas manuales, Mean Shift multinúcleo

Test de tracking

- Número total de Falsas Alarmas: 387

A simple vista mediante estos resultados se puede observar que este sistema nos da un gran número de falsas alarmas comparado con el anterior.

Test global

- Precision: 60,89 %
- Recall: 86,34 %
- Número de objetos no detectados: 614
- Número de falsas alarmas: 1152

Este método destaca sobre todo, por su elevado número de identificadores por objeto, lo cual se ha visto confirmado en el test automático con un número de falsas alarmas superior al modelo de davantis. Esto se refleja en el test global con un gran descenso del precision.

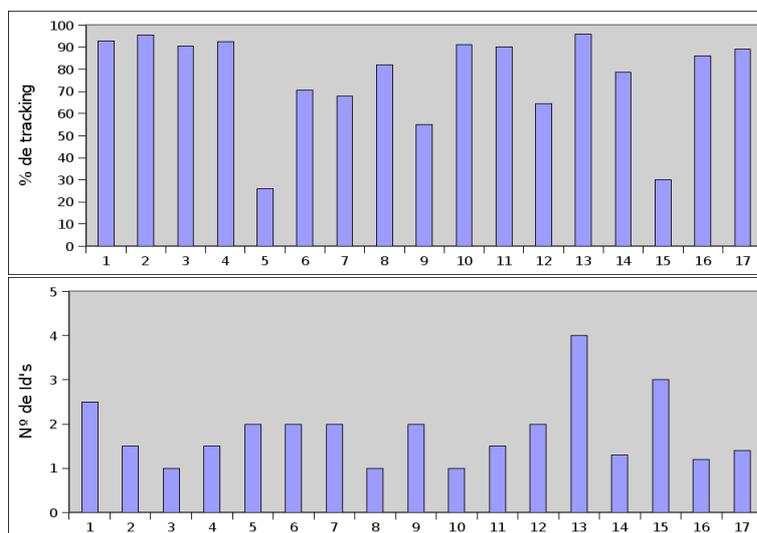
6.3. Mean Shift complementado

A continuación se presentan los resultados de la segunda implementación realizada. Evaluada con la metodología escogida para la evaluación de un sistema de tracking independiente, explicada en el capítulo anterior.

Para la **evaluación manual** se presentan el mismo tipo de resultados que en el método anterior. Lo cual incluye la tabla de resultados (ver Cuadro. 6.5), y sus gráficas (ver Figura. 6.3).

N°	Objetos	% Tracking	Id's	Cruces	Oclusiones	Iluminación
1	2	93	2.5	Sí	No	Baja
2	2	95	1.5	Sí	No	Baja
3	2	90	1	Sí	No	Baja
4	2	92	1.5	Sí	No	Baja
5	1	26	2	No	No	Media
6	2	70	2	Sí	No	Media
7	1	68	2	No	Parcial	Alta
8	6	82	1	Sí	Parcial	Media
9	2	55	2	Sí	No	Baja
10	2	91	1	Sí	Parcial	Media
11	2	90	1.5	Sí	Parcial	Media
12	2	65	2	Sí	Parcial	Alta
13	1	96	4	No	Sí	Alta
14	3	79	1.3	Sí	Parcial	Media
15	1	30	3	No	No	Media
16	6	86	1.2	Sí	No	Alta
17	12	89	1.4	Sí	No	Alta

Cuadro 6.5: Resultados manuales, Mean Shift complementado

Figura 6.3: Porcentaje de tracking y N_o medio de id's, Mean Shift complementado

Métrica	Valor Medio
% de Tracking	76.29
Nº de Identificadores	1.81

Cuadro 6.6: Medias de métricas manuales, Mean Shift complementado

Mirando las métricas generales (ver Cuadro. 6.6) se aprecia un porcentaje de tracking inferior al de los dos métodos anteriores y un número medio de identificadores inferior a dos, por lo tanto similar al tracker de Davantis en este sentido.

En cuanto a la **evaluación automática** se obtienen los siguientes resultados:

Test de tracking

- Número total de Falsas Alarmas: 166

Viendo estos resultados aparentemente parece que el número de falsas alarmas disminuye considerablemente respecto a los dos métodos anteriores.

Test global

- Precision: 75,88 %
- Recall: 84,31 %
- Número de objetos no detectados: 742
- Número de falsas alarmas: 855

Confirmando los resultados anteriores se ve que el número de falsas alarmas es inferior al resto. Sin embargo este tracking ha hecho disminuir las detecciones del sistema general. Ésto puede ser provocado por el bajo porcentaje de tracking que se aprecia en la evaluación manual.

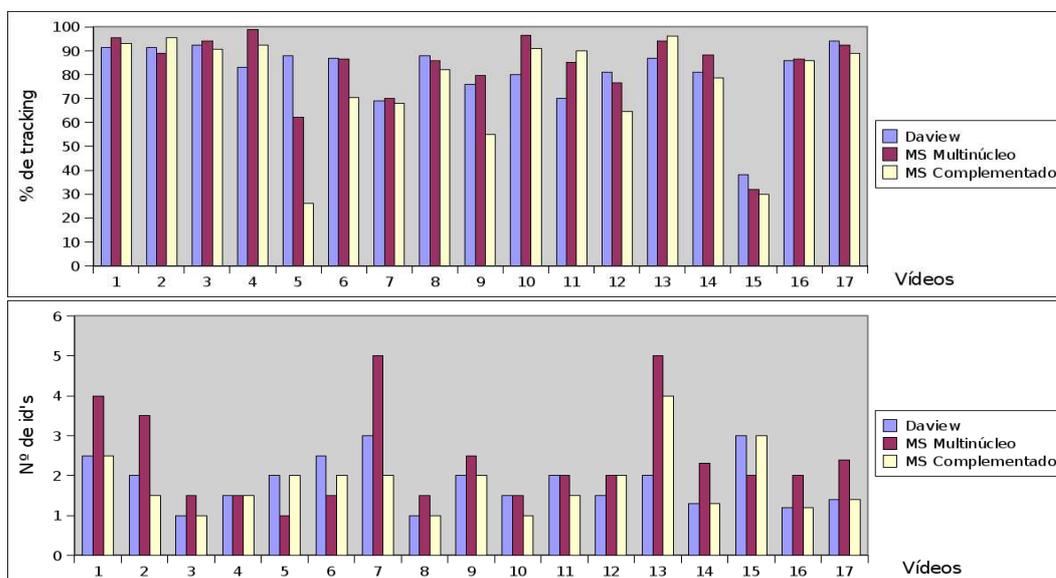


Figura 6.4: Porcentaje de tracking y N_o medio de id's, Comparativa

Modelo	% de tracking	N^o medio de id's	FA tracking
Daview	81.47 %	1.80	283
multinúcleo	82.82 %	2.42	387
combinado	76.29 %	1.81	166

Cuadro 6.7: Resultados de los tests automáticos

6.4. Comparativa

En este apartado se hace una comparativa de los tres métodos de los cuales se acaban de presentar resultados. Para comenzar se presentan unas gráficas comparativas de dichos métodos según la evaluación manual (Figura. 6.4) donde cada unidad corresponde a un vídeo.

También se presenta una tabla comparativa de los resultados manuales y del test automático de tracking (Ver Cuadro. 6.7).

Observando estos resultados, en cuanto a porcentaje de tracking se pueden ver similitudes entre los tres métodos a excepción de algunos vídeos concretos como el **Vídeo 5** o el **Vídeo 9**, donde el tracking del complementado baja considera-

blemente. Estos vídeos destacan por ser vídeos con muchas sombras donde se generan muchas regiones de movimiento. El hecho de que el modelo combinado baje aquí su efectividad indica, posiblemente, un problema de procesamiento. Es decir que el método sea computacionalmente demasiado costoso para este tipo de escenas. De todas formas, en general también se aprecia que este tracking es el que peores resultados de porcentaje de tracking obtiene.

En cuanto al número de identificadores, ninguno de los modelos superan al tracker del Daview. Destaca el elevado número del tracker multinúcleo, provocado por los diversos cambios de identificador producidos en breves períodos de tiempo.

En cuanto al número de falsas alarmas generadas por el test de tracking, se puede ver que el modelo multinúcleo cae drásticamente. Sin embargo se observa una mejora considerable en el tracking combinado. Esta mejora no es del todo fiable debido a su porcentaje de tracking, inferior al del resto.

En resumen, podemos apreciar que el tracker mononúcleo nos ofrece datos interesantes en cuanto a mantenimiento final del identificador. Sin embargo produce una serie de lapsus en sus identificadores durante el recorrido, posiblemente provocados por encontrar objetos similares en malas condiciones de iluminación. Ésto hace que sus resultados bajen drásticamente y no sea apto para sustituir el sistema actual de tracking. En cuanto al modelo combinado ofrece muy buenos resultados en el test automático, pero en cambio se detecta un bajo porcentaje de tracking en el test manual, por lo que parece ser que estos resultados no son del todo fiables (si se sigue al objeto menos tiempo, puede ser normal que tenga menos falsas alarmas). Este bajo porcentaje se produce por el elevado coste computacional, por lo cual el sistema podría mejorar mucho para implementaciones que no sean a tiempo real. Este segundo sistema tampoco sería una opción para sustituir al actual.

Si observamos los el **Test global** automático obtenemos los siguientes resultados (ver Cuadro. 6.8).

Dando un repaso a los dos métodos implementados, podemos observar que el método multinúcleo desencadena en la salida final del sistema una caída del

Modelo	Precision	Recall	FA general	No detectados
Daview	70,22 %	89,71 %	985	483
multinúcleo	60,89 %	86,34 %	1152	614
combinado	75,88 %	84,31 %	855	742

Cuadro 6.8: Resultados de los tests automáticos

precision. Ésto se podía prever viendo el gran número de falsas alarmas que obteníamos en los métodos anteriores. Por lo tanto resulta normal que el número de objetos final bien detectados sobre todos los detectados aumente, sobre todo por el aumento de los objetos mal detectados. Esto lo confirma un recall bastante elevado.

En cuanto al modelo combinado, en cambio, observamos que la salida del sistema refleja los mismos datos que el test de tracking. Observamos que mejora el precision, debido a la reducción de identificadores en el tracking. Sin embargo el bajo porcentaje de tracking hace caer el recall.

Durante esta evaluación se han utilizado diversos **modelos de test**. El test manual nos ha servido para ver la efectividad del tracker así como para la detección de los problemas de éste. En cuanto al test de tracking nos ha confirmado y ayudado a interpretar los resultados manuales. Sin embargo no nos ha aportado información decisiva. En cambio mirandolo al revés, si únicamente se tuviera la información del test automático no se podrían apreciar los problemas reales del tracker, ni tener unos datos fiables (como en el caso del modelo combinado). Por lo tanto el test manual no es sustituible por el automático, ya que el primero aporta más información.

6.5. Objetos estáticos

Para este apartado únicamente se valorará el tiempo que cada sistema es capaz de aguantar el objeto, el cual se muestra en forma de porcentaje para cada uno de los cinco vídeos de los que se compone este test, tanto para el Mean Shift mononúcleo (ver Cuadro. 6.9) como para el tracking con correlación (ver Cuadro.

Vídeo	% de Tracking	Duración	Distancia	Iluminación
1	100	5' 55"	Cercano	Alta
2	97	3' 15"	Medio	Alta
3	100	6' 23"	Lejano	Alta
4	99	1' 40"	Medio	Media
5	70	1' 44"	Lejano	Baja

Cuadro 6.9: Tiempo de tracking, Mean Shift mononúcleo

Vídeo	% de Tracking	Duración	Distancia	Iluminación
1	2	5' 55"	Cercano	Alta
2	4.7	3' 15"	Medio	Alta
3	1.7	6' 23"	Lejano	Alta
4	50	1' 40"	Medio	Media
5	10	1' 44"	Lejano	Baja

Cuadro 6.10: Tiempo de tracking, Tracking con Correlación

6.10).

6.5.1. Comparativa

Viendo los resultados (Figura. 6.5), se observa que el sistema de Mean Shift aguanta mucho mejor la posición que el sistema con correlación.

Se podría decir pues que el sistema de Mean Shift es mucho mejor que el de correlación pero sería una afirmación demasiado precipitada, cabe tener en cuenta que las pruebas han sido realizadas en el caso del Mean Shift con un sistema que únicamente se basa en el uso de este método. Sin embargo, en el caso de la correlación en un sistema que únicamente lo utiliza para objetos que reconoce como estáticos.

Hay que destacar que el sistema de Mean Shift ha aguantado el tracking del objeto con una eficacia cercana al 100 % a excepción de algunos casos, **Video 5**, donde el porcentaje ha bajado debido a ser una escena con poca iluminación.

Por lo tanto se podría decir que aun a pesar de los inconvenientes que acabo de mencionar, el sistema de Mean Shift puede aguantar mejor un objeto que sufre

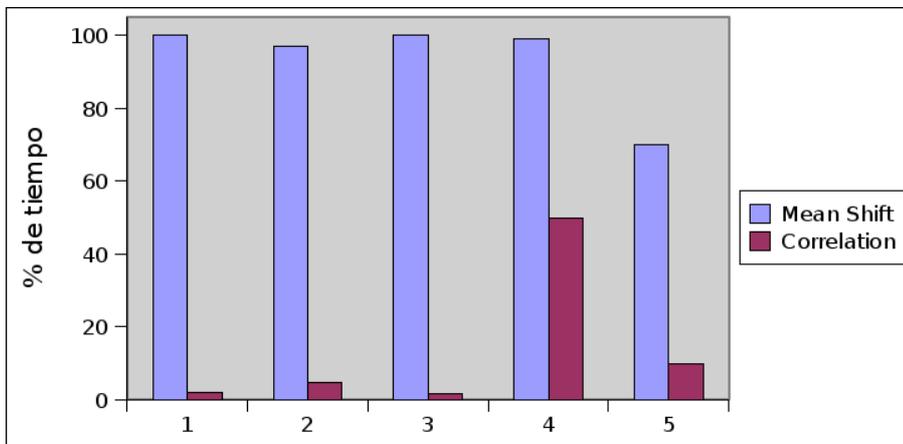


Figura 6.5: Tiempo de tracking, Comparativa

cambios mientras permanece en una posición estática. Por lo cual podría ser una excelente medida complementaria para un sistema de tracking compuesto de más características. Sin embargo esta eficacia es variante en cuanto a la iluminación de la escena, problema que podría llegar a ser estudiado para intentar encontrar alguna normalización fotométrica de la escena factible a tiempo real, con lo cual el método podría mejorar en este punto.

En cuanto al tiempo de procesamiento, a pesar de no extraer métricas exactas se puede decir que ambos métodos suponen un coste computacional elevado, donde predominaría la correlación pero siendo una diferencia mínima que no supondría un factor decisivo a la hora de escoger entre uno de los dos métodos.

Capítulo 7

Conclusiones

7.1. Valoración del método

En esta sección, se dará una visión global sobre la utilidad del método Mean Shift en la videovigilancia, basándose en los resultados que se han analizado en el capítulo anterior. Si se considera el método en su forma típica, como sistema auto-suficiente de tracking, se encuentran con muchos problemas al evaluar situaciones a bajo frame-rate, o con dificultades de iluminación. Introduciéndole adaptaciones, para hacer que el método utilice una búsqueda multinúcleo, y adoptando la información del background, se solucionan muchos de estos problemas. Sin embargo, sigue siendo muy susceptible a cambios bruscos de información, o frente a objetos similares en la misma zona. Cuando se encuentran más beneficios en este método, es al combinarlo con otras métricas, ya que nos proporciona una información adicional muy útil a la hora de escoger, además el punto más fuerte del método, es el de mantener el tracking de un objeto a pesar de que éste se detenga, que es precisamente donde las técnicas basadas en movimiento fallan más a menudo.

En cuanto a ambos métodos de evaluación, se complementan bien, pero ninguno de los dos sustituye completamente al otro. Gracias a la evaluación automática, se han podido extraer las métricas sobre que método fallaba más o menos en cuanto al número de objetos detectados, pero ha sido gracias al método manual

que se han podido extraer conclusiones del porqué de esos valores. El método manual, ha dado además información sobre la estabilidad del tracker siguiendo un objeto, métrica que no se puede extraer mediante el método automático. Se podría decir entonces, que el método manual ofrece una información, que no puede ser substituida por el método automático.

7.2. Valoración de la implementación

Este estudio, tenía como objetivo el de encontrar posibles mejoras frente al modelo de Davantis. Tal y como se ha visto en el capítulo anterior, ninguno de los métodos implementados supone una mejora notable como para ser sustituido por el propio tracking del Daview. Sin embargo, en cada una de ellas se ha observado algún dato interesante.

En el modelo de Mean Shift multinúcleo, a pesar de perder el tracking en diversas situaciones, el identificador final del objeto coincidía reiteradas veces con el de su inicio. Ésto hace plantearse este método como una posible mejora, frustrada por el gran número de falsas alarmas que produce. Una posible nueva línea de investigación, sería la de intentar, mediante algún otro método, reducir los constantes cambios de identificador producidos.

En cuanto al modelo de Mean Shift combinado, finalmente dio unos valores de precisión más elevados que los del propio Daview. El punto negativo, sería un nivel de detecciones inferior al del método actual. Sería interesante explorar en esta línea, para averiguar si a un mismo nivel de detecciones, se mantendría o no, dicha precisión. Otro punto negativo de este método, es un coste computacional superior al del resto, lo cual, si no se mejora, dificultaría su adaptación al sistema real.

Finalmente, la aplicación del Mean Shift clásico (mononúcleo), pero utilizando información del background, ha sido donde quizás se han extraído unas métricas más interesantes. Este método, como tracker individual, presenta diversos problemas. Sin embargo, se han obtenido buenas métricas en uno de los puntos más conflictivos de las técnicas utilizadas por el Daview. Cuando una persona en

movimiento se detiene, se deja de tener información acerca de su movimiento, por lo cual, cuesta de mantener el tracking de dicha persona con un método que no sea basado en su apariencia. Si la persona se mantuviera estática, métodos como la correlación funcionarían satisfactoriamente. Pero una persona, no suele permanecer totalmente estática, sino que se mueve ligeramente, o rota sobre su eje. Ante estos casos el método del Mean Shift, ha dado mejores resultados que la correlación, por lo cual podría ser interesante la implementación del método, para ser aplicada a objetos que detengan su movimiento durante la escena.

7.3. Valoración de la planificación inicial

En cuanto a la planificación inicial del proyecto, se podría decir, que se han asumido los objetivos básicos de éste, determinando los puntos, donde el Mean Shift podía ayudar a mejorar al sistema de tracking del Daview. Durante el desarrollo del primer semestre, la planificación temporal se cumplió correctamente mediante la documentación y decisión del método a utilizar, así como la elección de un sistema de evaluación manual y de su conjunto de test. Durante la segunda etapa del proyecto, finalmente se desarrollaron tres implementaciones diferentes, por lo cual, la fase de implementación se prolongó hasta pocas semanas antes de la finalización del proyecto. Este hecho obligó a trabajar en paralelo, en la fase de implementación y la de test de los métodos ya implementados, y finalmente también compartir la fase de redacción de la memoria, con los últimos tests.

Bibliografía

- [Bal82] D. Ballard y C. Brown, “Computer Vision”, Prentice-Hall, 1982.
- [Bla98] M. Black and A. Jepson, “Eigentracking: Robust matching and tracking of articulated objects using a view-based representation”, *Int. J. Comput. Vision*, vol. 25, No 1, 1998.
- [Bla03] J. Black, T. Ellis y P. Rosin, “A Novel Method for Video Tracking Performance Evaluation”, In *International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance 2003*.
- [Col03] R. T. Collins, “Mean-shift Tracking through Scale Space”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, 2003
- [Col05] R. Collins, Y. Liu y M. Leordeanu, “Online Selection of Discriminate Tracking Features”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, No. 10, Octubre 2005
- [Cole04] Luke Cole, D. Austin y Lance Cole, “Visual Object Recognition using Template Matching”, *Australasian Conference on Robotics and Automation*, 2004
- [Com02] D. Comaniciu y P. Meer, “A Robust Approach Toward Feature Space Analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, No. 5, Mayo 2002.

- [Com03] D. Comaniciu, V. Ramesh y P. Meer, “Kernel-Based Object Tracking”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, No. 5, Mayo 2003.
- [Coo01] T. Cootes, G. Edwards y C. Taylor, “Robust real-time periodic motion detection, analysis, and applications”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, No. 6, 2001.
- [Cro95] J. L. Crowley y J. Martin “Experimental Comparison of Correlation Techniques”, *Conference on Intelligent Autonomous Systems*, 1995.
- [Cvc] Computer Vision Center
<<http://www.cvc.uab.es/>>
- [Dav] Davantis
<<http://www.davantis.com/>>
- [Doe00] D. Doermann y D. Mihalcik ”Tools and Techniques for Video Performance Evaluation “, *5th International Conference on Pattern Recognition*, Vol. 4, 2000
- [Edw98] G. Edwards, C. Taylor y T. Cootes, “Interpreting face images using active appearance models”, *International Conference on Face and Gesture Recognition*, 1998.
- [Elg02] A. Elgammal, R. Duraiswami, D. Harwood y L. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance”, *Proceedings of IEEE*, vol. 90, No. 7, 2002.
- [Fie97] P. Fieguth y D. Terzopoulos, “Color-based tracking of heads and other mobile objects at video frame rates”, *IEEE International Conference of Pattern Recognition, (CVPR)*, 1997.
- [Har88] C. Harris y M. Stephens, “A combined corner and edge detector”, *4th Alvey Vision Conference*, 1988.

- [Isa98] M. Isard y A. Blake, “CONDENSATION-Conditional Density Propagation for Visual Tracking”, *International Journal of Computer Vision*, vol. 29 No. 1, Agosto 1998.
- [Jai79] R. Jain y H. Nagel, “On the analysis of accumulative difference pictures from image sequence of the real world scenes”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, No. 1, 1979.
- [Kas88] M. Kass, A. Witkin y D. Terzopoulos, “snakes: active contour models”, *Int. J. Comput. Vision* 1, 1988
- [Kuh55] H. Kuhn, “The hungarian method for solving the assignment problem”, *Naval Research Logistics Quart*, 1955.
- [Low04] D. Lowe, “Distinctive image features from scale-invariant keypoints”, *International J. Comput. Vision*, vol. 60, No. 2, 2004
- [Mor79] H. Moravec, “Visual mapping by a robot rover”, *Proceedings of the International Joint Conference on Artificial Intelligence, (IJCAI)*, 1979.
- [Mug97] B. Mughadam y A. Pentland, “Probabilistic visual learning for object representation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, No. 7, 1997.
- [OpCv] OpenCv
<<http://www.intel.com/technology/computing/opencv/>>
- [Pap98] C. Papageorgiou, M. Oren y T. Poggio, “A general framework for object detection”, *IEEE International Conference of Computer Vision (ICCV)*, 1998.
- [Pets] Pets Dataset
<<http://peipa.essex.ac.uk/ipa/pix/pets/>>

- [Por05] F. Porikli y O. Tuzel, “Object Tracking in Low-Frame-Rate Video”, Mitsubishi Research Laboratories, TR2005-013, Marzo 2005
<<http://www.merl.com/>>
- [Row00] H. Rowley, S. Baluja y T. Kanade, “Neuronal network-based face detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, No. 1, 2000.
- [Sch02] H. Schweitzer , J. W. Bell y F. Wu, “Very Fast Template Matching”, Proceedings of the 7th European Conference on Computer Vision-Part IV, Mayo 2002.
- [Ser04] D. Serby, S. Koller-Meier y L. V. Gool, “Probabilistic object tracking using multiple features”, IEEE International Conference of Pattern Recognition, (ICPR), 2004.
- [Shi94] J. Shi y C. Tomasi, “Good features to track”, International Conference on Face and Gesture Recognition (CVPR), 1994.
- [Shi00] J. Shi y J. Malik, “Normalized cuts and image segmentation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, No. 8, 2000.
- [Sh105] J. Shlens, “ Tutorial on Principal Component Analysis”, Systems Neurobiology Laboratory, 2005
- [Sta00] C. Stauffer y W. Grimson, “Learning patterns of activity using real time tracking”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8.
- [Uab] Universitat Autònoma de Barcelona
<<http://www.uab.es/>>
- [Vee01] C. Veenman, M. Reinders y E. Backer, “Resolving motion correspondence for densely moving points”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, No. 1, 2001.

- [Vio03] P. Viola, M. Jones y D. Snow, “Detecting pedestrians using patterns of motion and appearance”, IEEE International Conference of Computer Vision (ICCV), 2003.
- [Wu04] H. Wu y Q. Zheng, “Self-Evaluation for Video Tracking Systems”, International Multimedia Conference, Diciembre 2004
- [Yil04] A. Yilmaz, X. Li y M. Shah, “Contour based object tracking with occlusion handling in video acquired using mobile cameras”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, No. 11, 2004.
- [Yil06] A. Yilmaz, O.m Javed y M. Shah, “Object Tracking: A Survey”, ACM Computing Surveys, vol. 38, No. 4, Article. 13, Diciembre 2006
- [Zhu96] S. Zhu y A. Yuille, “Region competition: unifying snakes, region growing, and bayes/mdl for multi-band image segmentation.”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, No. 9, 1996.

Firmado: Javier Jubany Garrido
Bellaterra, 16 Junio de 2008

Resumen

Este proyecto, desarrollado en la empresa Davantis, tiene como objetivo encontrar posibles mejoras a su actual sistema de videovigilancia, el Daview. El proyecto está dedicado al estudio del algoritmo de seguimiento Mean Shift para la elaboración de un sistema de tracking. Para ello se han desarrollado y evaluado tres implementaciones diferentes, mediante las cuales se han encontrado mejoras que complementan al módulo de tracking del Daview. También se ha estudiado la utilidad de un sistema de evaluación manual frente a uno de automático.

Resum

Aquest projecte, desenvolupat a l'empresa Davantis, té com a objectiu trobar possibles millores al seu actual sistema de videovigilància, el Daview. El projecte esta dedicat a l'estudi de l'algoritme de seguiment Mean Shift per l'elaboració de un sistema de tracking. Per fer això, s'han desenvolupat i avaluat tres implementacions diferents, amb les quals s'han trobat millores que complementen el mòdul de tracking del Daview. També s'ha estudiat la utilitat d'un sistema d'avaluació manual en front d'un d'automàtic.

Abstract

This Project, developed by Davantis Company, aims to achieve improvements on its present video surveillance system, the Daview. The project is committed to the study of the Mean Shift tracking algorithm, in order to develop a tracking system. For this purpose it has been developed and evaluated three different implementations, which have lead to improvements on the tracking module of Daview. It also contains a utility study of a manual evaluation system opposite to an automatic one.