

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA INDUSTRIAL



PROYECTO FIN DE CARRERA

*TRACKING DE PERSONAS
A PARTIR DE
VISIÓN ARTIFICIAL*

Autor: JAVIER YÁÑEZ GARCÍA
Tutor: DR. LUIS MORENO LORENTE

SEPTIEMBRE DE 2010

A mi abuelo, por hacerme la persona que soy.

Agradecimientos

A mi tutor: D. Luis Moreno Lorente, por brindarme la oportunidad de realizar este proyecto a su lado y prestarme en todo momento el tiempo y los conocimientos necesarios para sacarlo adelante.

A mis padres, por estar siempre a mi lado y no dejar de creer en mí en ningún momento. También debo darles las gracias por ayudarme a convertir todos mis sueños en realidad.

A mi hermana, por ayudarme en todos los momentos difíciles y hacerme ver el mundo de otra manera.

A mi novia y mejor amiga Patricia, por llenar mi vida y hacerme sentir el chico más afortunado del mundo durante todos estos años.

A mi compañero de clase, de proyecto, de aventuras... en definitiva a mi amigo Ubaldo González Benítez, gracias al cual estos años han estado lleno de momentos estupendos e inolvidables.

A mis amigos de la infancia y de instituto, por hacerme sentir tan querido y no olvidarse de mí en todos estos años.

A mis compañeros de residencia y piso, y a mis nuevos amigos de universidad, que han convertido Madrid en mi segundo hogar.

A los compañeros de laboratorio, en especial a Jorge, Piotr, Ale y Fran, por ayudarme en todo momento y convertir el laboratorio en un lugar agradable donde el trabajo se hace más ameno.

Al Departamento de Ingeniería de Sistemas y Automática de la Universidad Carlos III, por brindarme las instalaciones para poder desarrollar este proyecto.

Y por último, a todos aquellos profesores que han contribuido en mi formación y han hecho que pueda cumplir un sueño, ser Ingeniero Industrial.

Gracias a todos, de corazón.

Javier Yáñez García.

ÍNDICE GENERAL

1. Introducción	13
2. Estado del arte	21
3. Arquitectura Hardware y Software	25
3.1. Sistema operativo	25
3.2. Biblioteca OpenCV 2.1.0	27
3.3. Librería cvblobslib	29
3.4. Cámara: Logitech QuickCam Pro 9000	29
4. Arquitectura funcional	31
4.1. Tracking de personas	32
4.1.1. Estudio de los distintos algoritmos existentes	33

4.1.2. Discusión	51
4.1.3. Presentación del algoritmo desarrollado	53
4.2. Detección de personas estáticas	56
4.3. Grabación de vídeo	59
5. Experimentación	61
5.1. Experimentación de la mejora: Tracking de personas	62
5.1.1. Seguimiento de una persona sola	62
5.1.2. Seguimiento de dos personas sin cruzarse	64
5.1.3. Seguimiento de dos personas cruzándose	67
5.1.4. Seguimiento de dos personas caminando pegadas	69
5.2. Experimentación de la mejora: Detección de personas estáticas	74
5.2.1. Persona permanece parada de pie o acostada en el interior de un edificio	74
5.2.2. Persona permanece parada de pie o acostada en el exterior	75
5.2.3. Persona permanece parada sentada en el interior de una habitación	75
5.2.4. Varias personas permanecen paradas a la vez	76
6. Conclusiones	79
7. Trabajo futuro	83

ÍNDICE DE FIGURAS

1.1. Relación de la visión por computador y otras áreas afines. . . .	14
1.2. Diagrama de flujo de la aplicación desarrollada en [4].	17
1.3. Ejemplo de fijación de una persona cuando esta permanece estática en la escena.	18
1.4. Ejemplo de tracking de personas.	19
2.1. Esquema taxonómico general de la clasificación de los métodos de seguimiento.	22
3.1. Paso 3 de la instalación de OpenCV 2.1.0	28
3.2. Paso 4 de la instalación de OpenCV 2.1.0	28
3.3. Paso 5 de la instalación de OpenCV 2.1.0	28
3.4. Paso 6 de la instalación de OpenCV 2.1.0	28
3.5. Paso 7 de la instalación de OpenCV 2.1.0	28

3.6. Webcam Logitech QuickCam Pro 9000.	30
4.1. Esquema de la estructura del capítulo.	32
4.2. Clasificación detallada de los algoritmos existentes para el tracking de personas.	33
4.3. Figura esquelética.	35
4.4. Modelo de contorno 2-D.	35
4.5. Modelo de contorno 2-D.	36
4.6. Modelos volumétricos.	36
4.7. Modelos híbridos.	37
4.8. El ciclo del algoritmo de Kalman.	44
4.9. Visión completa del filtro de Kalman.	47
4.10. Muestreo y pesos obtenidos al aplicar el filtro de partículas. . .	49
4.11. Funcionamiento del filtro de partículas.	49
4.12. Diagrama de flujo de la mejora de seguimiento de personas. . .	53
4.13. Pantalla de salida de la aplicación ante una situación en la que se dispone de toda la información.	55
4.14. Pantalla de salida de la aplicación ante una situación en la que hay ausencia de información.	55
4.15. Diagrama de flujo de la mejora de detección de personas estáticas. .	56
4.16. Detección de persona estática. Se puede observar como gracias a el módulo que aquí se describe la persona está detectada incluso cuando ya apenas aparece en el foreground y casi por completo en el background.	57

4.17. Persona estática se vuelve a mover. Demostración de como cuando la persona se vuelve a mover el módulo adicional deja de trabajar y la persona vuelve a ser detectada por los módulos anteriores desarrolladas en [4]. Se aprovecha la figura para mostrar uno problema típico, persona partida por la mitad debido a que pasa por detrás de una mesa que pertenece al background.	58
5.1. Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas ante la situación de seguir a una persona en un lugar interior. La secuencia de imágenes va de izquierda a derecha, y de arriba abajo.	63
5.2. Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas, ante la situación de seguir a dos personas moviéndose en paralelo en un lugar interior. . .	65
5.3. Imagen donde se observa el funcionamiento del módulo que realiza el seguimiento únicamente con la información de la predicción.	66
5.4. Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas, ante la situación de seguir a dos personas que se cruzan.	68
5.5. Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas, ante la situación de seguir a dos personas que caminan pegadas.	70
5.6. Imagen donde se muestra el resultado del tracking en una escena con falsos positivos sin desconectar el módulo de seguimiento en situaciones de falta de información.	72
5.7. Imagen donde se muestra el resultado del tracking en una escena con falsos positivos desconectando el módulo de seguimiento en situaciones de falta de información.	73
5.8. Detección de una persona que permance parada de pie en el interior de un edificio.	74

5.9. Detección de una persona que permance parada de pie en un ambiente exterior.	75
5.10. Detección de una persona que permance sentada en el interior de una habitación.	76
5.11. Detección de varias peronas paradas a la vez.	77
5.12. Permanencia y salida del módulo de personas estáticas.	78

ÍNDICE DE CUADROS

5.1. Características del estudio para la situación de seguimiento de una persona.	62
5.2. Características del estudio para la situación de seguimiento de dos personas caminando en paralelo.	64
5.3. Características del estudio para la situación de seguimiento de dos personas que se cruzan en un instante de tiempo dado. . .	67
5.4. Características del estudio para la situación de seguimiento de dos personas que caminan pegadas.	69

CAPÍTULO 1

INTRODUCCIÓN

La visión artificial o visión por computador es una disciplina compleja que involucra otras ciencias e incluye estudios de física, matemáticas, ingeniería electrónica... El continuo desarrollo de algoritmos, funciones y aplicaciones hace que sea una disciplina en continua evolución. La visión artificial tiene por objetivo modelar matemáticamente los procesos de percepción visual de los seres vivos y generar programas que permitan simular estas capacidades visuales por ordenador. Es una gran herramienta para establecer la relación entre el mundo tridimensional y las vistas bidimensionales tomadas de él. Por medio de esta teoría se puede hacer, por una parte, una reconstrucción del espacio tridimensional a partir de sus vistas y, por otra parte, llevar a cabo una simulación de una proyección de una escena tridimensional en la posición deseada a un plano bidimensional.

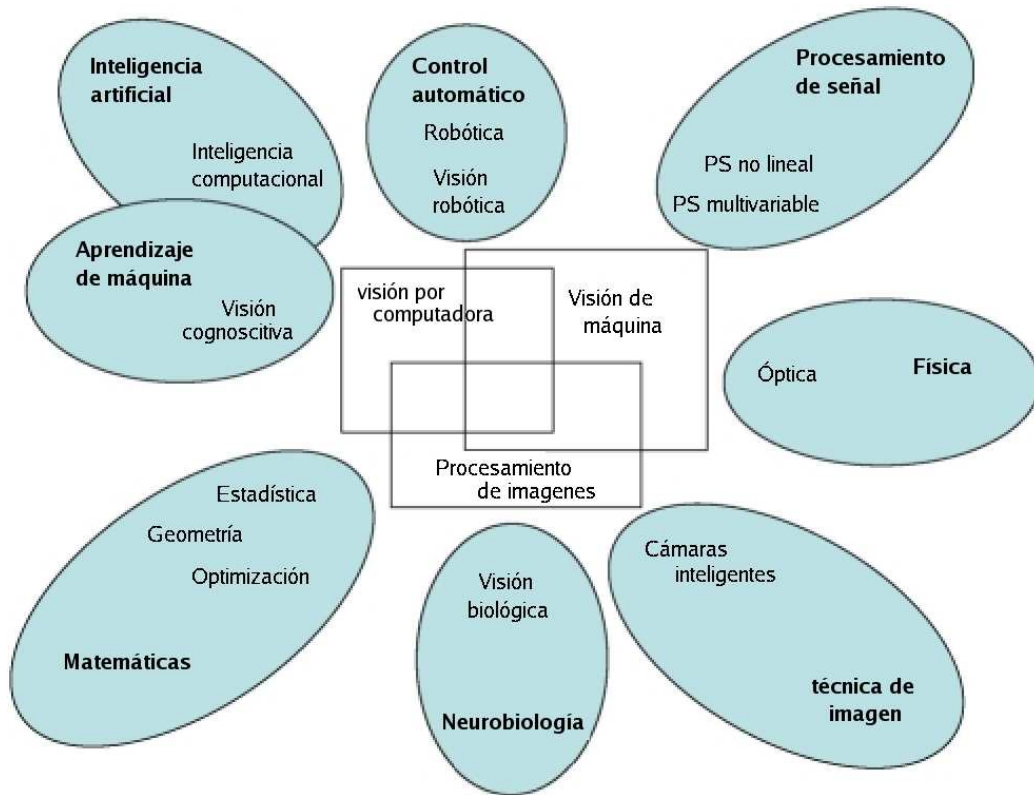


Figura 1.1: Relación de la visión por computador y otras áreas afines.

Los primeros conocimientos que se tienen de esta materia se remontan a los años veinte, cuando se mejora la calidad de las imágenes digitalizadas de los periódicos, enviadas por cable submarino entre Londres y Nueva York. Sin embargo, no es hasta la década de los 50 cuando empiezan a aparecer los primeros trabajos relacionados con la visión artificial. Al principio se piensa que es una tarea sencilla y alcanzable en pocos años, esto se debe a los importantes trabajos realizados por Roberts (1963) y Wichman (1967). El primero demuestra la posibilidad de procesar una imagen digitalizada para obtener una descripción matemática de los objetos que aparecían y el segundo presenta por primera vez una cámara de televisión conectada a un ordenador. Esta ilusión del principio propiciada por los avances comentados, los avances en los ordenadores y por la creencia de que si para los humanos ver es una tarea fácil para un ordenador también lo será, pronto se empieza a oscurecer debido a los limitados avances conseguidos y a que estos sólo se pueden aplicar a un número reducido de imágenes, además se observa como la visión que es una tarea que parece relativamente trivial para los humanos,

es muy compleja de llevar a cabo por ordenadores. Es por ello que los años setenta no presentan avances significativos ya que se produce un continuo abandono de las investigaciones.

En la década de los ochenta vuelven a aparecer las investigaciones relacionadas con la visión por computador en este caso encaminadas a la extracción de características. Así se tiene la detección de texturas Haralik (1979) y la obtención de la forma a través de ellas Witkin (1981). Además en esta década se publican también numerosos artículos entre los que destacan: visión estéreo (Mayhew y Frisby), detección de movimiento (Horn), interpretación de formas (Steven) y detectores de esquina (Kitechen y Rosendfekd). A pesar de la importancia de las investigaciones y artículos recién comentados, el trabajo más importante de la década es el libro de David Marr «Vision: a computational investigation into the human representation and processing of visual information», donde se abordaba por primera vez una metodología completa del análisis de imágenes a través de ordenador.

Por todo ello se puede decir que es a partir de la década de los 80 cuando esta disciplina empieza a cobrar una mayor importancia, manifestación de ello es que empieza a ser una de las principales líneas de investigación en muchas universidades, el número de congresos internacionales aumenta y cada vez aparece un mayor número de revistas especializadas en el tema.

El proceso de visión por ordenador puede subdividirse en seis áreas principales:

- 1. Sensado. Es el proceso que nos lleva a la obtención de una imagen visual.
- 2. Preprocesamiento. Trata de las técnicas de reducción de ruido y enriquecimiento de detalles en la imagen.
- 3. Segmentación. Es el proceso que particiona una imagen en objetos de interés.
- 4. Descripción. Trata con el cómputo de características útiles para diferenciar un tipo de objeto de otro.
- 5. Reconocimiento. Es el proceso que identifica esos objetos.
- 6. Interpretación. Asigna un significado a un conjunto de objetos reconocidos.

Los objetivos más típicos que se pretenden satisfacer mediante la visión artificial son:

- La detección, segmentación, localización y reconocimiento de ciertos objetos en imágenes.
- La evaluación de los resultados.
- Registro de diferentes imágenes de una misma escena u objeto.
- Seguimiento de un objeto en una secuencia de imágenes.
- Mapeo de una escena para generar un modelo tridimensional de la escena, tal modelo podría ser usado por un robot para navegar por la escena.
- Estimación de las posturas tridimensionales de humanos.
- Búsqueda de imágenes digitales por su contenido.

Las aplicaciones de la visión artificial en la actualidad son muy variadas e interesantes, a continuación se muestran algunas de ellas:

- Industria automotriz: medición de las dimensiones de cojinetes de frenos, calibración de ensamblado robótico de sensores de frenos 'anti-lock'.
- Industria de dispositivos médicos: inspección de catéteres en el corazón, lectura de códigos en marcapasos.
- Industrias financieras: inspección detallada de tarjetas financieras.
- Retroalimentación visual para robots.
- Comunicación visual hombre-máquina.
- Empresas de seguridad: videovigilancia.
- Control de tráfico.

Este proyecto fin de carrera, realizado conjuntamente con el desarrollado por Ubaldo González Benítez: «DETECCIÓN DE PERSONAS A TRAVÉS DE VISIÓN POR COMPUTADOR», nace con el objetivo de complementar el mismo introduciéndole varias mejoras. Este proyecto previo consigue la detección de personas en cada frame mediante un algoritmo que realiza la distinción entre primer plano en movimiento (del inglés, foreground) y el fondo (del inglés, background). De esta manera se consigue aislar los píxeles en los que hay movimiento del fondo, consiguiendo detectar los objetos que se mueven. Una vez se realiza esta segmentación, agrupa los píxeles en el foreground que conforman un objeto en movimiento para saber su posición, tamaño, etc para después distinguir mediante diferentes técnicas entre personas y el resto de objetos (vehículos por ejemplo). Una vez se consigue la detección de personas en un frame se vuelve a realizar la misma operación para el siguiente.

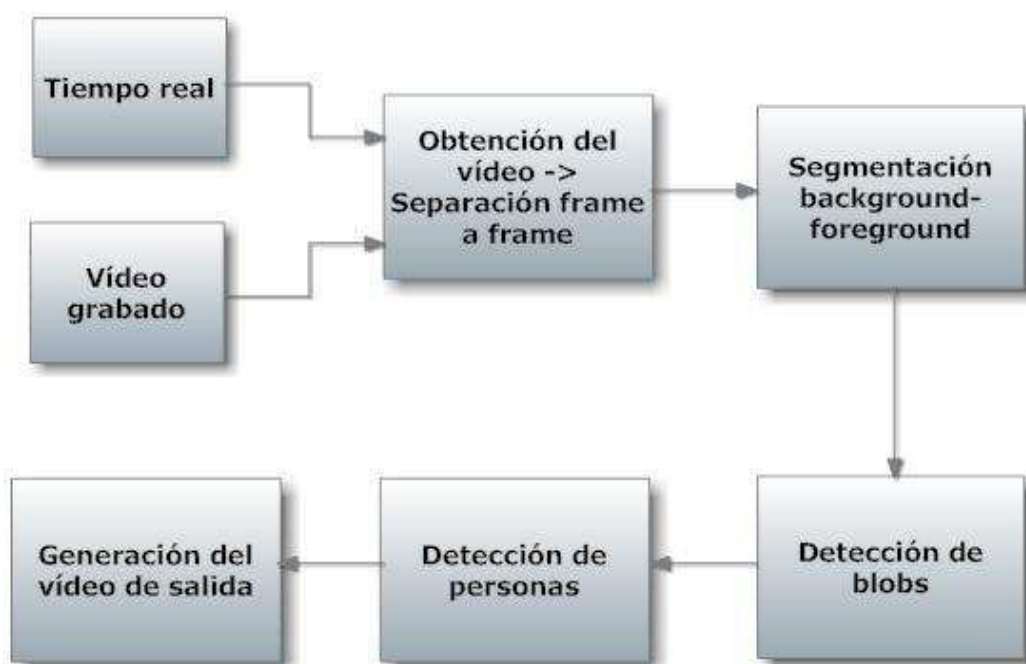


Figura 1.2: Diagrama de flujo de la aplicación desarrollada en [4].

A simple vista se puede observar algunos puntos donde es posible mejorar el rendimiento de la aplicación. Por un lado al detectar el movimiento es evidente que si una persona permanece estática un cierto tiempo en la escena se acabará dejando de detectar. Por otro lado es claro que realizando la

detección frame a frame solo se consigue el seguimiento en el caso de que haya una sola persona. En el caso de que aparezcan más se consigue el seguimiento global, es decir, se determina las nuevas posiciones de las personas existentes pero no se particulariza la nueva posición de cada cual en concreto.

Como se ha comentado el proyecto aquí presentado busca resolver los problemas anteriormente mencionados. Por un lado desarrollar un módulo que sea capaz de fijar una persona que permanece estática antes de que se pierda la más mínima información del foreground.



Figura 1.3: Ejemplo de fijación de una persona cuando esta permanece estática en la escena.

Por otro lado, dado que esta aplicación global está pensada principalmente para la interacción de una o dos personas con un robot, también se busca implementar un módulo que sea capaz de realizar el seguimiento individualizado de dos personas por separado, determinando su posición incluso cuando se produce la pérdida de información, bien sea por un cruce entre ellas, o por la pérdida de información debido a ruido (en estos casos la aplicación solo detecta un blob, es decir, un grupo de píxeles blancos adyacentes en el foreground). Adicionalmente también se quiere desarrollar un bloque encargado de grabar el resultado de la aplicación, bien sea para realizar un post-optimizado de la aplicación, como medida de seguridad, etc.



Figura 1.4: Ejemplo de tracking de personas.

Además de los objetivos comentados, este Proyecto Fin de Carrera tiene una serie de subobjetivos vinculados que son los siguientes:

1. Estudio teórico de la bibliografía existente sobre técnicas de seguimiento con el objetivo de optimizar el algoritmo.
2. Experimentación con la aplicación desarrollada para determinar su eficiencia en los distintos entornos posibles.
3. Análisis de posibles deficiencias del algoritmo para buscarle solución siempre y cuando sea posible.
4. Obtención de conclusiones y posibles trabajos futuros.

En cuanto a la estructura de este documento, este está compuesto por 7 capítulos. En el siguiente capítulo se realiza un resumen de como se encuentra actualmente el estado del arte de la visión por computador, particularizando en el seguimiento de personas. En él se analiza las distintas tendencias, facilitando al lector la información necesaria para situar el proyecto. En el tercer

capítulo se desarrolla un manual sobre las diferentes herramientas y dispositivos necesarios para poder llevar a cabo el proyecto, así como una guía para su instalación. Seguidamente, en el capítulo 4 (Arquitectura funcional de las aplicación), se describe de manera precisa y detallada todas las mejoras que conforman la aplicaciones desarrolladas. Para ello se desarrollan todas las técnicas utilizadas en la actualidad para este tipo de mejoras, analizándolas de forma teórica y comparándolas, para finalmente exponer las aplicaciones desarrolladas. Una vez mostradas y analizadas las mejoras del algoritmo, en el capítulo 5 se realizará una serie de experimentaciones, en distintos entornos y ante distintas situaciones, para así determinar el rango de acción y eficacia del mismo. En los dos últimos capítulos, conclusiones y trabajos futuros, respectivamente, se analizan todas las conclusiones obtenidas durante el desarrollo de esta aplicación y se expondrán tanto, los puntos donde se debe de seguir mejorando en este campo, como los puntos concretos en los que se puede mejorar este algoritmo.

CAPÍTULO 2

ESTADO DEL ARTE

Como se comenta en el capítulo anterior la visión por computador y en concreto el seguimiento de personas es un tema de enorme interés debido al amplio rango de aplicaciones en las que se encuentra inmerso. Es por ello que existe un gran número de investigaciones relacionadas con esta materia.

Después de la detección del movimiento, las aplicaciones están interesadas en darle seguimiento a los objetos de un frame a otro. Los algoritmos de seguimiento están estrechamente vinculados con la detección de movimiento durante el procesamiento. El seguimiento involucra el encuentro de relaciones coherentes entre características de la imagen en frames consecutivos con respecto al color, textura, velocidad y posición.

El seguimiento robusto de una persona es un problema cuyos enfoques cambian constantemente debido a la complejidad y no rigidez de los movimientos del cuerpo humano, los cambios de iluminación, ruido... por lo que los algoritmos de seguimiento son en su mayoría dependientes de la aplicación.

Analizando trabajos anteriores, se observa que la clasificación más relevante de los algoritmos de seguimiento se basa en si estos utilizan o no

modelos de forma para seguir los objetos, o sea seguimiento basados en modelos o no. Además hay una tercera clase de algoritmos que se basan en la utilización de filtros de predicción o estimación.

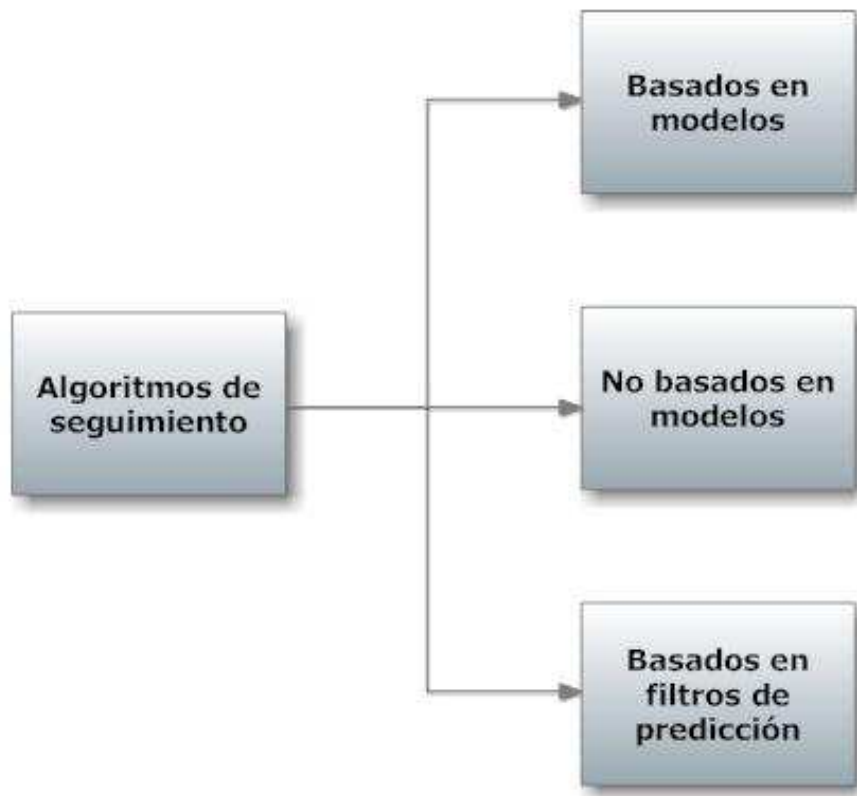


Figura 2.1: Esquema taxonómico general de la clasificación de los métodos de seguimiento.

El seguimiento basado en modelos se refiere a todas aquellas técnicas donde el seguimiento de una persona en el tiempo se basa en la comparación del contenido de cada imagen con un modelo. Estos algoritmos se enfocan más en el problema de la representación de la persona mediante los cambios de apariencia de la misma. Los modelos se construyen off-line con mediciones manuales.

Este método utiliza generalmente un modelo predeterminado del individuo para su seguimiento, se extraen las características de la imagen y se asocian con la estructura del modelo y el movimiento. Los modelos pueden ser creados para el tracking del cuerpo humano entero y para zonas más es-

pecíficas como manos y rostro. Mientras los gestos de la mano son similares en casi todas las personas, las posturas del cuerpo cambian mucho entre una persona y otra.

Como se puede apreciar esta tarea es compleja computacionalmente y requiere de una fuerte segmentación del objeto en movimiento desde el back-ground, por lo que estas técnicas son difíciles de implementar.

Es por ello que surge el seguimiento no basado en modelo, la idea de este tipo de tracking es que la información estructural no siempre es requerida para seguir un objeto y por consiguiente la complejidad y el coste computacional pueden ser reducidos utilizando algún otro método.

Por último se encuentran los algoritmos que realizan el seguimiento visual de personas basándose en predicciones y medidas sobre el sistema destacando entre estos últimos el filtro de Kalman y el filtro de partículas.

En el filtro de Kalman se predice un estado con un determinado nivel de confianza y se estima la diferencia entre los datos sintéticos y las medidas, es decir, es un algoritmo recursivo que se utiliza para estimar la posición y la incertidumbre de la medida en la siguiente imagen.

Mientras que en el filtro de partículas se propone un enfoque que se basa en múltiples hipótesis para construir un conjunto de estados probables. El algoritmo PF permite modelar un proceso estocástico mediante una función de densidad de probabilidad arbitraria, describiéndola numéricamente por medio de un conjunto de puntos, llamados partículas, pertenecientes al espacio de estados del proceso.

Finalmente, en cuanto a la dirección que deben tomar las investigaciones futuras en esta área, estas deben estar enfocadas principalmente a desarrollar algoritmos que sean más rápidos, más precisos, más sencillos y, por tanto con un mejor rendimiento que los actuales. Además de estas mejoras a nivel general, serían necesarias otras para solventar el tracking en algunos casos puntuales como son el seguimiento de personas en situaciones o escenarios muy poblados o el seguimiento de personas que se encuentran ocultas.

CAPÍTULO 3

ARQUITECTURA HARDWARE Y SOFTWARE

Este capítulo nace con la finalidad de proporcionar al lector toda la información que necesita sobre las distintas herramientas y dispositivos necesarios para ejecutar la aplicación sobre detección que aquí se trata. A continuación se muestran los diferentes puntos a tener en cuenta:

3.1. Sistema operativo

La aplicación se desarrolla sobre el sistema operativo Ubuntu. Este es una distribución Linux basada en Debian GNU/Linux que proporciona un sistema operativo actualizado y estable para el usuario, con un fuerte enfoque en la facilidad de uso y de instalación del sistema. Al igual que otras distribuciones se compone de múltiples paquetes de software normalmente distribuidos bajo una licencia libre o de código abierto.

Una de las grandes ventajas que proporciona este sistema operativo es

que se puede utilizar como sistema de programación permitiendo compilar C, C++, Java, Ada, entre otros muchos lenguajes. Este proyecto es programado bajo el lenguaje de programación C++.

1) Requisitos

Los requisitos mínimos recomendados para ejecutar Ubuntu son los siguientes:

- Procesador: 1 GHz x86.
- Memoria RAM: 512 MB.
- Disco Duro: 5 GB (para una instalación completa con swap incluida).
- Tarjeta gráfica VGA y monitor capaz de soportar una resolución de 1024x768.
- Lector de CD-ROM o tarjeta de red.
- Tarjeta de sonido.
- Conexión a Internet.

Cabe destacar que por lo general se puede ejecutar Ubuntu en hardware más antiguos de lo especificado, aunque el rendimiento necesariamente va a ser menor.

2) Instalación

En caso de no tener instalado el Ubuntu es necesario seguir una serie de sencillos pasos:

1. Descargar el CD de instalación de Ubuntu, el Desktop Cd.
2. El archivo descargado es una imagen ISO que se debe grabar en un disco para proceder con la instalación.
3. Arrancar ordenador desde el CD, para ello reiniciar equipo con el disco grabado en el lector.

4. Por último se deben ir siguiendo los pasos de la instalación. Como ayuda se recomienda visitar la siguiente página web:

<http://www.guia-ubuntu.org/index.php?title=Instalaci%C3%B3n.est%C3%A1ndar>

En ella, a parte de ver los pasos a seguir durante la instalación, también se puede encontrar el Desktop Cd y tutoriales.

3.2. Biblioteca OpenCV 2.1.0

Para la realización del proyecto es necesario descargar e instalar una librería específica, a parte de las bibliotecas generales que vienen incorporadas ya en Ubuntu. El nombre de esta librería es OpenCV 2.1.0.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que aparece su primera versión alfa en el mes de enero de 1999, esta se utiliza en infinidad de aplicaciones. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Como meta el proyecto pretende proveer un marco de desarrollo fácil de utilizar y altamente eficiente. Esto se logra realizando su programación en código c y c++ optimizados, aprovechando además las capacidades que proveen los procesadores multi núcleo. Open CV puede además utilizar el sistema de las primitivas de rendimiento integradas de Intel, que es un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

La web oficial del proyecto es la siguiente:

<http://sourceforge.net/projects/opencv/>

1) Instalación

Para la instalación de OpenCV 2.1.0 en Ubuntu se deben seguir los siguientes pasos:

1. Se descarga la biblioteca comprimida de la web oficial.
2. Se extrae.
3. Se accede a la carpeta en el terminal.



Figura 3.1: Paso 3 de la instalación de OpenCV 2.1.0

4. Al no tener makefile se descarga el cmake:

```
javier@javier-laptop:~/Descargas/OpenCV-2.1.0$ sudo apt-get install cmake
```

Figura 3.2: Paso 4 de la instalación de OpenCV 2.1.0

5. Se ejecuta el cmake para crear el makefile:

```
javier@javier-laptop:~/Descargas/OpenCV-2.1.0$ cmake .
```

Figura 3.3: Paso 5 de la instalación de OpenCV 2.1.0

6. Se ejecuta el makefile:

```
javier@javier-laptop:~/Descargas/OpenCV-2.1.0$ make
```

Figura 3.4: Paso 6 de la instalación de OpenCV 2.1.0

7. Por último se instala:

```
javier@javier-laptop:~/Descargas/OpenCV-2.1.0$ sudo make install
```

Figura 3.5: Paso 7 de la instalación de OpenCV 2.1.0

Una vez que se ha concluido la instalación se debe tener cuidado con la ruta donde se ha instalado las librerías de OpenCV para cuando se llamen en el programa se referencie la ruta correctamente. En caso de que la posición no sea la que se quiere solo hace falta copiarlas en la ruta elegida mediante la instrucción cp.

3.3. Librería cvblobslib

A parte de las librerías que ya vienen incorporadas al descargar la biblioteca OpenCV 2.1.0, esta aplicación requiere la descarga e instalación de una librería adicional para OpenCV llamada Cvblobslib. Es un componente algo similar a la librería regionprops de Matlab.

1) *Instalación*

Para la instalación de esta librería se debe, al igual que con la biblioteca OpenCV 2.1.0, descargar la librería de la web oficial. Para encontrar la librería es tan fácil como escribir cvblobslib en el buscador de la web o entrar directamente a la siguiente dirección:

<http://opencv.willowgarage.com/wiki/cvBlobsLib>

Una vez descargada la librería se deben seguir los mismos pasos que para la instalación de la biblioteca, teniendo en cuenta que lo más recomendable es situar la librería dentro de los include de OpenCV 2.1.0.

3.4. Cámara: Logitech QuickCam Pro 9000

El éxito o no de un algoritmo de visión por ordenador depende en gran medida de la calidad de la imagen sobre la que se trabaja, llegando incluso a ser más importante para el éxito de la aplicación que el propio algoritmo. Es por ello que en este proyecto se ha trabajado con una cámara de vídeo de contrastada validez suministrada por el departamento de Automática y Robótica de la Universidad Carlos III, la Logitech QuickCam Pro 9000.

Esta cámara es capaz de producir un vídeo fluido y natural e instantáneas de hasta 8 megapíxeles. Y gracias a su enfoque automático de gama alta las imágenes son siempre nítidas, incluso en primeros planos (a 10 cm de la lente).

1) *Especificaciones*

- Óptica Zeiss® con enfoque automático.

- Sensor nativo de alta resolución de 2 megapíxeles.
- Vídeo en alta definición (hasta 1600 x 1200*).
- Modo de pantalla panorámica de 720p (con sistema recomendado).
- Fotos de hasta 8 megapíxeles (mejoradas desde el sensor de 2 megapíxeles).
- Micrófono con tecnología Logitech RightSound.
- Vídeo de hasta 30 cuadros por segundo.
- Certificación USB 2.0 de alta velocidad.
- Clip universal para monitores LCD, CRT o portátiles

A continuación se muestra una imagen de la cámara utilizada para la obtención de imágenes en tiempo real.



Figura 3.6: Webcam Logitech QuickCam Pro 9000.

2. Instalación

Al trabajar en el sistema operativo Ubuntu no es necesario realizar ningún tipo de instalación, únicamente conectando la cámara al puerto USB se instalará de forma automática. Por el contrario cabe destacar que para grabar vídeos con la misma es necesario descargar el programa Cheese a través de la aplicación Synaptic de Ubuntu.

CAPÍTULO 4

ARQUITECTURA FUNCIONAL

A continuación se realiza la descripción de la aplicación que aquí se trata, estudiando por separado cada una de las mejoras introducidas al proyecto desarrollado en [4]. Para facilitar la comprensión del lector en primer lugar se muestra un esquema de como se encuentra estructurado el capítulo. Para a continuación de esto analizar en detalle cada una de las mejoras, estudiando tanto la teoría de las técnicas existentes como su aplicación concreta en la aplicación.



Figura 4.1: Esquema de la estructura del capítulo.

4.1. Tracking de personas

Como se ha comentado en el capítulo 1 con este módulo se pretende conseguir el seguimiento de dos personas, es decir, desarrollar un programa que sea capaz de determinar cual es la posición de cada persona en el siguiente frame. Para conseguir un seguimiento robusto también se pretende que la aplicación sea capaz de determinar donde se encuentran las personas incluso en ausencia de datos (ruido, cruces entre ellas,...). Para ello a continuación se realiza un estudio teórico de las distintas alternativas existentes para realizar dicho seguimiento para a posteriori elegir la solución más óptima para este caso.

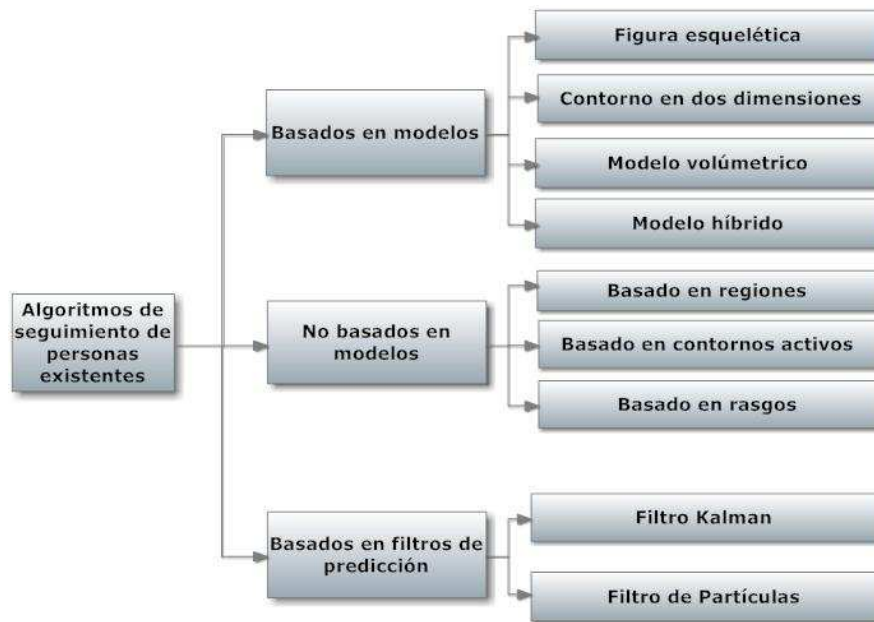


Figura 4.2: Clasificación detallada de los algoritmos existentes para el tracking de personas.

4.1.1. Estudio de los distintos algoritmos existentes

Algoritmos de seguimiento basado en modelos

Como se observa en el capítulo 2 esta clase de algoritmos usan generalmente un modelo predeterminado del sujeto para su seguimiento. La aproximación general de este algoritmo para el seguimiento del cuerpo humano utiliza un estilo de predicción-matching-actualización. Primero se predice la postura del modelo teniendo en cuenta la información previa. Una vez realizado esto, el modelo estimado se sintetiza y se proyecta en el plano de la imagen para su comparación. Para medir la similitud entre la proyección y la imagen, se necesita una función de evaluación. Según diferentes estrategias de búsqueda, se hace ya sea recursivamente o utilizando técnicas de muestreo hasta que la proyección correcta es finalmente encontrada y se utiliza para actualizar el modelo. La estimación de la postura en el primer frame tiene que ser tratada de manera especial. Generalmente, el seguimiento del cuerpo humano basado en modelos involucra tres procesos principales:

- Construcción del modelo del cuerpo humano.
- Representación del conocimiento previo de las restricciones y modelos del movimiento.
- Predicción y estrategia de búsqueda

La construcción de modelos para el cuerpo humano es la base del tracking de personas basado en modelos. Generalmente, a mayor complejidad del modelo y mayor precisión de los resultados se observa un mayor gasto computacional. Tradicionalmente, la estructura geométrica del cuerpo se puede representar de cuatro maneras.

- Figura esquelética
- Contornos en dos dimensiones (2D)
- Modelos volumétricos
- Modelos híbridos

Figura esquelética

Este tipo de seguimiento se basa en el hecho de que la estructura inherente para el movimiento del cuerpo humano es el esqueleto, concluyendo que la esencia del movimiento humano está típicamente restringida en los movimientos del tronco, cabeza y extremidades. Así que este método se basa en representar las partes del cuerpo humano en segmentos y unirlos por puntos. Así el movimiento del torso, las extremidades (brazos y piernas) y cabeza puede ser aproximado al movimiento del correspondiente segmento de línea.

El seguimiento de personas basado en la figura esquelética ha sido desarrollado para el cuerpo humano en su totalidad y para partes de este. Los autores de [9] aproximan los segmentos de líneas en la imagen por la esquelización del cuerpo entero y calculan el ángulo delimitado por tres partes principales del cuerpo con la abscisa vertical, estas tres partes son el tronco, la componente superior de la pierna y la componente baja de esta. En [10] se calcula la transformada de Hough para extraer las líneas que representan las piernas en una secuencia de imágenes. El cambio de inclinación de estas

líneas siguen un movimiento armónico simple y este se utiliza como estándar biométrico. Se utiliza el análisis de Fourier para extraer la frecuencia del cambio de inclinación de las piernas y la figura esquelética para calcular el ángulo relativo entre ellas.



Figura 4.3: Figura esquelética.

Contornos en dos dimensiones (2D)

Los contornos 2D son una representación cerrada de la proyección del sujeto sobre una imagen plana. Las restricciones del movimiento humano, tales como los límites anatómicos de las partes que están unidas, la interpenetración de las partes del cuerpo y posiciones de equilibrio, se le aplican también a las uniones de los contornos 2D, como su contraparte de las figuras esqueléticas.

La principal desventaja de los modelos 2D es que están restringidos por el ángulo de vista. Para resolver esta desventaja surgen los modelos 3D (modelos volumétricos). Pero estos requieren muchos más parámetros y por consiguiente un mayor gasto computacional. En los trabajos [11] y [12] se crean modelos 2D basados en el ajuste de elipses a varios segmentos del cuerpo de la silueta extraída. Entonces derivan vectores característicos tales como la localización del centro de cada elipse, etc.



Figura 4.4: Modelo de contorno 2-D.

En [13] se construye un modelo muy simple para la apariencia del cuerpo humano. Este utiliza tanto la información del color como la espacial de los cuerpos. Ellos toman un rectángulo y lo caracterizan por su centro, altura y ancho, haciéndolo corresponder al cuerpo de una persona.

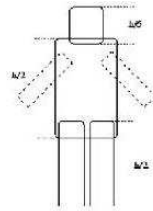


Figura 4.5: Modelo de contorno 2-D.

Modelos volumétricos

Como se ha comentado en el modelo anterior los modelos volumétricos o 3D son computacionalmente muy complejos, la ventaja que poseen es que logran que el seguimiento sea independiente de la vista de la cámara y por lo tanto logra resultados muy buenos frente a la oclusión. Además logra una representación geométrica del cuerpo humano muy detallada y precisa. Para representar las distintas partes del cuerpo se utilizan cilindros elípticos, conos y esferas. La superficie del cuerpo se representa como una malla poligonal. En [14] desarrollan un modelo para el cuerpo completo con 22 grados de libertad. Los sujetos, sin embargo, estaban obligados a usar ropa ceñida al cuerpo con colores contrastados para cada extremidad.

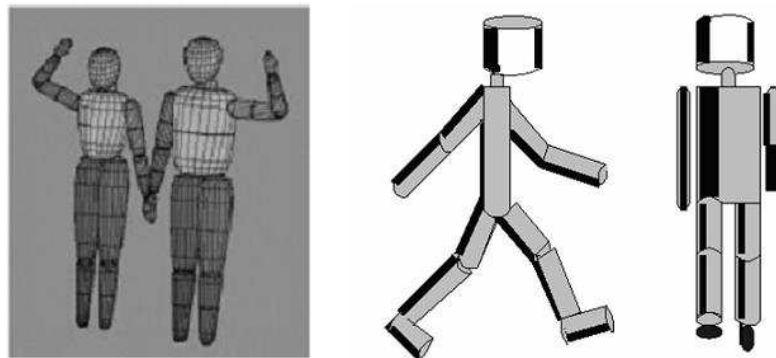


Figura 4.6: Modelos volumétricos.

Modelos híbridos

A lo largo de la historia muchos modelos híbridos han sido propuestos, sin embargo uno de los más importantes es el desarrollado en [15] llamado cuerpo clonado, el cual es dinámicamente clasificado respecto al tamaño y textura y emparejado con cada persona para obtener un tracking más robusto tanto de bordes como de textura de regiones. Este combina modelos 2D y volumétricos.

En [16] definen un modelo para el cuerpo humano como el mostrado en la figura 4.7. Los grados de libertad del modelo incluyen la libre rotación y la traslación de la cintura, los hombros, los codos y todas las uniones en general. Cada rotación de una unión se representa por los ángulos de Euler, lo que conlleva a un total de 21 grados de libertad y las formas del cuerpo son modeladas como elipses 3D.

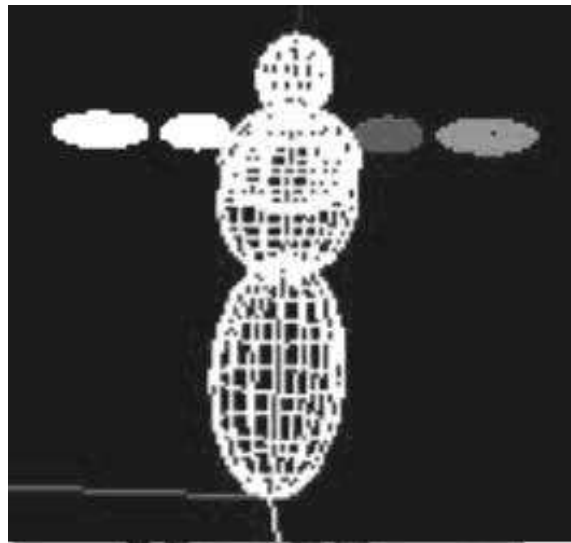


Figura 4.7: Modelos híbridos.

Algoritmos de seguimiento no basado en modelos

Como se ha podido observar los algoritmos de seguimiento basado en modelos son bastante robustos, sin embargo el coste computacional es demasiado elevado, además en muchos casos no es necesaria la información estructural para seguir los objetos y es así como nacen los algoritmos de tracking no basado en modelos.

Los principales algoritmos no basados en modelos son:

- Seguimiento basado en regiones
- Seguimiento basado en contornos activos
- Seguimiento basado en rasgos

Seguimiento basado en regiones

Este tipo de algoritmos identifican un blob o región conectada en el espacio que se asocia a cada objeto de interés y se sigue sobre el tiempo utilizando una medida de similaridad o un parámetro de correlación. El background de la imagen tiene que ser calculado y mantenido dinámicamente, con ello se puede detectar fácilmente las regiones en movimiento por sustracción de este.

El algoritmo descrito en [17] utiliza el matching de matrices en dos direcciones utilizando el criterio de que cuando dos cajas delimitadas se solapan en un frame, se hace un matching del blob resultante con los anteriores juntos. Mientras que el método propuesto en [18] utiliza redes neuronales. Se emplea una red neuronal de retro propagación perceptrón multicapa (RRPM) para el tracking en secuencias de imágenes de baja resolución usando la posición, forma y el color como rasgos. El uso de redes neuronales hace que el tracking sea muy eficiente sobre todo en situaciones reales, particularmente en aquellas donde las imágenes no sean buenas debido a la baja resolución o a que ha salido borrosa.

Una persona detectada en el frame anterior es buscada en el siguiente utilizando una red neuronal. Alrededor de la persona detectada se dibujan rectángulos para delimitarlas y se recolectan las variables de altura y espesor del rectángulo, conformando así la red neuronal. Estos algoritmos solamente obtienen los resultados de tracking a nivel de regiones y se encaminan esencialmente hacia la detección del movimiento.

Seguimiento basado en contornos activos

Los algoritmos basados en contornos activos siguen objetos representando sus contornos como bordes bien delimitados y actualizando estos dinámicamente en frames sucesivos. Las serpientes (contornos activos) son funciones suaves que poseen una función de energía interna y otra de energía externa.

La primera está definida por su configuración y la segunda definida por la energía de la imagen. Dado un conjunto inicial de puntos de la serpiente, esta intenta lograr una posición que resulte un máximo local de las funciones de energía. Aunque los contornos activos son computacionalmente menos complejos que el seguimiento basado en regiones ellos necesitan ser inicializados como contornos separados de cada sujeto individual. Esta inicialización puede ser muy difícil para objetos complejos. En [19] se usan serpientes para recubrir los contornos del cuerpo del caminante.

Estos algoritmos logran extraer directamente formas de los sujetos y proveen de una más efectiva descripción de los objetos que los basados en regiones.

La dificultad principal es que los algoritmos basados en contornos activos son altamente sensibles a la inicialización del seguimiento, haciéndose muy difícil la inicialización automática

Seguimiento basado en rasgos

El fundamento del seguimiento basado en rasgos es la reducción de la complejidad funcional cuando se siguen rasgos sobresalientes del objeto, en vez de la región continua del objeto o sus contornos. Por tanto trabajan el reconocimiento y seguimiento de objetos a través de la extracción de elementos, agrupándolos en el más alto nivel de rasgos y luego haciendo un matching entre imágenes. Se utilizan como rasgos parámetros como las esquinas, la información del color y la textura. Existen dos amplias aproximaciones para el tracking basado en rasgos: seguimiento de rasgos dinámicos y estáticos. Se denomina estático cuando los rasgos se extraen a priori en cada frame y el algoritmo calcula la correspondencia óptima entre ellos. En el seguimiento de rasgos dinámicos los rasgos son determinados y seguidos dinámicamente sobre frames consecutivos, estimando movimiento del rasgo y buscándole en el próximo frame.

Un punto con el seguimiento basado en rasgos es la paradoja entre la complejidad y la eficiencia del seguimiento. Para rasgos de bajos niveles como la posición de las coordenadas de los bordes, que son muy fáciles de extraer, es muy difícil realizar el seguimiento, debido a la dificultad de establecer una correspondencia uno a uno entre ellas. Para rasgos de alto nivel, como blobs y volúmenes 3D, es muy fácil el seguimiento pero muy difícil la extracción.

Estos algoritmos pueden ser subdivididos a su vez en tres categorías según

la naturaleza de los rasgos que se seleccionan: algoritmos basados en rasgos globales, basados en rasgos locales, y basados en la dependencia gráfica.

- 1. Los rasgos que se utilizan en el algoritmo basado en rasgos globales incluyen centroides, perímetros, áreas, algunos órdenes de cuadraturas y colores.
- 2. Los rasgos utilizados en el algoritmo basado en rasgos locales incluyen segmentos de línea, de curvas y vértices de las esquinas.
- 3. Los rasgos utilizados en los algoritmos basados en la dependencia gráfica incluyen una variedad de distancias y relaciones geométricas entre ellos.

Los tres métodos descritos pueden ser combinados.

En [20] se usa un algoritmo de segmentación del background en combinación con un clasificador de objetos para rápidamente encontrar peatones en cada frame del vídeo. Luego de que el posible peatón es detectado, el objeto en movimiento es subdividido en tres zonas (cabeza, parte alta del cuerpo, parte baja del cuerpo) y se extraen el color y las propiedades espaciales de cada parte las cuales forman el modelo de apariencia básico en este sistema. La información del color se analiza en el espacio de colores HSI (matiz, saturación, intensidad).

La efectividad del tracking depende fuertemente de la selección de los rasgos. Su clasificador utiliza el detector para obtener la información básica espacial de un posible objeto y utiliza el modelo de background para añadir mayor información. El detector provee de un conjunto de ventanas de detección en un frame. Cada ventana está completamente definida por su dimensión y posición. Primeramente dividen cada ventana de detección en tres zonas del cuerpo individuales (cabeza, parte alta del cuerpo, parte baja del cuerpo) utilizando la razón (1/4, 3/8, 3/8 respectivamente). Luego cada parte es procesada hasta que su contenido principal consista en una región conectada de píxeles del foreground. Luego se extrae la información del color. El vector característico de cada parte del cuerpo contiene la posición, la dimensión y el histograma.

Un enfoque muy interesante es el publicado por [21]. En él se combinan los métodos de procesamiento de imágenes con los modelos dinámicos del

comportamiento de los peatones. La meta es detectar peatones prestando atención a su dinámica y comportamiento. De manera que necesitan información preferentemente sobre sus desplazamientos que sobre su apariencia.

Teniendo la posición hipotética de un peatón y la correspondiente región de la imagen en un frame, se computa su correlación con la región correspondiente en el frame que le sigue. El máximo de correlación da la localización del mejor matching entre las dos regiones. El vector identificado por la posición corresponde al vector desplazamiento de la región de la imagen actual sobre los dos frames. Lo interesante en este método es que en dos frames consecutivos un ser humano puede cubrir una distancia limitada, de manera que es razonable pensar que la región de búsqueda, usada para computar la correlación, contiene la verdadera posición del blanco. Se aplica entonces restricciones de comportamiento generadas por los vectores de movimiento. Para cada par de frames consecutivos computan el vector desplazamiento por la maximización de la correlación. Esta simple estrategia evita definir modelos de deformación más complejos.

Algoritmos de seguimiento basado en filtros de predicción

Estas técnicas realizan el seguimiento a través de las predicciones que realizan los filtros sobre la nueva posición de la persona gracias a toda la información que estos han ido obteniendo en frames anteriores. En este grupo de algoritmos de seguimiento encontramos principalmente dos: el filtro de Kalman y el de partículas.

Filtro Kalman

El filtro de Kalman consiste en un conjunto de ecuaciones matemáticas que proveen una solución recursiva óptima. La meta de esta solución consiste en calcular un estimador de un sistema t con base en la información disponible en $t-1$, y actualizar, con la información adicional en t dichas estimaciones.

La solución es óptima ya que el filtro combina toda la información observada y el conocimiento previo acerca del comportamiento del sistema para producir una estimación del estado de tal manera que el error es minimizado estadísticamente. El término recursivo significa que el filtro recalcula la solución cada vez que una nueva observación o medida es incorporada al sistema.

El filtro de Kalman es el principal algoritmo para estimar sistemas dinámi-

cos representados en la forma de estado-espacio. En esta representación el sistema es descrito por un conjunto de variables denominadas de estado. El estado contiene toda la información relativa al sistema en un cierto punto en el tiempo. Esta información debe permitir la inferencia del comportamiento pasado del sistema, con el objetivo de predecir su comportamiento futuro.

Lo que hace al filtro tan interesante es precisamente su habilidad para predecir el estado de un sistema en el pasado, en el presente y en el futuro, aún cuando la naturaleza precisa del sistema modelado es desconocida. En la práctica, las variables de estado de un sistema dinámico no pueden ser determinadas exactamente por una medición directa por lo que su medición se realiza por medio de procesos estocásticos que involucran algún grado de incertidumbre en la medición.

En el ámbito del tracking de personas por medio de visión por computador el filtro de Kalman es un algoritmo recursivo que se utiliza para estimar la posición de un punto o característica y la incertidumbre en la medida en la siguiente imagen. Se trata de buscar la característica (punto, borde, esquina, región, etc.) en un área determinada de la siguiente imagen alrededor de la posición predicha, en la que estamos seguros de encontrar la característica dentro de un cierto grado de confianza.

El proceso a ser estimado por el filtro de Kalman tiene como objetivo resolver el problema de estimar el estado $X \in R^n$ de un proceso controlado en tiempo discreto, el cual es dominado por una ecuación lineal en diferencia estocástica de la siguiente forma:

$$X_t = AX_{t-1} + w_{t-1} \quad (4.1)$$

donde:

X_t : Estado en el momento t .

A : Matriz de dimensión $n \times n$ que relaciona el estado en el período previo con el estado en el momento t .

X_{t-1} : Estado en el momento $t-1$.

w_{t-1} : Error del proceso en el momento $t-1$.

con una medida $Z \in R^m$, que es

$$Z_t = HX_t + v_t \quad (4.2)$$

donde:

Z_t :Medidas en el momento t.

H :Matriz de dimensión $m \times n$ que relaciona el estado con la medición.

X_t :Estado en el momento t.

v_t :Error de la medida.

Las variables w_t y v_t que como se ha visto representan el error del proceso y de la medida respectivamente, se asume que son independientes entre ellas, que son ruido blanco y con distribución de probabilidad normal:

$$p(w) \simeq N(0, Q) \quad (4.3)$$

$$p(v) \simeq N(0, R) \quad (4.4)$$

En la práctica las matrices de covarianza de la perturbación del proceso Q , y de la perturbación de la medida, R , podrían cambiar en el tiempo, por simplicidad en general se asumen que son constantes.

El filtro de Kalman estima el proceso anterior utilizando una especie de control de retroalimentación, esto es, estima el proceso a algún momento en el tiempo y entonces obtiene la retroalimentación por medio de los datos observados.

Desde este punto de vista las ecuaciones que se utilizan para derivar el filtro de Kalman se pueden dividir en dos grupos:

- Ecuaciones que actualizan el tiempo o realizan la predicción.

- Ecuaciones que actualizan los datos observados o ecuaciones de actualización.

Las del primer grupo son responsables de la proyección de estado al momento t tomando como referencia el estado $t-1$ y de la actualización intermedia de la matriz de covarianza del estado. El segundo grupo de ecuaciones son responsables de la retroalimentación, es decir, incorporan nueva información dentro de la estimación anterior con lo cual se llega a una estimación mejorada del estado. Es por todo lo explicado que el algoritmo de filtro de Kalman se puede considerar como un algoritmo de pronóstico-corrección.

El primer paso consiste en generar un pronóstico del estado hacia adelante en el tiempo tomando en cuenta toda la información disponible en ese momento y en un segundo paso, se genera un pronóstico mejorado del estado, de tal manera que el error es minimizado estadísticamente. En la siguiente imagen se observa lo descrito:

Figura 1. El ciclo del filtro de Kalman

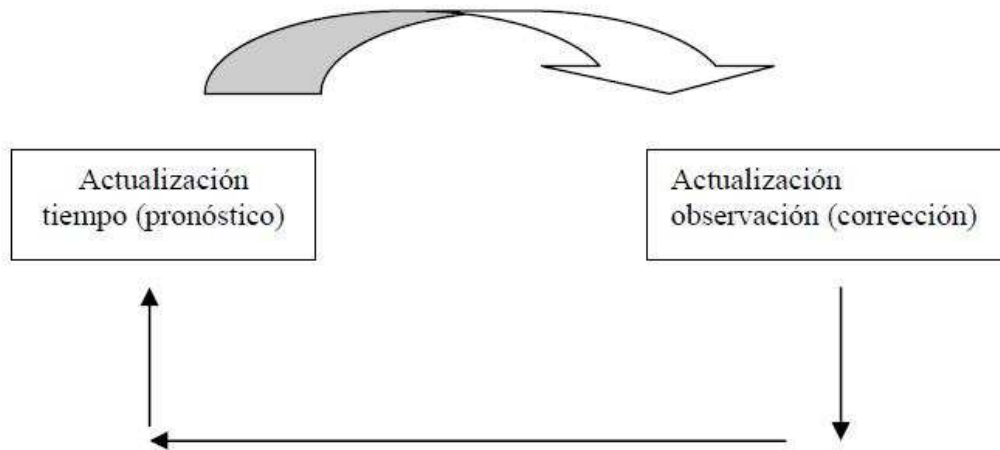


Figura 4.8: El ciclo del algoritmo de Kalman.

A continuación se pasa a mostrar las ecuaciones específicas para la etapa de predicción.

$$\hat{X}_t^* = A\hat{X}_{t-1} \quad (4.5)$$

donde:

$\hat{X}_t^*$: Pronóstico del estado en el momento t .

A : Matriz de dimensión $n \times n$ que relaciona el estado en el período previo con el estado en el momento t .

$\hat{X}_{t-1}$ Estado en el momento $t-1$ ya actualizado.

$$P_t^* = AP_{t-1}A^T + Q \quad (4.6)$$

donde:

P_t^* : Pronóstico de la covarianza del error en el momento t .

A : Matriz de dimensión $n \times n$ que relaciona el estado en el período previo con el estado en el momento t .

P_{t-1} : Covarianza del error actualizada en el momento $t-1$.

Q : Covarianza de la perturbación aleatoria del proceso que trata de estimar el estado.

Una vez mostradas las ecuaciones de la etapa de predicción se muestra la etapa de corrección.

$$K_t = P_t^* H^T (HP_t^* H^T + R)^{-1} \quad (4.7)$$

donde:

K_t : Ganancia de Kalman en el momento t .

P_t^* : Pronóstico de la covarianza del error en el momento t .

H : Matriz de dimensión $m \times n$ que relaciona el estado con la medición.

R : Covarianza de la perturbación aleatoria de la medida.

$$\hat{X}_t = \hat{X}_t^* + K_t(Z_t - H\hat{X}_t^*) \quad (4.8)$$

donde:

$\hat{X}_t$: Actualización de la estimación de estado en el momento t.

X_t^* : Pronóstico del estado en el momento t.

K_t : Ganancia de Kalman en el momento t.

Z_t : Medida en el momento t.

H : Matriz de dimensión $m \times n$ que relaciona el estado con la medición.

$$P_t = (I - K_t H) P_t^* \quad (4.9)$$

donde:

P_t : Actualización de la estimación de estado en el momento t.

K_t : Ganancia de Kalman en el momento t.

H : Matriz de dimensión $m \times n$ que relaciona el estado con la medición.

P_t^* : Pronóstico de la covarianza del error en el momento t.

La primera tarea durante la corrección de la proyección del estado es el cálculo de la ganancia de Kalman, K_t . Este factor de ganancia es seleccionado de forma que minimice la covarianza del error de la nueva estimación del estado. El siguiente paso es realmente medir el proceso para obtener Z_t y entonces generar una nueva estimación del estado que incorpora la nueva observación. El último paso es obtener una nueva estimación de la covarianza del error.

Después de las actualizaciones el proceso es repetido tomando como punto de partida las nuevas estimaciones del estado y de la covarianza del error.

En la siguiente imagen se ofrece un cuadro completo de la operación del

filtro.

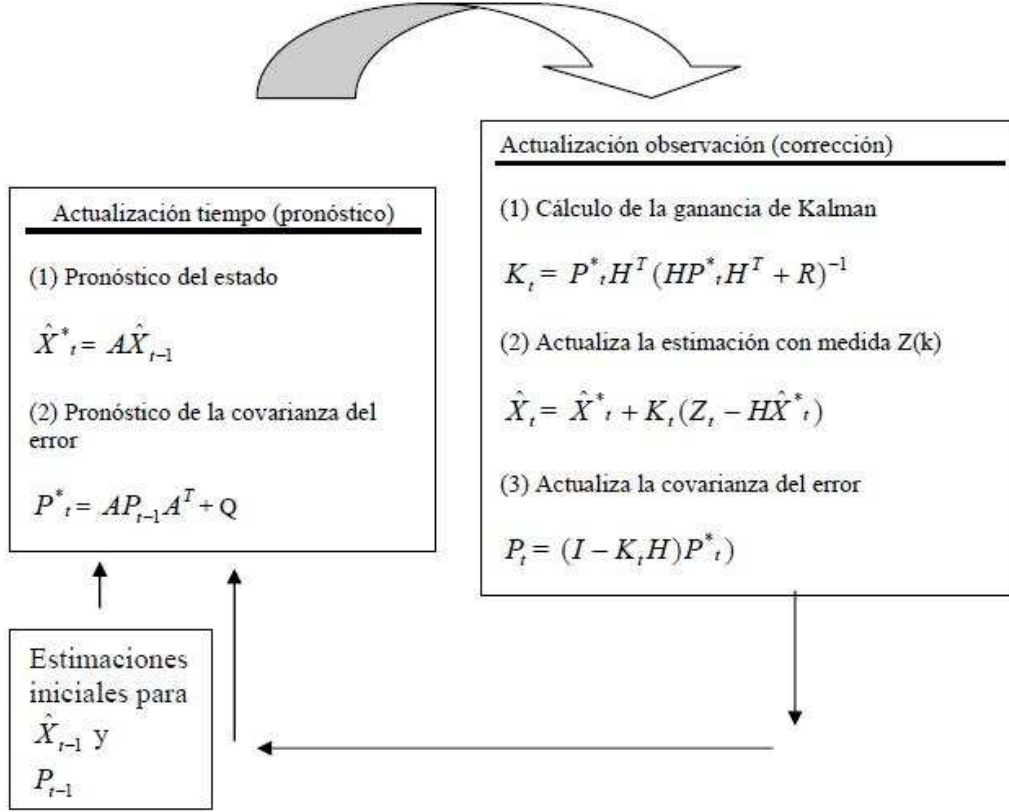


Figura 4.9: Visión completa del filtro de Kalman.

Cuando el filtro de Kalman se utiliza en aplicaciones para visión artificial como en [22] suele suceder que el estado X se corresponde con el de posición del objeto en la imagen determinado por las coordenadas de posición y por las coordenadas de velocidad (x,y,vx,vy) . La observación Z por contrario es un vector de dos componentes (x,y) que se corresponden con las coordenadas de la posición observada del objeto de interés por medio de la cámara de vídeo.

Filtro de Partículas

El filtro de partículas se introdujo originalmente en el área de la visión por computador como el algoritmo llamado Condensation y proporciona una forma simple y efectiva de modelar procesos estocásticos con funciones de densidad de probabilidad arbitrarias (no necesariamente normales o gau-

sianas) por aproximación numérica de éstas. Las partículas (muestras de la función de densidad a posteriori) son estados posibles del proceso que se pueden representar como puntos en el espacio de estados de dicho proceso.

El filtro de partículas es un método secuencial aplicable a cualquier transición de estados o modelo de medida, y por consiguiente se puede utilizar en muchos campos al igual que el filtro de Kalman. El filtro de partículas supone una nueva manera de representar y generar recursivamente una aproximación del estado de la Función de Densidad de Probabilidad (FDP). La idea central es representar la FDP requerida como un conjunto de muestras aleatorias. La aproximación Bayesiana se utiliza para construir la FDP, que define el estado del objeto en el instante actual.

El filtro de partículas representa la densidad a posteriori mediante un conjunto discreto de N partículas $(m_1, \dots, m_N)$ y sus probabilidades asociadas $(\pi_1, \dots, \pi_N)$.

Inicialmente, el conjunto de partículas se escoge a partir de la distribución a priori $p(x_0)$. Si no existe información a priori, entonces las partículas se distribuyen uniformemente por el espacio de estados. Posteriormente, en cada instante de tiempo t , se actualizan las N partículas en función de la acción anterior a_{t-1} y la observación actual z_t . Para ello, se aplica el modelo de movimiento $p(x_t | x_{t-1}, a_{t-1})$ a cada una de las N partículas, generando un nuevo conjunto de partículas. Las partículas nuevas representan la predicción de la variable de estado, sin considerar la observación, se obtiene el peso π^i asociado a cada partícula.

El conjunto de pesos de cada partícula es proporcional a la probabilidad de su estado y a la suma normalizada de sus pesos. La densidad de los pesos es igual al producto de la densidad previa y la probabilidad.

En la siguiente imagen se puede observar lo explicado, en ella se representa un muestreo discreto de una función de densidad de probabilidad continua mediante partículas, cuyo tamaño hace referencia al peso asignado de las mismas.

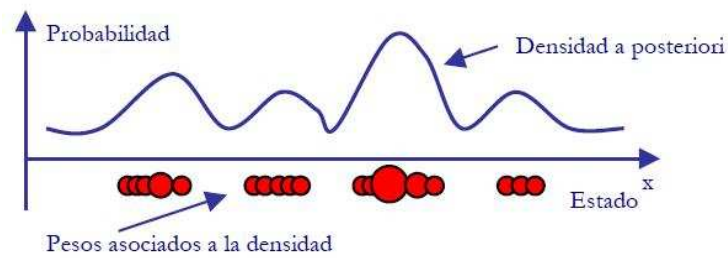


Figura 4.10: Muestreo y pesos obtenidos al aplicar el filtro de partículas.

En un último paso, se remuestrea el conjunto de partículas remplazando N partículas del conjunto actual, proporcional al peso de cada una. En este nuevo conjunto tendrán más probabilidad de desaparecer aquellas partículas que tengan un peso más bajo. Una vez construido el nuevo conjunto de partículas, según la probabilidad de éstas se asocia un peso a cada una. Este nuevo conjunto de partículas constituye una representación muestral de la probabilidad a posteriori.

En la siguiente figura se representa la evolución de las partículas en cada fase, suponiendo que las partículas están estimando un único parámetro, distribuido en el eje horizontal. El área de los círculos representa el peso de cada partícula.

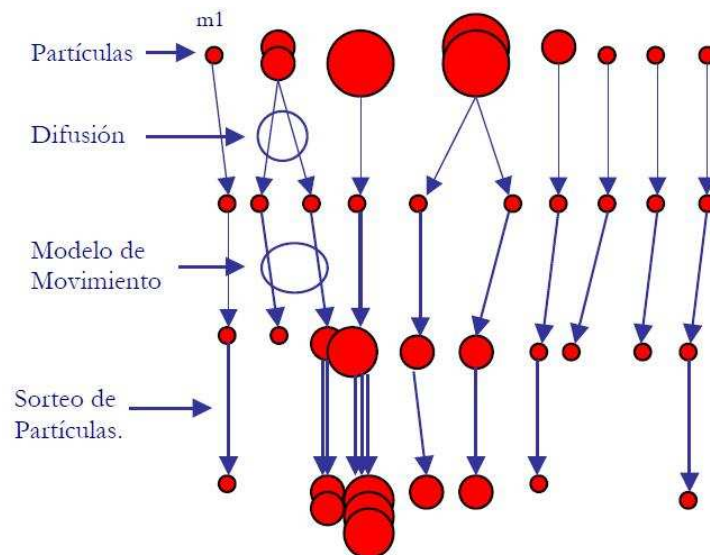


Figura 4.11: Funcionamiento del filtro de partículas.

El algoritmo del filtro de partículas considera una serie de elementos que son los vectores de medida, el modelo de movimiento y el modelo verosimilitud.

Los vectores de medida Z son las medidas que utiliza el sistema, en los casos de visión artificial estas medidas son obtenidas de las imágenes.

El modelo de movimiento F se utiliza para predecir la posición del objeto en el instante actual, a partir de la densidad de probabilidad del instante anterior ($x_{t+1} = F(x_t)$).

El modelo de verosimilitud $P(Z_t | x_t)$, la estimación de la función de probabilidad condicional a posteriori $P(Z_t | x_t)$ define la verosimilitud de la medida observada dado un punto del espacio de estados.

Para poder implementar el algoritmo de actualización es necesario un conjunto bien ponderado de partículas en el instante t con pesos iguales $\frac{1}{N}$, y así actualizar este conjunto para reflejar las nuevas medidas obtenidas en el instante $t+1$. La traza del algoritmo es la siguiente:

- 1. Propagar cada partícula m_i en el tiempo utilizando el modelo de movimiento F del objeto para obtener un conjunto actualizado de partículas $\{m_i^*\}$
- 2. Obtener un nuevo vector de medidas Z y evaluar la densidad de probabilidad posterior π_i^* para cada $m_i^*, \pi_i^* = p(m_i^* | Z)$ que cuantifica la verosimilitud de m_i^* dado un vector de medidas Z . Esto puede ser descrito usando la regla de Bayes anteriormente descrita:

$$Pr(m_i^* | Z) = \frac{p(Z | m_i^*)p(m_i^*)}{p(Z)} \quad (4.10)$$

donde:

$p(Z)$ es la probabilidad a priori de la medida que se asume constante y conocida.

$p(m_i^*) = \frac{1}{N}$, por lo que: $p(m_i^* | Z) = K_p(Z | m_i^*)$ puede ser calculado sin inversión de las ecuaciones de medida.

- 3. Volver a muestrear a partir del conjunto $\{m_i^*\}$ con probabilidades π_i^* y generar un nuevo conjunto bien ponderado $\{m_i'\}$ con pesos iguales

$\frac{1}{N}$ para cada partícula. El muestreo de cada partícula consiste en un estado, un peso y otra información disponible.

- 4. Repetir los pasos 1 y 3 para instantes sucesivos.

4.1.2. Discusión

Una vez analizadas teóricamente todas las posibles alternativas para la implementación del seguimiento de personas se procede a determinar la solución más eficaz para este caso.

Dado que se busca que el programa sea capaz de pronosticar la posición de una persona que se encuentra dentro del campo de visión incluso cuando la información obtenida en el frame es insuficiente, es evidente que es necesario hacer uso de un filtro de predicción.

Las principales ventajas del filtro de partículas son que el tamaño de la pérdida puede ser estimado con exactitud y con una rápida convergencia, el algoritmo es aplicable a procesos de ruido gaussiano o no gaussiano, y puede detectar, relativamente, pequeñas pérdidas. Sin embargo tiene un tiempo de ejecución elevado, el cual se puede reducir ya que existe la posibilidad de ajustar su requerimiento computacional para la capacidad disponible, pero en el caso de reducir el coste computacional los resultados obtenidos no son mejores que los obtenidos por el filtro de Kalman. Por tanto, ante la sencillez de implementación y el reducido coste computacional de este último se ha seleccionado este para la aplicación.

Adicionalmente para conseguir un sistema de seguimiento más robusto y completo se ha optado por utilizar un algoritmo combinado, es decir, que utilice el filtro de Kalman y un algoritmo basado en modelo o no.

Las principales ventajas que presentan los algoritmos basados en modelos son las siguientes:

- Son muy robustos debido a la utilización del conocimiento previo de los contornos 3D.
- Se pueden fusionar la estructura del cuerpo humano, las restricciones de su movimiento y otros conocimientos previos.

- El tracking basado en 3D es capaz de extraer la pose del objeto.
- Los algoritmos de tracking basados en modelo 3D puede ser aplicados aún cuando los objetos cambien en gran medida las orientaciones durante el movimiento.

El inconveniente principal de estos algoritmos es el elevado coste computacional necesario para llevarlo a cabo, más si cabe teniendo en cuenta que esta aplicación tiene el objetivo de ser robusta y conseguir buenos resultados de detección y seguimiento en cualquier situación, sin particularizar en ninguna en concreto. Veamos que se quiere decir con esto, imagínese que se desea implementar una aplicación para la detección y seguimiento de una persona erguida y de figura completa. En este caso se puede ser muy restrictivo a la hora de filtrar objetos en movimiento y por tanto se asegura que el ruido es prácticamente nulo. Con ello se consigue que el número de modelos sea relativamente bajo, por lo que este tipo de seguimiento es adecuado porque se consiguen resultados muy optimizados con un coste computacional aceptable. Incluso si solo se quiere seguir a una persona en concreto de entre todas se puede realizar con este método obteniendo un modelo de esta persona y siendo restrictivos a la hora de aceptarlo. Pero en nuestro caso, donde el programa tiene la intención de seguir personas en diferentes posturas, en entornos complicados donde puede que solo se detecte a la persona de cintura para arriba o cuya forma es muy diferente a la real bien sea por ruido o porque las dos personas se cruzan dando una forma extraña por ejemplo, este tipo de seguimiento es muy complicado pues para poder hacerlo fielmente en todos estos casos el coste computacional es elevadísimo a la vez que es muy complicado conseguir el seguimiento de una persona si por ejemplo de un frame a otro a la persona se le cortan las piernas por culpa de que su camisa es muy parecida al fondo.

Por otro lado las principales ventajas que presentan los algoritmos no basados en modelos son:

- Sencillez.
- Reducido gasto de memoria.
- Reducido gasto computacional

Por tanto con este tipo de algoritmos, que se basan en características concretas de una persona y no en su modelo concreto, se consigue una solución con menor coste computacional (de vital importancia para trabajar en tiempo real) además de unos mejores resultados cuando se trabaja en entornos complicados y cambiantes.

Una vez discutidas las ventajas e inconvenientes de ambas alternativas es obvio que se ha optado por una técnica que combine los algoritmos de seguimiento basados en no modelo con el filtro de predicción de Kalman.

4.1.3. Presentación del algoritmo desarrollado

A continuación se detalla de manera más precisa el algoritmo desarrollado para esta aplicación con el objetivo de que el usuario pueda tener una mejor comprensión al observar el programa en ejecución.

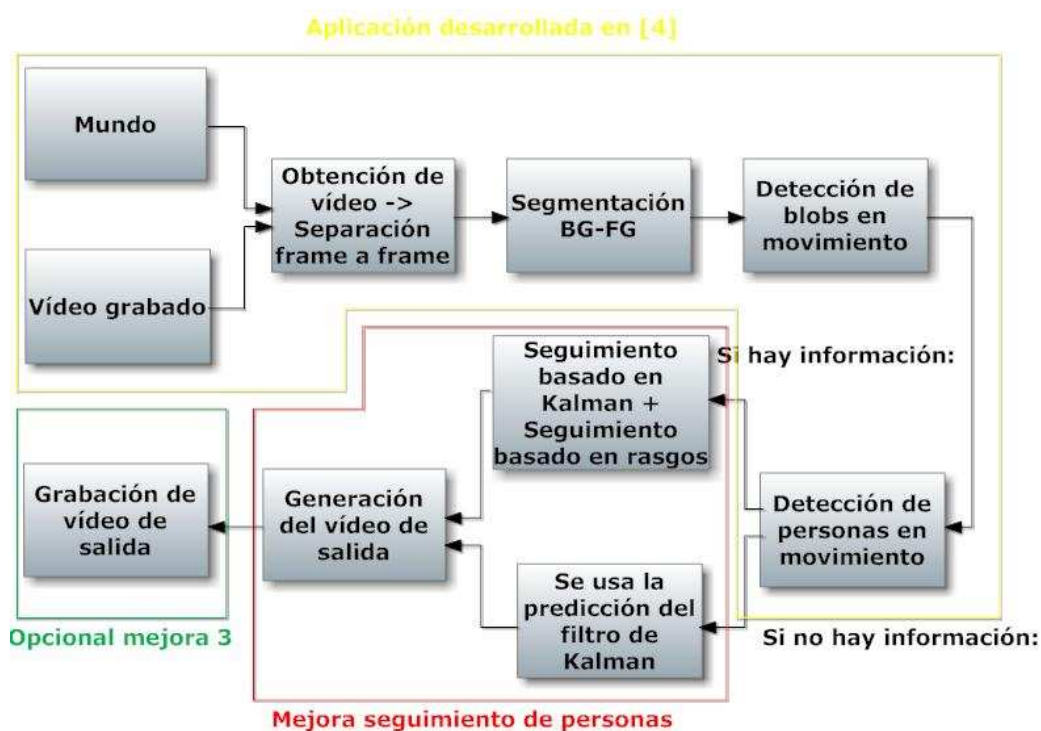


Figura 4.12: Diagrama de flujo de la mejora de seguimiento de personas.

En primer lugar concretar que el tipo de seguimiento basado en no modelo utilizado es del tipo rasgos. Así la técnica combinada con el filtro de Kalman consiste en primer lugar en realizar la predicción de la nueva posición de cada persona a través del filtro para a continuación hacer una matriz matching consistente en una distancia euclídea con la proporción de cada color en el espacio RGB y la proporción de la distancia euclídea entre los centroides del frame actual y las predicciones obtenidas por el filtro. Así a continuación se asigna a cada persona aquel blob cuya distancia euclídea respecto a su predicción sea menor. Posteriormente se realiza la corrección del filtro de Kalman sabiendo que la posición real es la asignada en el frame actual. Y así sucesivamente, es decir, en el siguiente frame se vuelve a realizar las predicciones, las asignaciones por la matriz matching y por último las correcciones del filtro. Indicar que el filtro de Kalman es inicializado cada vez que la persona sale de la escena para que cuando otra vuelva a entrar utilice el primer dato que se tiene de esta persona como valor inicial para a partir de ahí comenzar con las predicciones y correcciones, de no ser así, cuando una persona apareciera después de que otras ya se hayan movido por la escena, el filtro tardaría mucho más tiempo en converger hacia valores óptimos. Dado que la aplicación procesa 25 frames por segundos se está hablando de pocos segundos pero estos pocos segundos pueden ocasionar seguimientos incorrectos en determinadas situaciones límite.

Por último se explica como funciona el programa cuando hay ausencia de información repentina. El módulo está dotado de un algoritmo de modo que cuando la trayectoria de una persona indica que aun está dentro de la escena, esto es, no ha sobrepasado los límites exteriores del campo de visión, a esta persona se le asigna como nueva posición la que predice el filtro de Kalman, por tanto en este caso no hay corrección y la aplicación muestra la posición de la persona en frame sucesivos de acuerdo a la información que ha obtenido Kalman. Una vez que vuelve a haber información al respecto de su posición (las personas terminan de cruzarse en la escena por ejemplo) este módulo deja de funcionar y el seguimiento vuelve a funcionar como se explicó anteriormente.

Para una visión más intuitiva la pantalla de salida de este programa se ha desarrollado de tal manera que una persona es seguida mediante una raya roja, la cual une la posición del centroide en el frame anterior con el actual, mientras que la otra es seguida mediante una raya verde que realiza la misma función. Por último cuando la posición de la persona es determinada por predicción a través de Kalman y no por el resultado obtenido mediante la detección en el frame actual la persona aparece englobada por un rectángulo

CAPÍTULO 4. ARQUITECTURA FUNCIONAL

amarillo en vez de rojo. En las siguientes imágenes se puede observar lo comentado, en la primera de ellas se puede ver el funcionamiento normal del programa, es decir, se dispone de toda la información y en la segunda cuando hay ausencia de información y se utiliza tan sólo la información proporcionada por el filtro de Kalman.

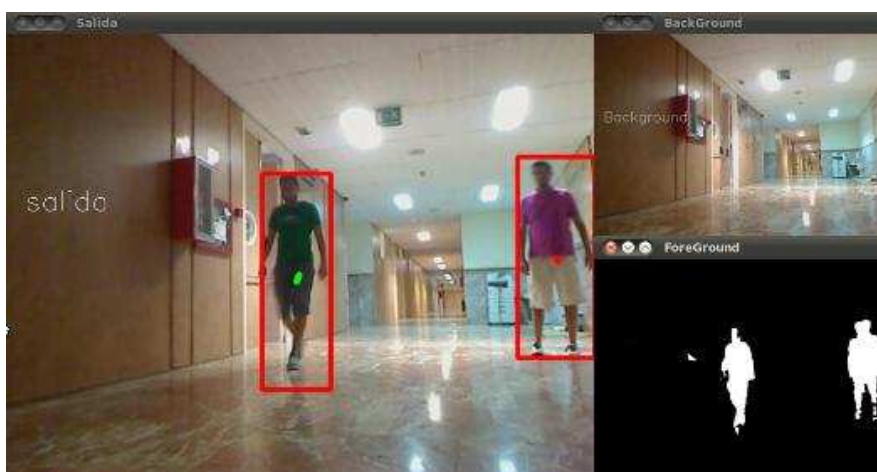


Figura 4.13: Pantalla de salida de la aplicación ante una situación en la que se dispone de toda la información.



Figura 4.14: Pantalla de salida de la aplicación ante una situación en la que hay ausencia de información.

4.2. Detección de personas estáticas

La aplicación desarrollada en [4] es capaz de detectar personas, sin embargo tiene el gran inconveniente de que si una persona permanece inmóvil en el radio de acción del programa, bien porque duerme, ve la tele o porque está en una mesa trabajando, a medio-largo plazo se deja de hacer la detección de esta persona y se determina que ya esa persona ha desaparecido (sale del foreground). Con el objetivo de solventar este problema nace esta mejora de detección de personas estáticas (aquí no hay seguimiento pues la persona esta fija en una posición).

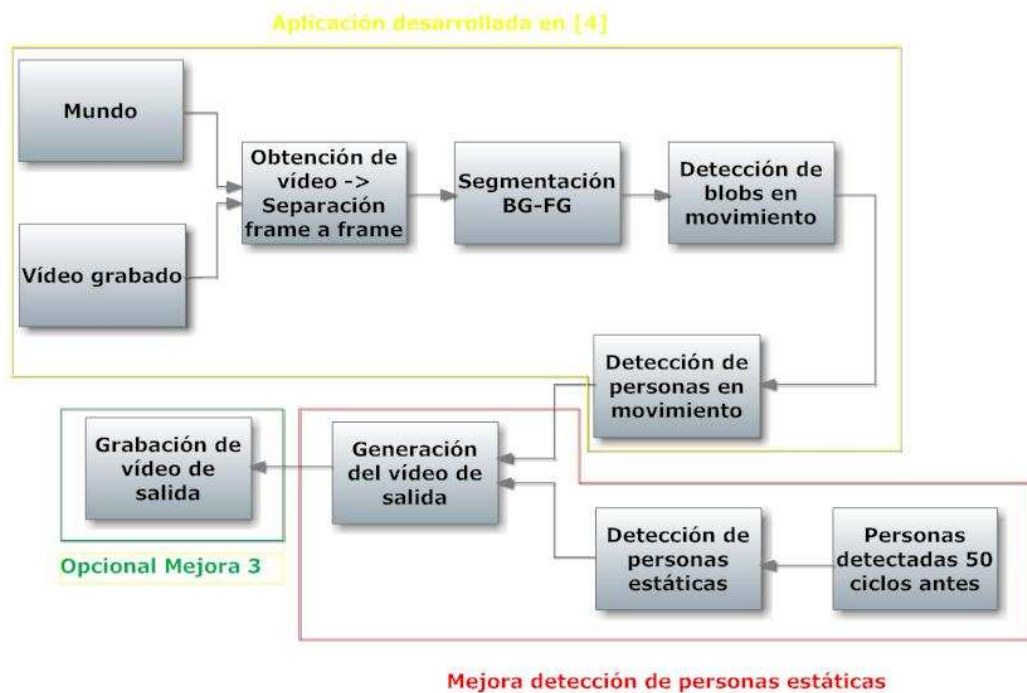


Figura 4.15: Diagrama de flujo de la mejora de detección de personas estáticas.

Este bloque se encarga, a grandes rasgos, de comparar cada cierto tiempo si hay una persona igual (o muy parecida), en cuanto tamaño y posición, a las que había hace un determinado número de ciclos. De esta manera se aprovecha que la persona estática permanece un tiempo en el foreground antes de desaparecer. Si cuando se realiza esta comparación sale afirmativa, entonces se determina que hay una persona estática en esa posición y

CAPÍTULO 4. ARQUITECTURA FUNCIONAL

se almacena un blob (persona) con esas dimensiones y posición. Esta persona permanece almacenada, y por tanto se muestra hasta que se cumpla la condición de salida.

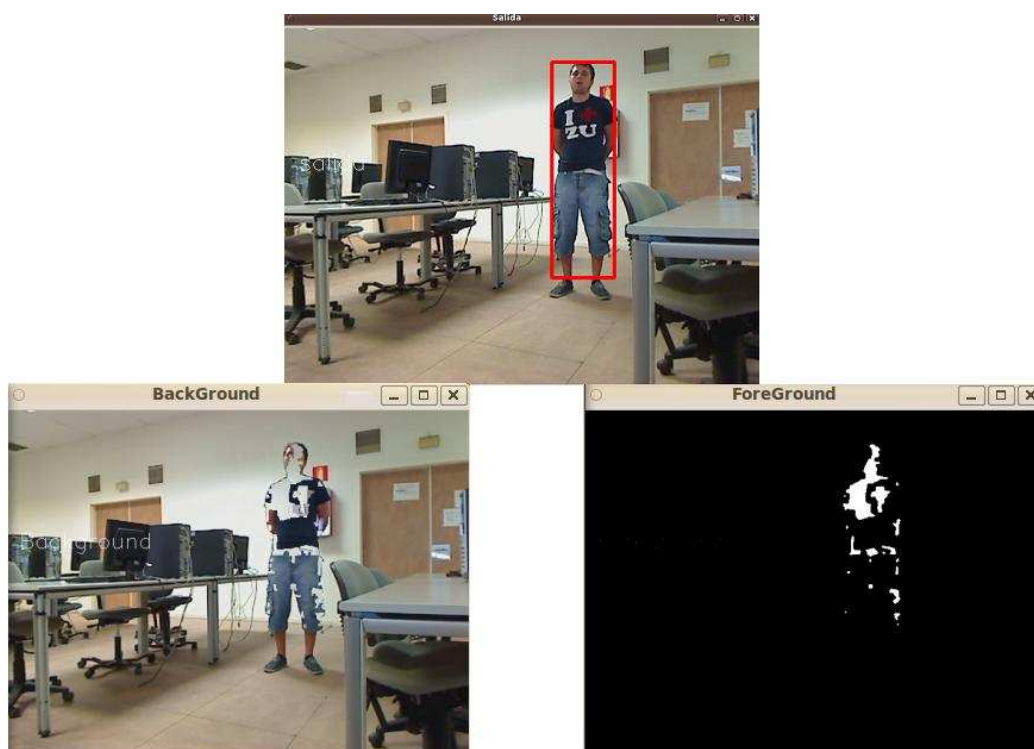


Figura 4.16: Detección de persona estática. Se puede observar como gracias a el módulo que aquí se describe la persona está detectada incluso cuando ya apenas aparece en el foreground y casi por completo en el background.

La condición de salida consiste en que un blob de las mismas (o muy parecidas) características aparezca al lado del que está almacenado como estático con lo que se determina que la persona se ha vuelto a mover y por tanto se vuelve a detectar a través de los módulos anteriormente explicados.

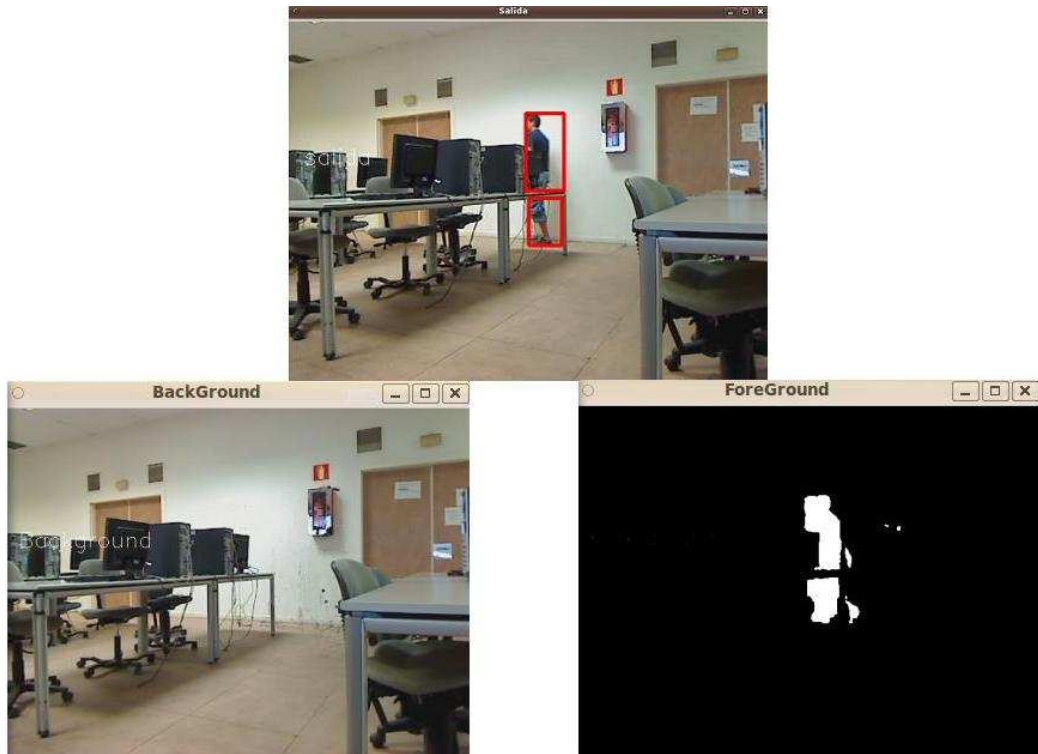


Figura 4.17: Persona estática se vuelve a mover. Demostración de como cuando la persona se vuelve a mover el módulo adicional deja de trabajar y la persona vuelve a ser detectada por los módulos anteriores desarrolladas en [4]. Se aprovecha la figura para mostrar uno problema típico, persona partida por la mitad debido a que pasa por detrás de una mesa que pertenece al background.

El lector puede plantearse las siguientes dudas al analizar este módulo:

- ¿Qué ocurre si 50 ciclos después hay una persona justo en el mismo sitio que había otra antes?: Pues lo normal es que no detecte como que existe una persona estática pues no solo tiene que haber una persona en la misma posición, sino que tiene que tener un área, diferencia de x , diferencia de y , ratio (área blob/área caja englobante), relación ancho/alto muy parecido para que se determine que es la misma persona.
- ¿Qué ocurre si 50 ciclos después la misma persona se ha estado moviendo pero justo en ese instante está en la misma posición?: En este caso si la detectaría como estática, pero no es problema, pues desde que siga

moviéndose en el siguiente frame ya se solventaría el error y en caso de que quedará en esa posición pues ya estaría fijada.

- ¿Qué ocurre si cuando una persona está fijada como estática pasa una por delante?: Las dos personas seguirían estando detectadas y trackeadas, la estática no cumple la condición de salida porque es muy difícil que la que pasa por delante tenga las mismas características (área, diferencia de x, ratio,...) y por tanto sigue detectada en esa posición y la que se está moviendo se sigue con su seguimiento con los módulos explicados anteriormente.

4.3. Grabación de vídeo

Finalmente se introdujo un módulo opcional que permite realizar el grabado del vídeo de salida. Este módulo nace con el propósito de retroalimentar la aplicación de manera que se pueda estudiar de manera minuciosa y detallada los resultados obtenidos una y otra vez, no solo en tiempo real. Además la grabación del vídeo puede ser muy interesante para las empresas de seguridad pues necesitan tener vídeos con todo lo ocurrido para en caso de que ocurra una incidencia tener todo grabado para un posible uso posterior.

CAPÍTULO 5

EXPERIMENTACIÓN

Una vez que se explica con detalle las diferentes mejoras que se introducen en [4], así como los objetivos que se tratan de alcanzar con ellas, es necesario realizar un proceso de experimentación para determinar si esos objetivos se cumplen. De esta manera se podrá determinar cuales son los límites y los trabajos futuros a realizar para mejorarlas.

Como se comenta en el capítulo 4 el número de mejoras introducidas son 3 (tracking de personas, detección de personas estáticas y grabación del vídeo salida), en este capítulo se hace la experimentación de las dos primeras ya que hacer la experimentación de la tercera carece de sentido, pues en ella tan sólo se almacena el vídeo de salida.

Así el capítulo se encuentra estructurado en dos partes, una primera que abarca la experimentación de la mejora para realizar el tracking de personas y una segunda que trata la detección de personas estáticas, a su vez dentro de cada una de estas secciones se realiza el estudio de situaciones concretas de interés para analizar el comportamiento del programa ante ellas. Se estudian diferentes situaciones en la que el grado de complejidad es cada vez mayor.

5.1. Experimentación de la mejora: Tracking de personas

El objetivo de este apartado es determinar el éxito de la mejora de seguimiento personas descrita en 4.1. Para ello se realizan una serie de pruebas o evaluaciones que van aumentando su grado de complejidad.

5.1.1. Seguimiento de una persona sola

Después de la infinidad de pruebas que se realizan en distintos lugares (todos ellos interiores) para esta situación se concluye que los resultados que se obtienen son óptimos, siendo la persona seguida el 100 % del tiempo. En la figura 5.1 se muestra una secuencia de imágenes de una de las experiencias donde se puede observar la calidad de los resultados de la misma, en la siguiente tabla se muestran las características impuestas para el estudio.

Lugar: Tercera planta del edificio Bethancourt.	
Descripción: El vídeo se graba en el pasillo de la tercera planta del edificio Bethancourt. Se coge este lugar pues es un escenario típico para el que funciona la aplicación.	
Hora: 17:15 h	Vídeo: 1:10 min
Iluminación: Artificial, de los fluorescentes situados en el techo del pasillo.	Altura a la que se coloca la cámara: La cámara se sitúa a medio metro sobre el suelo.
Parámetros del algoritmo de segmentación: ▪ $L_c = 128$ ▪ $L_{cc} = 64$ ▪ $\alpha_2 = 0.01$	Parámetros de algoritmo de filtrado de personas: ▪ $4.000 < \text{Área} < 90.000$ ▪ $0,3 \leq \text{Ratio} \leq 0,8$ ▪ $0,1 \leq \text{Rec} \leq 1$

Cuadro 5.1: Características del estudio para la situación de seguimiento de una persona.

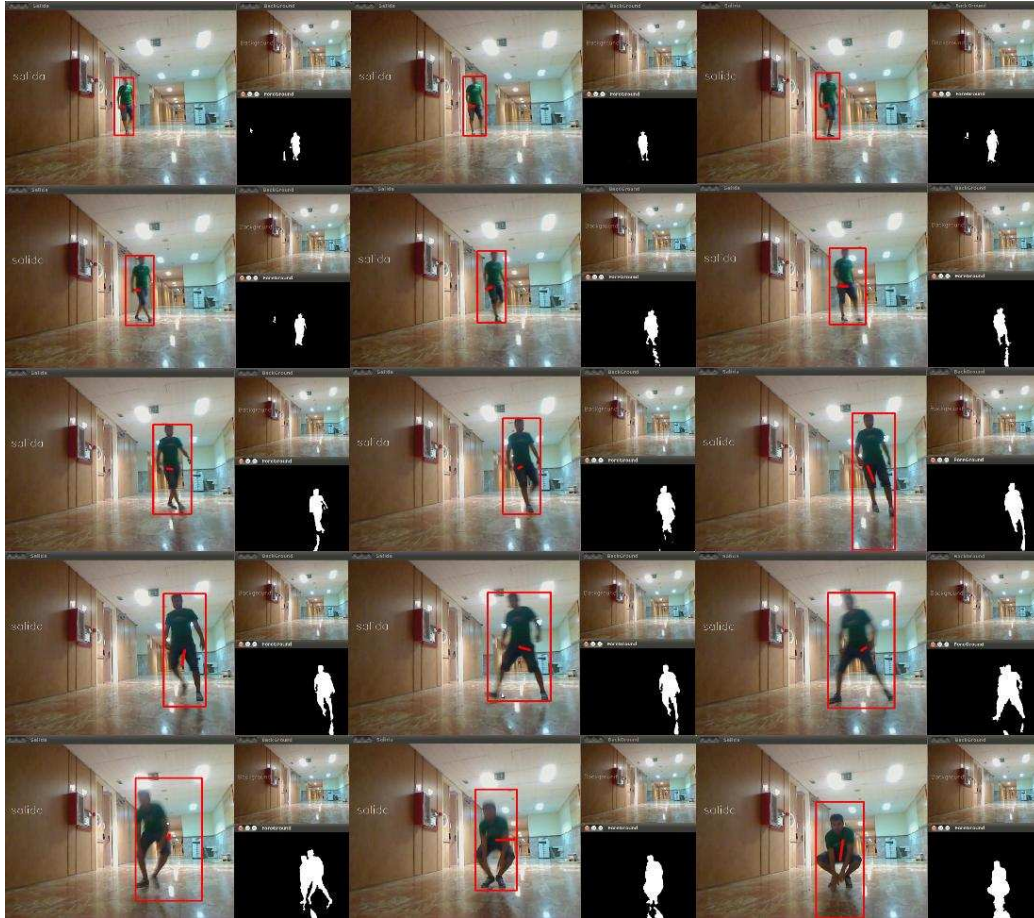


Figura 5.1: Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas ante la situación de seguir a una persona en un lugar interior. La secuencia de imágenes va de izquierda a derecha, y de arriba abajo.

Como se puede observar la persona se sigue ininterrumpidamente desde el frame 1 al frame 15 mediante una raya roja que une los centroides del frame anterior y del actual, además como se puede ver en ningún momento el sistema se queda sin información por lo que no se hace uso del módulo utilizado para este tipo de casos.

5.1.2. Seguimiento de dos personas sin cruzarse

Con esta situación se pretende probar el programa ante una serie de experiencias más exigentes en las cuales aparecen dos personas caminando paralelamente entre ellas, es decir, sin cruzarse. Al igual que en la situación anterior en este caso los resultados obtenidos son excelentes en todas las pruebas que se realizan, cumpliéndose el seguimiento de ambas personas (una en rojo y otra en verde) durante el 100 % del tiempo, lo que demuestra la solidez de la aplicación. En la imagen 5.2 se muestra una de las experiencias con las que se evalúa la aplicación, las características de la misma vienen dadas en la siguiente tabla.

Lugar: Segunda planta del edificio Bethancourt.	
Descripción: El vídeo se graba al final del pasillo de la segunda planta del edificio Bethancourt debido a que es un lugar que cumple con una serie de condiciones (poco tránsito, interior, buena iluminación, carencia de objetos que oculten a las personas...) que lo hacen ideal para la prueba del mismo.	
Hora: 9:50 h	Vídeo: 1:52 min
Iluminación: Artificial y natural, la artificial proveniente de los fluorescentes situados en el techo del pasillo y la natural de un ventanal situado a la derecha del escenario.	Altura a la que se coloca la cámara: La cámara se sitúa a medio metro sobre el suelo.
Parámetros del algoritmo de segmentación: ▪ $L_c = 128$ ▪ $L_{cc} = 64$ ▪ $\alpha_2 = 0.01$	Parámetros de algoritmo de filtrado de personas: ▪ $1.700 < \text{Área} < 60.000$ ▪ $0,45 \leq \text{Ratio} \leq 0,95$ ▪ $0,1 \leq \text{Rec} \leq 1.20$

Cuadro 5.2: Características del estudio para la situación de seguimiento de dos personas caminando en paralelo.

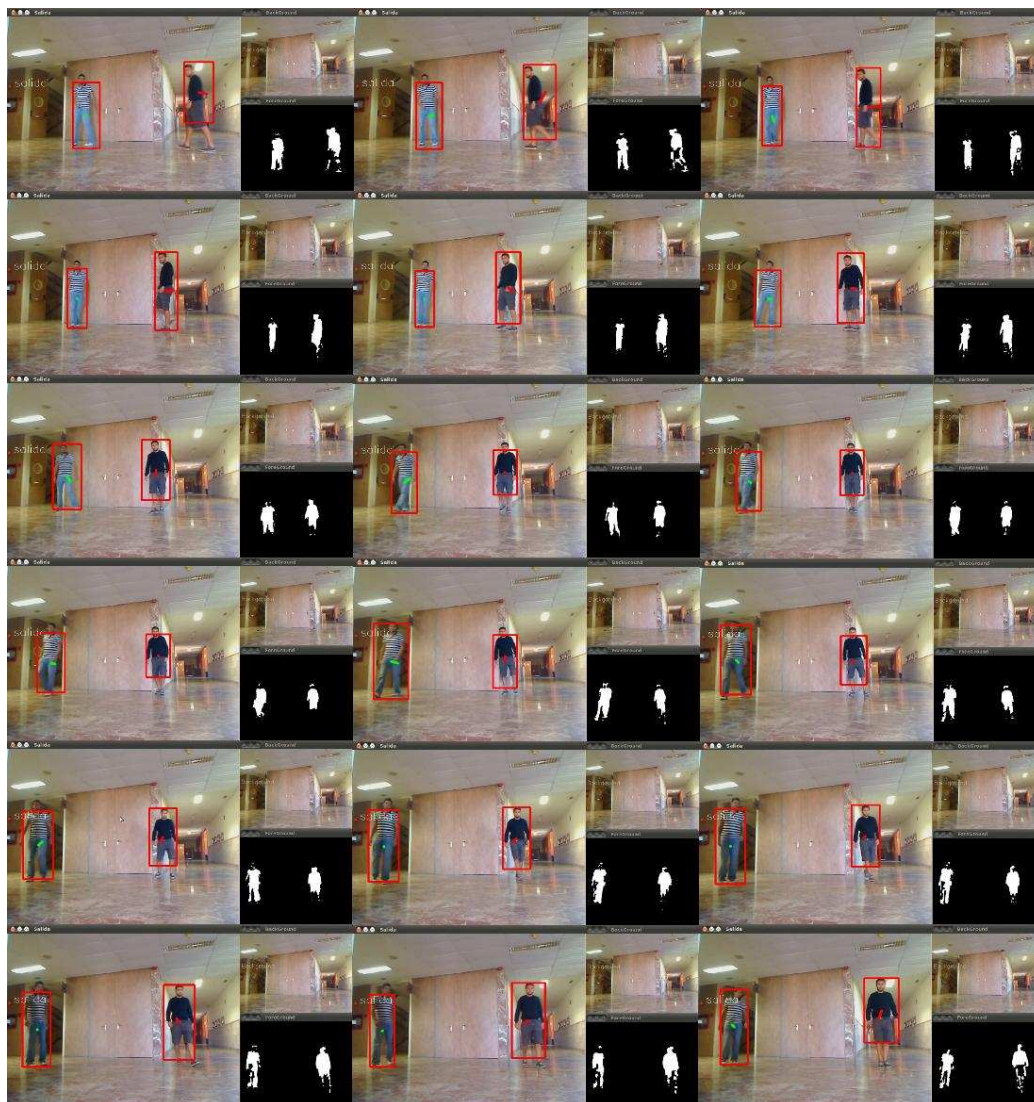


Figura 5.2: Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas, ante la situación de seguir a dos personas moviéndose en paralelo en un lugar interior.

Cabe resaltar la situación que sucede en la siguiente imagen en la cual actúa el módulo que determina la posición de las personas únicamente a partir de la predicción realizada por el filtro de Kalman, como se puede apreciar en ella el módulo de detección deja de detectar a la persona de la izquierda en un determinado momento por lo que el sistema se queda sin la información necesaria para realizar el seguimiento, pero sabe que la persona

aún se encuentra en la escena por lo que utiliza la información del filtro de predicción y sigue a la persona mediante un recuadro amarillo. Esta situación es interesante destacarla ya que este módulo se crea principalmente para aplicarlo a situaciones como las que a continuación se muestran, sin embargo se puede apreciar que puede actuar en otras más sencillas proporcionando robustez al sistema.



Figura 5.3: Imagen donde se observa el funcionamiento del módulo que realiza el seguimiento únicamente con la información de la predicción.

5.1.3. Seguimiento de dos personas cruzándose

En estas situaciones se puede observar como gracias a la actuación del filtro de predicción la aplicación es capaz de seguir a las personas aunque éstas se crucen. El filtro de predicción actúa de manera eficiente en la mayoría de las situaciones, si bien es verdad que hay un pequeño porcentaje (menos de un 5%) de situaciones en las que el filtro no acierta la trayectoria seguida de los individuos debido a que estos antes de cruzarse realizan una serie de movimientos que no siguen ninguna tendencia. En la imagen 5.4 se observa una de las muchas grabaciones que se realizan en la que se cruzaban personas, para realizar esta grabación se imponen las características dadas por la tabla 5.3.

Lugar: Laboratorio del Departamento de Robótica y Automática.	
Descripción: El vídeo se graba en el laboratorio donde se desarrolla la aplicación debido a que se puede controlar el número de personas que entran y salen de la escena.	
Hora: 18:35 h	Vídeo: 1:18 min
Iluminación: Artificial y natural, la artificial mediante los fluorescentes situados en el techo del laboratorio y la natural debida a una serie de ventanas que se encuentran situadas tras la cámara.	Altura a la que se coloca la cámara: La cámara se sitúa sobre una mesa que tiene una altura de aproximadamente un metro.
Parámetros del algoritmo de segmentación: ▪ $L_c = 128$ ▪ $L_{cc} = 64$ ▪ $\alpha_2 = 0.01$	Parámetros de algoritmo de filtrado de personas: ▪ $1.200 < \text{Área} < 50.000$ ▪ $0,35 \leq \text{Ratio} \leq 0,85$ ▪ $0,1 \leq \text{Rec} \leq 1.20$

Cuadro 5.3: Características del estudio para la situación de seguimiento de dos personas que se cruzan en un instante de tiempo dado.

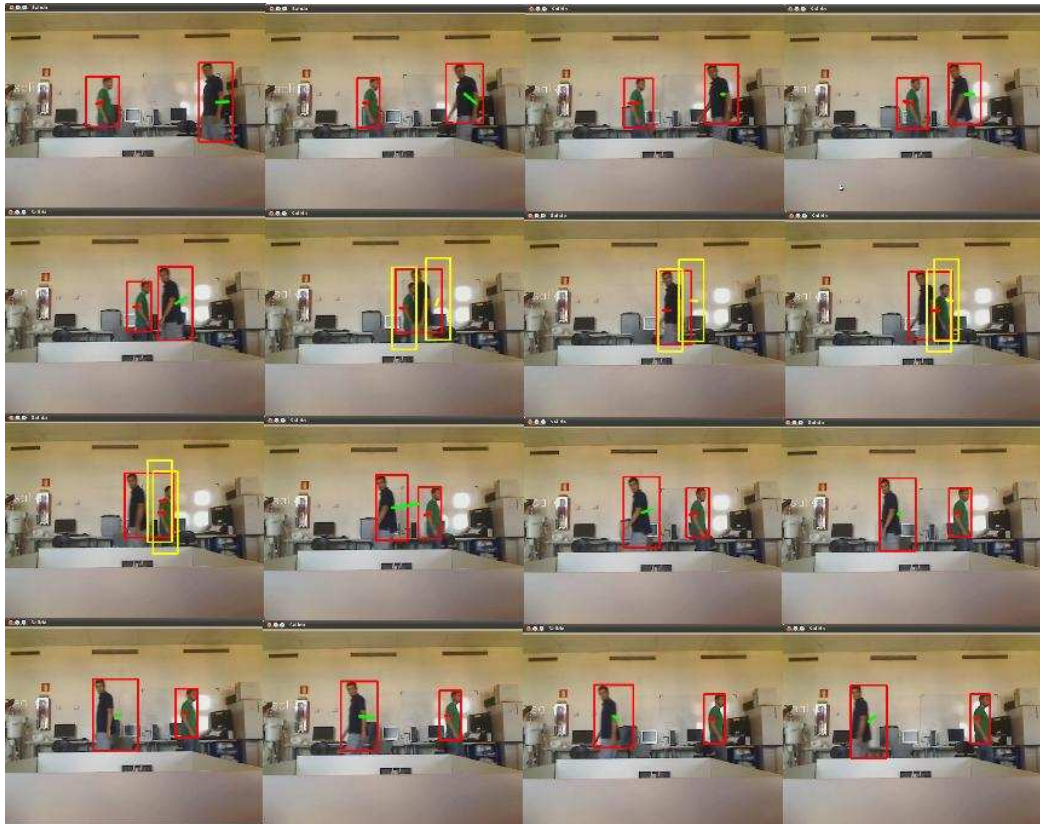


Figura 5.4: Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas, ante la situación de seguir a dos personas que se cruzan.

5.1.4. Seguimiento de dos personas caminando pegadas

En esta situación al igual que para el caso anterior hay un momento en el que el sistema a pesar de que las personas continúan en la escena se queda sin información (detecta a las 2 personas como si fuera una) de forma repentina por lo que el algoritmo asigna como nueva posición de cada una de las personas a la que predice el filtro de Kalman. Para estos casos los resultados obtenidos son mejores que en la situación anterior pues las personas suelen seguir una trayectoria sin cambios repentinos o bruscos. En la figura 5.5 se puede observar como el algoritmo se desenvuelve en una de estas situaciones, las características principales de la experiencia se recogen en la tabla inferior.

Lugar: Tercera planta del edificio Bethancourt.	
Descripción: El vídeo se graba en el pasillo que se encuentra situado en el exterior del laboratorio del Departamento de Robótica y Automática ya que es el típico escenario donde se puede producir esta situación debido a la estrechez y a la cantidad de gente que atraviesa diariamente el mismo.	
Hora: 15:18 h	Vídeo: 0:58 min
Iluminación: Artificial, mediante los fluorescentes situados en el techo del pasillo.	Altura a la que se coloca la cámara: La cámara se sitúa a medio metro de altura.
Parámetros del algoritmo de segmentación: ▪ $L_c = 128$ ▪ $L_{cc} = 64$ ▪ $\alpha_2 = 0.01$	Parámetros de algoritmo de filtrado de personas: ▪ $1.000 < \text{Área} < 90.000$ ▪ $0,4 \leq \text{Ratio} \leq 0,95$ ▪ $0,1 \leq \text{Rec} \leq 1.20$

Cuadro 5.4: Características del estudio para la situación de seguimiento de dos personas que caminan pegadas.

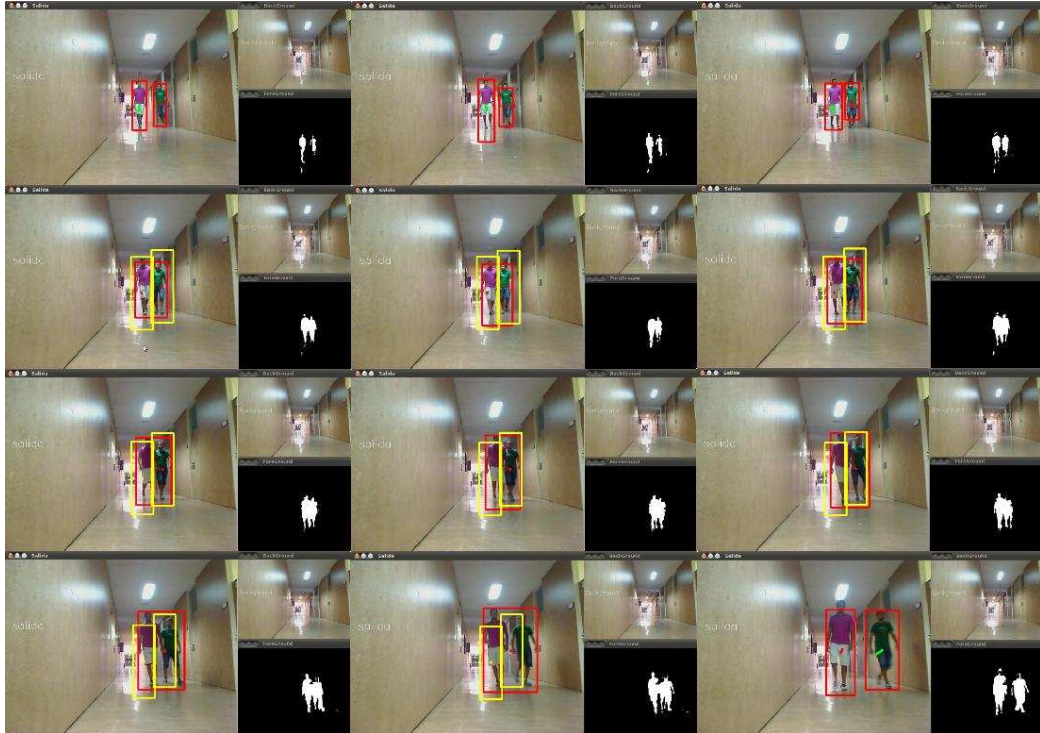


Figura 5.5: Secuencia de imágenes donde se muestra el resultado de aplicar la mejora de tracking de personas, ante la situación de seguir a dos personas que caminan pegadas.

Se puede observar como a partir del frame número cuatro el módulo de detección sólo representa un único blob formado por las dos personas unidas. En esta situación en la que la información obtenida por este algoritmo no es suficiente para realizar el seguimiento de ambas personas se comienza a representar la posición de cada persona únicamente a partir de la predicción realizada por el filtro de Kalman. En el momento en el que las personas se separan lo suficiente como para ser detectadas de manera independiente la aplicación vuelve a su funcionamiento normal, realizando el seguimiento mediante el uso simultáneo del filtro de Kalman y el algoritmo de rasgos descritos en 4.1.

Una vez que se determina el funcionamiento del módulo en las cuatro situaciones anteriores, se debe estudiar ahora un punto que es común a todas ellas en mayor o en menor medida. Hay determinadas situaciones en las que el ambiente no es propicio para utilizar el módulo que determina la posición de las personas únicamente a partir de la predicción que realiza el filtro de Kalman. Imaginad una situación compleja donde el número de falsas detecciones momentáneas debido a ruido o a particiones del blob que representa una persona es elevada. En este caso habrá momentos en el que el filtro de Kalman realice el algoritmo de predicción-corrección de estos falsos positivos, de manera que cuando estas falsas detecciones desaparezcan el programa determinará que nos encontramos ante la situación anteriormente comentada en la que se representará la posición de estos falsos positivos a través únicamente de la predicción realizada por el filtro de Kalman. Es evidente que esto es una situación no deseable, la detección de falsos positivos es en ocasiones inevitable pero el seguimiento de estas a partir de Kalman si lo es. En caso de encontrarse en tales situaciones se recomienda desactivar el módulo de seguimiento en situaciones de falta de información. Con esto no se consigue realizar el seguimiento de personas en situaciones de pérdida de información, pero se evitan falsos seguimientos no deseados.

En la siguiente imagen se puede observar lo comentado, se trata de un frame de uno de los estudios realizados en el aula de informática del edificio Bethancourt donde como se puede observar hay un número elevado de objetos(mesas, sillas, ordenadores...) que hacen que la detección no se realice de forma correcta durante toda la prueba, provocando así que el seguimiento de personas no sea correcto ya que el sistema se queda sin información de forma repentina en numerosas situaciones haciendo actuar al filtro de Kalman para que estime la posición de un individuo que no existe sino que es una parte de otro.

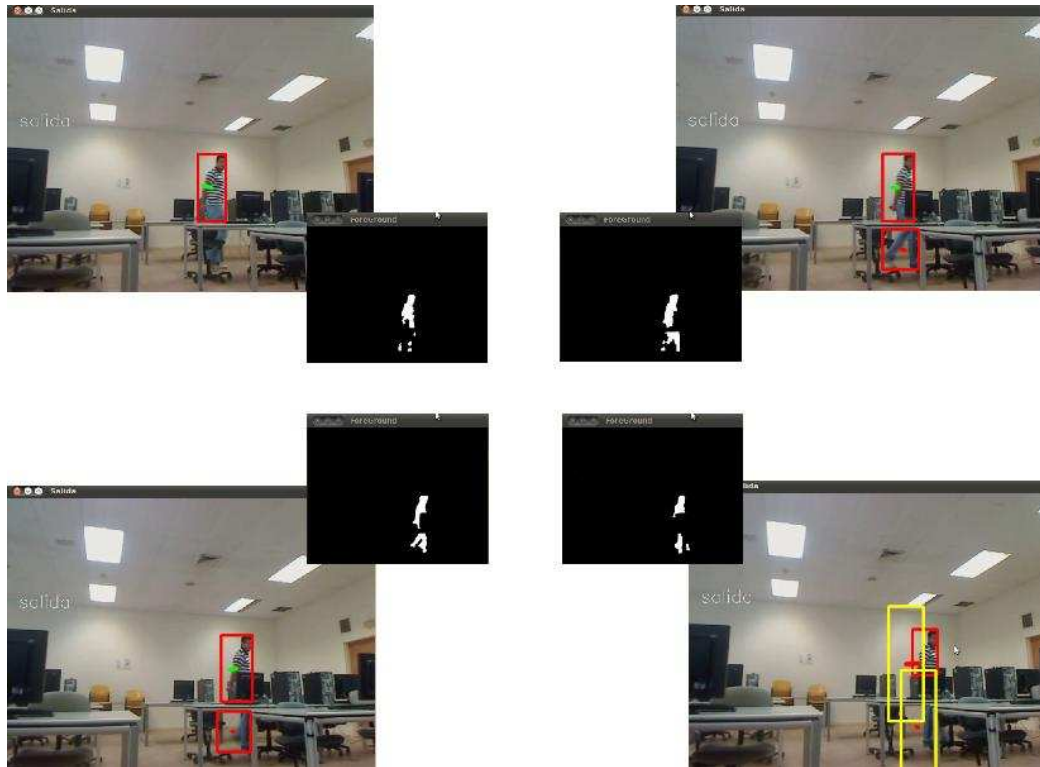


Figura 5.6: Imagen donde se muestra el resultado del tracking en una escena con falsos positivos sin desconectar el módulo de seguimiento en situaciones de falta de información.

En la siguiente experiencia se ha desactivado el módulo, con lo que sólo se hará un seguimiento basado en el filtro de Kalman y en el algoritmo basado en rasgos siempre y cuando haya información. De esta forma se observa que aunque haya ratos donde la personas no sean seguidas en global se consigue un funcionamiento más robusto del sistema.

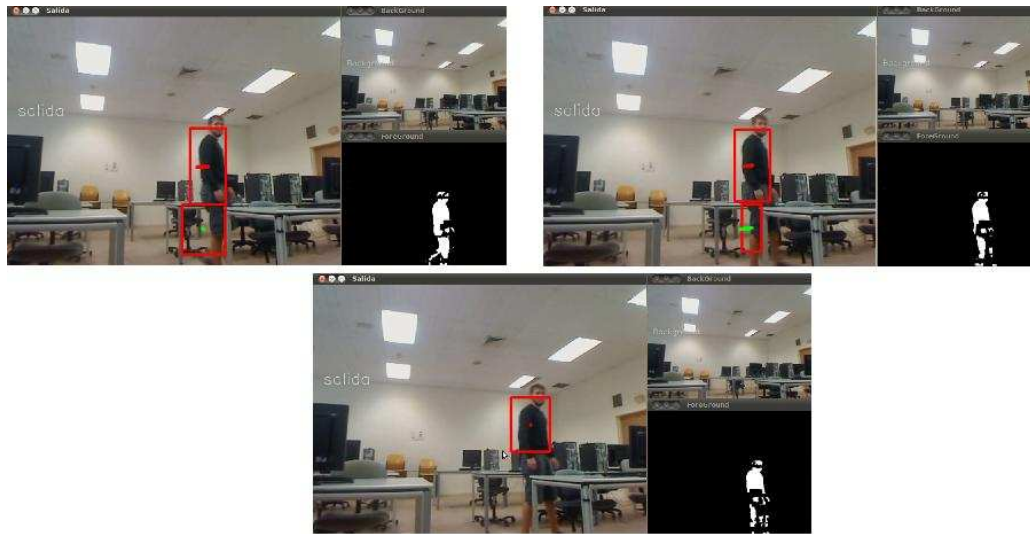


Figura 5.7: Imagen donde se muestra el resultado del tracking en una escena con falsos positivos desconectando el módulo de seguimiento en situaciones de falta de información.

5.2. Experimentación de la mejora: Detección de personas estáticas

El objetivo de este apartado es determinar el éxito del módulo de detección de personas estáticas descrito en 4.2. Para ello se han realizado una serie de pruebas que al igual que en el caso anterior van elevando su nivel de complejidad.

5.2.1. Persona permanece parada de pie o acostada en el interior de un edificio

En este caso todas las pruebas realizadas son positivas, consiguiendo un porcentaje de éxito del 100 %. Este resultado es previsible pues la mejora está diseñada principalmente para trabajar en situaciones de interior y fijar a una persona. En la imagen siguiente se puede observar un ejemplo de esta situación, donde se ve como la persona ha pesar de haber pasado a formar parte del background sigue siendo detectada.



Figura 5.8: Detección de una persona que permanece parada de pie en el interior de un edificio.

5.2.2. Persona permanece parada de pie o acostada en el exterior

En este caso el éxito no es tan rotundo como en interiores, debido a que esta aplicación utiliza el módulo de segmentación desarrollado en [4] que tiene peores prestaciones en exteriores ya que las personas no están bien definidas en el foreground debido a la mala iluminación. Así pues, en raras ocasiones no se llega a fijar a la persona parada y en otros muchos se fija pero con un tamaño algo diferente al real como se puede apreciar en la siguiente imagen.



Figura 5.9: Detección de una persona que permanece parada de pie en un ambiente exterior.

5.2.3. Persona permanece parada sentada en el interior de una habitación

En este caso se obtienen unos magníficos resultados al igual que en la primera situación, es decir un 100 % de éxito siempre y cuando los parámetros de filtrado permitan que la persona sentada sea detectada. Esto se debe al igual que en el primer caso a que la aplicación desarrollada presenta unos excelentes resultados en interiores. En la siguiente imagen se puede observar

esta situación en uno de los muchos ambientes donde fue probada.



Figura 5.10: Detección de una persona que permanece sentada en el interior de una habitación.

5.2.4. Varias personas permanecen paradas a la vez

Debido a la compleja programación necesaria para resolver este caso se comprueba que el algoritmo solo fija a la primera persona, por el contrario el resto serán fijadas mientras permanezcan en el foreground. Mejorar esta situación requiere un coste computacional exponencial lo cual para una aplicación en tiempo real no es positivo. En la siguiente imagen se puede observar el caso explicado en esta situación, donde la persona sentada permanece fijada aun formando parte del background y la que está de pie sólo será detectada mientras forme parte del foreground.



Figura 5.11: Detección de varias personas paradas a la vez.

Una vez que se ha determinado el funcionamiento del módulo en las situaciones de mayor importancia, se deben estudiar dos puntos más que son comunes a todas ellas.

El primero de ellos es determinar si permanece fijada todo el tiempo que está parada. En este caso mientras la persona permanece sola se puede afirmar que sí, mientras que cuando hay más personas que se mueven por delante o por detrás se consiguen muy buenos resultados, fallando en 1 de cada 30 casos aproximadamente. En el caso 1 mostrado en la figura 5.12 se muestra un ejemplo de respuesta positiva. Se puede observar como la persona sentada que está fijada (aparece en el background) sigue siendo representada como estática mientras otro individuo se mueve por delante. Cabe destacar que para que la persona sentada se represente el requisito para filtrar ancho/alto es entre 0.1 y 1.

El siguiente punto a estudiar es si cuando la persona se vuelve a mover esta sigue siendo seguida y deja de permanecer como que está fijada. En este caso siempre y cuando la persona se mueve en dirección no vertical el funcionamiento es el esperado, mientras que cuando la persona se mueve en el eje vertical permaneciendo dentro del rectángulo pero más pequeña sigue estando fijada como estática (ver caso 2 en la figura 5.12).

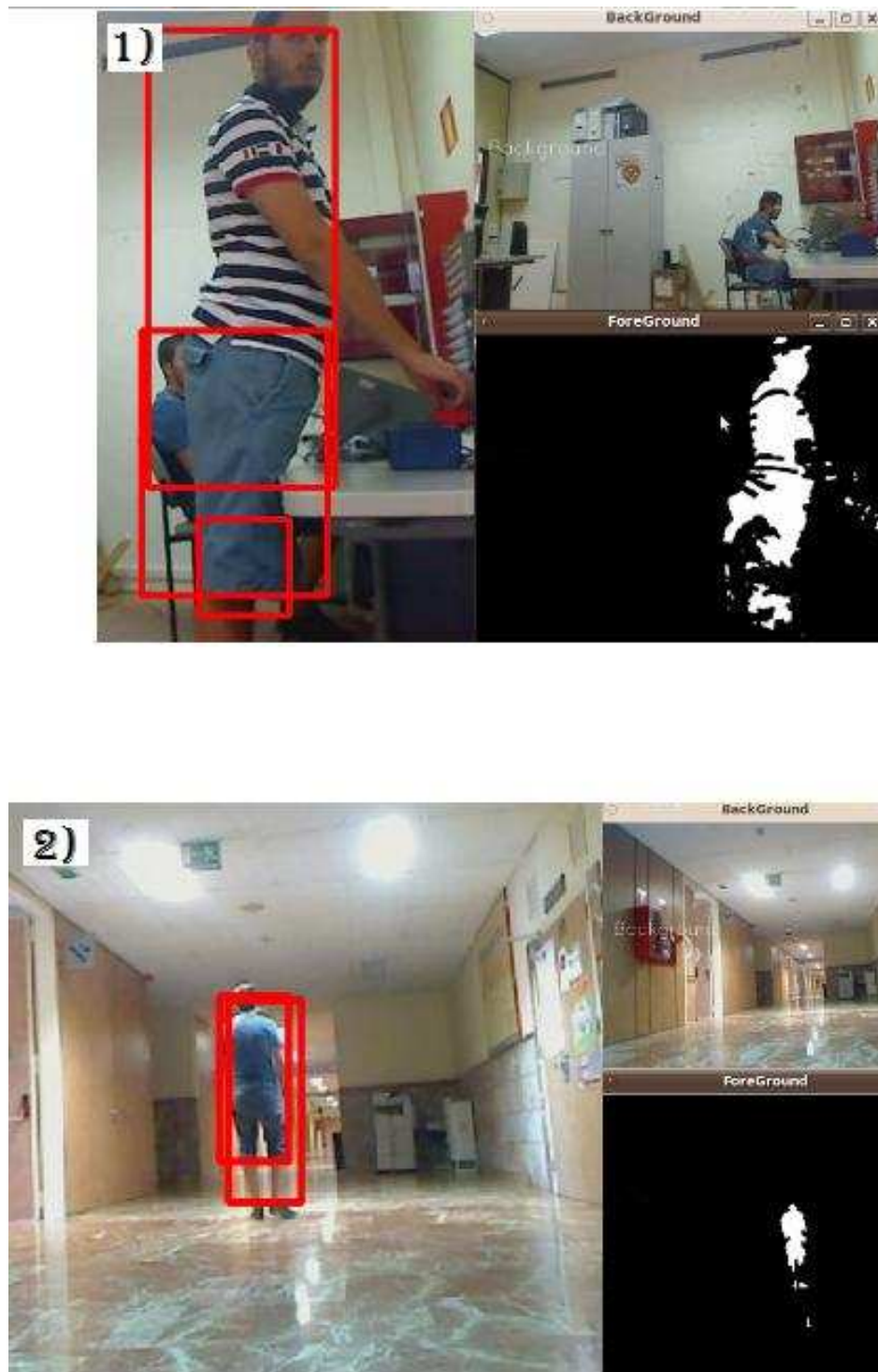


Figura 5.12: Permanencia y salida del módulo de personas estáticas.

CAPÍTULO 6

CONCLUSIONES

Este capítulo se ha creado con la finalidad de presentar las principales conclusiones que se obtienen tras probar la aplicación en distintos ambientes y situaciones. Antes de ello se realiza un breve repaso sobre la aplicación desarrollada.

Este proyecto nace con el objetivo de complementar [4] añadiéndole una serie de mejoras:

- Tracking de personas.
- Detección de personas estáticas.
- Grabación de vídeo.

En la primera de las mejoras se pretende conseguir el seguimiento de dos personas, desarrollando un programa que sea capaz de estimar cual es la posición de cada persona en el siguiente frame. Además de esto también se pretende que la aplicación sea capaz de determinar donde se encuentran las personas incluso en ausencia de datos (ruido, cruces entre ellas,...). Debido a

estas dos circunstancias es necesario seleccionar un filtro de predicción que acompañe a un algoritmo de seguimiento basado en el uso de modelos o no.

Para determinar que algoritmos son los más apropiados para implementar la aplicación se hace un minucioso estudio de las distintas alternativas posibles, llegando a la conclusión de que para el filtro de predicción es mejor usar el filtro de Kalman frente al de partículas debido a su menor coste computacional y su sencillez de implementación, y que es mejor usar un algoritmo de seguimiento no basado en modelos debido a que el programa tiene la intención de seguir personas en diferentes posturas y en entornos complicados, donde el uso de un algoritmo basado en modelos es inviable computacionalmente.

Una vez determinado los algoritmos más apropiados para realizar el seguimiento de personas se desarrolla la mejora y se implementa, de forma que sea lo más óptima posible.

La segunda de las mejoras pretende que se pueda detectar una persona aunque esta lleve un rato detenida y haya pasado a formar parte del background, para lograr este objetivo se implanta un algoritmo basado en comparar cada cierto tiempo si hay una persona igual (o muy parecida), en cuanto tamaño y posición, a la que había hace un determinado número de ciclos. Se aprovecha así que la persona estática permanece un tiempo en el foreground antes de desaparecer. Si el algoritmo realiza esta comparación y sale afirmativa, entonces se determina que hay una persona estática en esa posición y se almacena un blob (persona) con esas dimensiones y posición, por lo que esta persona permanece almacenada, y por tanto se está mostrando hasta que se cumpla la condición de salida.

La tercera mejora introducida pretende guardar el contenido de los vídeos de salida en algún fichero para que de esta manera puedan ser estudiados a posteriori, y así mejorar la aplicación o detectar algún fallo de la misma.

Una vez realizado este breve resumen de cada una de las mejoras añadidas, se pasa a comentar las conclusiones principales obtenidas de la experimentación.

En cuanto a la primera mejora (tracking de personas), una vez se realiza la experimentación en distintos ambientes y situaciones se puede concluir que funciona de forma correcta en aproximadamente el 100 % de los casos siempre y cuando el escenario sea propicio para ello, esto es, buena iluminación, pocas sombras y no más de dos personas. En caso de que el escenario no sea

adecuado debido a la aparición constante de falsos positivos se concluye que es mejor desactivar el módulo de seguimiento ante falta de información ya que aunque con ello quede limitada la aplicación al seguimiento de las personas siempre y cuando se posea toda la información, se consigue una mayor robustez del sistema.

Las conclusiones más importantes de la segunda mejora es que funciona mejor en ambientes de interior que de exterior, que mientras tan sólo sea una persona la que está parada se puede realizar la detección sin comprometer el cálculo computacional para que el resto de la aplicación funcione de manera correcta en tiempo real y que la persona permanece fijada todo el tiempo que está parada en el 95 % de los casos estudiados.

La conclusión principal que se ha obtenido de la tercera mejora es que simplifica mucho la labor de depurar las dos primeras mejoras, pues permite observar una y otra vez los vídeos de experimentación para ver que errores se han cometido y solucionarlos con la opción más óptima.

CAPÍTULO 7

TRABAJO FUTURO

De la aplicación desarrollada se abre un amplio conjunto de posibles mejoras y líneas de investigación para conseguir un comportamiento aún más fiable, con la finalidad de explicar cada una de ellas nace el siguiente capítulo.

En cuanto a la primera mejora (seguimiento de personas) vista en el apartado 4.1 se propone que se realice una actualización de la misma para los casos que hayan más de dos personas, para de esta forma potenciar el uso de la aplicación en más campos para los que fue desarrollada en un principio. Al mismo tiempo también es interesante trabajar en el filtrado de sombras, pues así esta aplicación se puede utilizar en el exterior ya que en la actualidad debido a este problema la aplicación no presenta resultados tan buenos como lo hace en el interior.

Otra línea de investigación futura pero en este caso a nivel general y no específico de la aplicación aquí expuesta es la de mejorar la relación eficiencia-coste computacional de los diferentes algoritmos vistos en el capítulo 4, ya que con ello se puede desarrollar una aplicación mucho más potente.

Para la segunda mejora (detección de personas estáticas) vista en el

apartado 4.3 se propone aumentar el número de personas que son detectadas al estar detenidas, ya que en la actualidad debido a la complejidad del algoritmo necesario y el elevado coste computacional para llevar a cabo esta misión con éxito la aplicación sólo fija a la primera persona y al resto sólo cuando permanezcan en el foreground.

Para la tercera mejora (grabación del vídeo de salida) vista en el apartado 4.3 se propone almacenar en un fichero de datos el número de personas que entran en la escena, así como la trayectoria que sigue cada una de ellas.

BIBLIOGRAFÍA

- [1] Bradski,G.; Kaebler A. «*Learning OpenCV: Computer Vision with the OpenCV library*».
- [2] Página web principal de las librerías OpenCV.
«<http://opencv.willowgarage.com/wiki/>».
- [3] Página web de introducción a las librerías OpenCV.
«<http://futura.disca.upv.es/imd/cursosAnteriors/2k3-2k4/copiaTreballs/serdelal/trabajoIMD.xml>».
- [4] González Benítez,U. «*Detección de personas a través de visión por computador*». Departamento de Ingeniería de Sistemas y Automática, Universidad Carlos III Madrid.
- [5] De la Escalera,A. «*Visión por computador*». Prentice Hall.
- [6] Martínez,J; Herrero,J; Orrite,C. «*Automatic Detection and Tracking using a Multi-camera UKF*». Proc. of PETS 2006, pp 59-66.
- [7] Página web oficial de Ubuntu. «<http://www.ubuntu.com/>».
- [8] Página web del profesor Antonio Sanz Montemayor.
«<http://www.escet.urjc.es/asanz/>».
- [9] Ali,A.; Aggarwal,J.K. «*Segmentation and Recognition of Continuous Human Activity*». Computer and Vision Research Center.

- [10] Cunado,D. «*Using Gait as a Biometric, via Phase-Weighted Magnitude Spectra.*». Department of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, England.
- [11] Lee,L. «*Gait Analysis for Recognition and Classification*». Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
- [12] Shakhnarovich,G. «*Integrated Face and Gait Recognition From Multiple Views.*». Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
- [13] Wang,H.; Suter,D. «*A consensus-based method for tracking: Modelling background scenario and foreground appearance*». Department of Electrical and Computer Systems Engineering, Monash University, Clayton Vic. 3800, Australia
- [14] Gavrilu,D.M. «*3-D model-based tracking of human in action: a multi-view approach.*». Computer Vision Laboratory, University of Maryland.
- [15] Green,R.D. «*Quantifying and Recognizing Human Movement Patterns from Monocular Video Images - Part I: A New Framework for Modeling Human Motion.*». Human Interface Technology Lab, University of Canterbury, Christchurch, New Zealand.
- [16] Chu,C.W.; Nevatia,R. «*Real-Time 3D Body Pose Tracking from Multiple 2D Images.*». Institute for Robotics and Intelligent System, University of Southern California.
- [17] Fuentes,L.M.;Velastin,S.A. «*People tracking in surveillance applications*».
- [18] Do,Y. «*Tracking People in Video Camera Images Using Neural Networks*».
- [19] Niyogi,S.A.; Adelson,E.H. «*Analyzing gait with spatiotemporal surfaces*».
- [20] Antonini,G.; Venegas,S. «*Priors for Detection and Tracking of Pedestrians in Video Sequences*».
- [21] Seitner,F.H.;Hanbury,A. «*Fast pedestrian tracking based on spatial features and colour*».

BIBLIOGRAFÍA

- [22] Rodríguez,P. «*Aplicación del filtro de Kalman al seguimiento de objetos en secuencias de imágenes*». Departamento de Informática, Universidad Rey Juan Carlos.
- [23] Guillies,D; Zhen,Y. «*Dynamic Interactions with the Philosophy of Mathematics, Theoria*».
- [24] Kalman,R.E.. «*A new approach to linear filtering and prediction problems*».
- [25] Morris H. «*Probabilidad y Estadística*». Segunda Ed. Addison Wesley Iberoamericana.
- [26] Isard,M; Blake,A. «*CONDENSATION: Conditional density propagation for visual tracking*». International Journal of Computer Vision.
- [27] MacCormick,J; Blake.A. «*Probabilistic exclusion and partitioned sampling for multiple object tracking*».