

1.- MOTIVACIÓN Y OBJETIVOS:

1.1.- Introducción:

El seguimiento de siluetas y objetos dentro de secuencias de video, también conocido como “tracking”, es una de las tareas más significativas dentro del área de la visión artificial. Esto se debe, principalmente, al amplio abanico de aplicaciones que la precisan para su correcto funcionamiento, como son la detección de movimiento, sistemas de seguridad y vigilancia, monitorización de escenas, control de robots manipuladores, guiado automático de vehículos, aplicaciones relacionadas con la bioingeniería (eye-trackers, desarrollo de interfaces persona-ordenador), etc. Es por esta razón y por el auge de las tecnologías basadas en el procesamiento digital de imágenes que el seguimiento de objetos ha sufrido un notable avance durante estos últimos años.

Esta disciplina puede englobarse dentro de la “inteligencia artificial”, también conocida como AI, del inglés “artificial intelligence”, la cual comenzó a desarrollarse a partir de los años 70. El objetivo principal de la misma es la de dotar de habilidades humanas a las máquinas, es decir, crear programas para ordenadores o robots para conseguir que éstos imiten la comprensión y el comportamiento humano.

Centrándonos en la visión artificial, las habilidades buscadas eran la capacidad de ver objetos, localizarlos y reconocerlos. Debido a la gran cantidad de información captada en una sola imagen, esta tarea no resulta trivial, complicándose aún más si lo que tenemos que procesar es una secuencia de video en vez de una imagen estática, ya que a las limitaciones propias de las técnicas de visión estáticas tenemos que añadir el hecho de tener que procesar una cantidad de información mucho mayor y de tener que hacerlo en tiempo real en la mayoría de las aplicaciones.

Suponiendo un sistema con una estructura de datos eficiente y robusta y unos algoritmos que la aprovechen para un rápido procesamiento de la información, la dificultad en el seguimiento de los objetos vendrá dada, principalmente, por los cambios que se puedan producir en los mismos. Estos cambios se pueden clasificar en:

- Cambios morfológicos: Donde los objetos pueden presentar modificaciones en su forma.
- Cambios fotométricos: Donde la iluminación de la escena que se está analizando es variable, afectando así a la percepción visual que se tiene del objeto.
- Desplazamientos y giros.

Muchos de los estudios realizados durante los comienzos de esta disciplina se basaban en el seguimiento de objetos rígidos (que no modifican su forma); sin embargo, en la mayoría de los casos que nos encontramos en la vida real, los objetos en movimiento son objetos cambiantes (personas, animales, etc...), por lo que en los últimos años los estudios se han centrado en sistemas de seguimiento robustos frente a cambios morfológicos.

Además de los cambios que pueda sufrir el objeto a seguir, existen otras circunstancias que nos afectarán directamente en el desarrollo de nuestro sistema. Por un lado, el número de objetos a seguir es un factor a tener en cuenta, ya que trabajar con múltiples objetos conlleva una mayor carga computacional y una política de procesamiento distinta. Y por otro lado, no podemos olvidar los cambios en las características del entorno, como son la aparición de sobras, oclusión del objeto a seguir por otros elementos, cambios en la iluminación y en el escenario donde se sitúa el objeto, los cuales provocarán cierta incertidumbre que tendrá que ser corregida adecuadamente. Existen muchos sistemas que trabajan en entornos conocidos y estructurados, evitando así los problemas que se acaban de mencionar; sin embargo, estos sistemas presentan un ámbito de funcionamiento muy reducido, siendo preferible la implementación de sistemas robustos frente a estos cambios a costa de un aumento en el tiempo de computación.

Todas las consideraciones anteriores se centran en la tarea de tracking o seguimiento, sin embargo, en gran parte de las aplicaciones esta tarea sólo es una parte de otra de más alto nivel, donde no sólo se detecta el movimiento y se sigue el objeto, sino que además se analiza dicho movimiento, obteniendo datos como la posición, la orientación, la velocidad y la dirección del flujo del movimiento. Estos datos podrán ser utilizados a posteriori en aplicaciones como el control del movimiento de robots, la detección de caídas o la creación de interfaces persona-computador como los “eye-trackers”, basados en el control del movimiento de la pupila.

1.2.- Motivación

Teniendo en cuenta las consideraciones anteriores, el objeto principal de este proyecto fin de carrera comenzó siendo la creación de un software para el seguimiento de objetos basado en el estudio e implementación de los distintos algoritmos existentes para este propósito.

Tras realizar un estudio sobre el estado del arte de la materia, que más adelante desarrollaremos, se vio que la cantidad de métodos existentes para realizar el seguimiento de objetos mediante procesamiento digital de imágenes era muy amplia, siendo los algoritmos muy distintos entre sí, dependiendo de las necesidades de la aplicación que se pretendiera abordar. Esta razón unida al hecho de que la empresa Secosol Soluciones y Domótica S.L. estaba interesada en realizar una colaboración con el Dpto. de Ingeniería de Sistemas y Automática propició la idea de desarrollar una primera versión de un software para la detección de caídas de personas discapacitadas o de la tercera edad, consiguiendo así una serie de especificaciones que concretaron nuestro área de trabajo y que a su vez lo dotaron de un carácter más práctico y no meramente teórico. Lógicamente, la posibilidad de diseñar un sistema destinado a aumentar la calidad de vida de personas discapacitadas resultó ser una motivación añadida a la realización de este proyecto.

1.3.- Objetivos

La naturaleza de este proyecto fuerza una serie de restricciones y especificaciones que nos marcarán unos objetivos muy concretos. Estos objetivos principales se detallan a continuación:

- En general, el objetivo de un sistema de seguimiento visual consiste en determinar para cada una de las imágenes de la secuencia de

video, los parámetros de movimiento de la escena en su conjunto o sólo de aquellos objetos que resultan de interés para el sistema a desarrollar. En concreto, para la detección de caídas, el objetivo general está en el seguimiento de determinadas partes del cuerpo de la persona en cuestión, determinando en todo momento su posición, orientación, dirección y velocidad del flujo del movimiento. Es importante resaltar que al ser una aplicación ejecutada en tiempo real, nos encontraremos también con unas condiciones muy restrictivas en cuanto a los tiempos de computación, lo cual nos delimitará las posibles técnicas que podremos emplear.

- Como acabamos de comentar, el tiempo de computación en estos sistemas va a ser muy importante, pero no sólo por el hecho de trabajar en tiempo real, sino por la necesidad de no perder al individuo monitorizado en ningún instante, pues una caída es un movimiento de muy alta velocidad y por lo tanto es bastante probable que se pierdan los elementos que estamos siguiendo si el procesamiento de cada imagen es lento. Es por esta razón que la interfaz gráfica del proyecto es sencilla, debido a que una más elaborada consume demasiados recursos y hace que el sistema pierda eficiencia. Así pues, nos centraremos principalmente en la programación de los algoritmos de detección haciendo que éstos sean rápidos y robustos y dejaremos para mejoras futuras la programación de una interfaz gráfica mejor.
- Por otro lado, debido a que la aplicación está basada en el procesamiento digital de imágenes, la elección del hardware empleado para la captación de las mismas es un aspecto primordial. Debido a que el sistema formará parte de una futura instalación domótica, la solución adoptada utiliza una cámara web, ya que éstas presentan un bajo coste y son muy comunes dentro del mercado de consumo de los ordenadores personales.
- Un aspecto que no debemos olvidar es el hecho de que estamos trabajando con personas y que un sistema para la detección de caídas mediante procesamiento de imágenes obliga a que la aplicación esté activa siempre y que la persona esté siendo grabada en todo momento. Es por eso que se nos planteó la necesidad de hacer un procesamiento lo más autónomo posible, es decir, donde

la intervención de un operario para su correcto funcionamiento sea mínima y así garantizar la intimidad de la persona.

- El seguimiento de objetos o personas mediante el procesado de imágenes viene comúnmente acompañado de una serie de incertidumbres que hacen aún más complicada esta tarea, como son una iluminación no uniforme, aparición de brillos y sombras, situaciones de oclusión, distorsión en la apariencia de los elementos a seguir como consecuencia de su movimiento, efectos de perspectiva o incluso movimientos que violan de forma esporádica los criterios generales asumidos por los modelos dinámicos establecidos. Todas estas circunstancias provocan errores en el seguimiento, llegando a producir con cierta facilidad la pérdida definitiva del objeto seguido. En muchos sistemas, con la intención de simplificar y eliminar estas incertidumbres se suele considerar un entorno de trabajo conocido y estructurado; sin embargo, si tenemos en cuenta que este software se va a emplear en un futuro en entornos reales no estructurados, es importante marcar como objetivo que nuestra aplicación incorpore mecanismos capaces al menos de detectar la aparición de este tipo de errores y, en la medida de lo posible, que sea capaz de compensarlos y corregirlos para su correcto funcionamiento.
- Por último, y en relación al punto anterior, otro de los objetivos propuestos para este proyecto fin de carrera es conseguir minimizar al máximo la ocurrencia de falsas alarmas. Una falsa alarma en nuestro caso sería la detección de una caída que realmente no ha ocurrido. En un futuro, la idea es incluir este software dentro de un sistema global de domótica donde se avise a quien sea oportuno (un familiar, una enfermera, una ambulancia, etc) en caso de detectar una caída, por lo que dotar al software de redundancia para evitar estas falsas alarmas será muy importante. Lógicamente, esta redundancia no debe ser excesiva, pues no olvidemos que toda la carga computacional que agreguemos al sistema provocará una ralentización y pérdida de eficiencia del mismo.

Una vez claros los objetivos principales del proyecto pasaremos a detallar cómo se ha estructurado la memoria para facilitar su lectura.

1.4.- Organización de la memoria

La memoria se ha estructurado en seis capítulos principales y tres anexos. Comenzando por los capítulos, el primero de ellos “Motivación y objetivos” en el cual nos encontramos describe por un lado las motivaciones y objetivos que pretendemos alcanzar con la realización de este proyecto fin de carrera y por otro lado hace una breve explicación de las partes de las que va a constar la memoria.

El segundo capítulo “Introducción teórica”, realiza primero una explicación global del estado del arte de la visión por computador y el procesamiento digital de imágenes para luego especializarnos en las áreas de visión centradas en la segmentación, seguimiento y análisis del movimiento, que son las tareas que nos interesan para la programación de un detector de caídas. En este apartado también se muestra una descripción del método de trabajo a seguir para conseguir nuestros objetivos y se plantea de forma amplia todo el conjunto de algoritmos existentes en la actualidad para implementar la aplicación, comparándolos entre ellos y decidiendo cuáles son los más apropiados para nuestro sistema. Finalmente, una vez seleccionados los algoritmos que vamos a utilizar se detallan todos sus fundamentos teóricos y matemáticos.

El capítulo tercero “Componentes del sistema” describe los elementos, tanto a nivel hardware como software, pertenecientes al entorno de trabajo. Entre ellos se encuentran la cámara web, equipo informático, entorno de programación, librerías y se han analizado también las fuentes de iluminación empleadas, por tratarse de un sistema de visión y por lo tanto ser éstas parte fundamental de la aplicación.

En el siguiente capítulo, “Implementación”, se muestra el funcionamiento del sistema final en sí, donde se incluyen todos los diagramas de flujo, la estructura interna del sistema así como las funciones propias y de librería utilizadas en la programación. Dentro de este capítulo también mostramos capturas de pantallas de la ejecución del programa para comprender mejor el funcionamiento de la aplicación.

En el quinto capítulo “Conclusiones y líneas de desarrollo futuras” se explican las conclusiones alcanzadas tras el diseño, implementación y pruebas del sistema y en base a las deficiencias del mismo se detallan las futuras líneas de desarrollo en las que seguir trabajando para mejorar la aplicación.

En el último de los seis capítulos, “Bibliografía” se indican todas las fuentes de información de donde se han extraído los datos necesarios para la

realización del proyecto y de esta memoria. Dentro de estas fuentes de información encontraremos principalmente artículos, tutoriales y multitud de enlaces a sitios web donde se dispone de información y material adicional.

En cuanto a los anexos, en el primero de ellos “Anexo A: Instalación y configuración de OpenCV” hemos detallado los pasos a seguir para la instalación y puesta en marcha de esta librería especializada en el procesamiento digital de imágenes y que es la que hemos utilizado en la programación de la aplicación. Dentro de este anexo también detallamos los pasos que hemos seguido para configurar el entorno de programación Microsoft Visual Studio C++ 2005.

El segundo de los anexos, “Anexo B: Versión de prueba con *Background Subtraction*”, es similar al capítulo cuarto de nuestra memoria, pero esta vez centrado en otro método distinto para el diseño de la aplicación. Esto es así debido a que en primer lugar se intentó realizar el sistema empleando el método de “Background subtraction”. Al no obtener los resultados esperados, puesto que el sistema a pesar de su robustez no era capaz de detectar caídas, se implementó una nueva versión utilizando la técnica “Camshift” con la que se obtuvieron mejores resultados siendo así la definitiva y la que se ha explicado en el capítulo 4. En este anexo, a diferencia del capítulo cuarto, también se introducen los códigos de esa primera aplicación y un breve manual de instalación de la librería CBlobsLib que fue necesaria para la programación.

Y por último “Anexo C: Código, ficheros y resultados”, donde incluimos el resto de recursos asociados a la aplicación definitiva.