

5.- CONCLUSIONES Y LÍNEAS DE DESARROLLO FUTURAS:

5.1.- Estructura:

En este capítulo vamos a desarrollar las conclusiones a las que hemos llegado después de la realización del proyecto fin de carrera. Para ello vamos a comparar los objetivos iniciales del proyecto, detallados en el capítulo 1, con los objetivos que realmente hemos alcanzado, viendo así cuáles son las posibles deficiencias que presenta nuestro diseño y poder plantear una serie de mejoras para posteriores versiones de la aplicación.

Además, dentro de este capítulo también desarrollaremos posibles líneas de avance en el campo de la detección de caídas mediante procesamiento digital de imágenes.

5.2.- Conclusiones

En primer lugar, antes de exponer las conclusiones a las que hemos llegado después de la realización del proyecto, vamos a realizar una comparación de cuáles fueron los objetivos propuestos inicialmente y cuáles han

sido los objetivos realmente alcanzados. Recordaremos uno a uno cada uno de ellos y los iremos analizando.

- *“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.”*

El párrafo anterior nos muestra el primer objetivo propuesto. Teniendo en cuenta los resultados obtenidos, podemos concluir que este objetivo se ha cumplido satisfactoriamente, ya que con nuestra aplicación tenemos un control total en todo momento de la cabeza y el torso (que son las partes en las que nos hemos querido centrar), conociendo su posición y orientación mediante el algoritmo Camshift y conociendo su dirección y velocidad del flujo de movimiento mediante el método de Lucas&Kanade.

Por otro lado, hemos visto que con la aplicación definitiva, al contrario que con el método de Background Subtraction, se consigue un procesamiento de hasta 30 frames por segundo, suficientemente alto para nuestro sistema por lo que el objetivo de procesamiento en tiempo real se ha conseguido también.

- *“...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...”*

El segundo de los objetivos, como vemos, consiste en conseguir un sistema 100% fiable donde no se pierda al individuo monitorizado en ningún instante. Este objetivo desgraciadamente no ha sido alcanzado, pues nuestro sistema no es lo suficientemente robusto como para evitar pérdidas en el seguimiento. Como se comenta en los resultados expuestos en el capítulo cuarto, trabajando

con una cámara capturando a 30 frames por segundo, de 20 pruebas sobre el sistema, sólo 14 se detectaron satisfactoriamente, por lo que nuestro sistema presenta aproximadamente un 70% de éxito actualmente. En el siguiente apartado del capítulo detallaremos una serie de mejoras que podemos aplicar para conseguir que este porcentaje de éxitos aumente considerablemente.

- *“...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.”*

Como se desarrolla en el capítulo tercero, los componentes que forman parte de nuestra aplicación cumplen los requisitos expuestos anteriormente, pudiéndose implantar este sistema de forma sustancialmente económica. Alcanzamos así otro de los objetivos propuestos.

- *“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.”*

En nuestra aplicación, una vez inicializada y puesta en marcha, no se precisa la intervención de ningún operario para el correcto funcionamiento del mismo ni tampoco se precisa la intervención del individuo monitorizado, consiguiendo así que nuestro sistema sea bastante autónomo, mantenga íntegra la intimidad del individuo (pues las imágenes sólo son procesadas y vistas por el computador) y además no sea intrusivo, pues el individuo no tiene que llevar dispositivos especiales como ocurre en las aplicaciones que usan sensores o acelerómetros para la detección de caídas.

Sin embargo, otro aspecto que sí afecta directamente a la autonomía de la aplicación es el hecho de que aún no tenga implementado un sistema de recuperación ante fallos (en caso de pérdida de los elementos a seguir), lo cual hace que el sistema no sea capaz de ponerse en marcha de nuevo sólo. Ésta entre otras será una de las mejoras que propondremos.

- *“...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.”*

En el párrafo anterior se habla de dos aspectos diferentes. En primer lugar se habla de implementar un sistema que no considere el entorno como conocido y estructurado (con escenarios fijos, iluminación controlada, etc); este primer aspecto se ha tenido en cuenta en nuestra aplicación, de tal forma que el escenario en el que se mueve el individuo puede evolucionar de forma dinámica sin que esto afecte al seguimiento, además, las técnicas empleadas son también bastante robustas ante situaciones de cambios de iluminación, efectos de perspectiva, situaciones de oclusión (si éstas no duran un tiempo excesivo) e incluso distorsión en la apariencia de los elementos a seguir. Por otro lado se hace mención a la detección de situaciones de pérdida del elemento y a la recuperación del sistema ante estos fallos. En nuestro caso podemos decir que el sistema aún no es capaz de detectar y solventar esta situación para recuperarse.

- *“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...”*

Sabemos que cuando hablamos de falsas alarmas nos referimos al hecho de que el sistema detecte una caída cuando realmente ésta no ha pasado. Hemos de decir que de todas las pruebas realizadas en las que no hubo pérdida del elemento a seguir no hemos detectado ninguna situación de estas características. Sin embargo, en situaciones de error donde el foco del tracker ha pasado por ejemplo de la cabeza al brazo (puesto que ambos presentan la tonalidad de la piel y por lo tanto el sistema los ha confundido), si el brazo de movía a gran velocidad hacia abajo el sistema consideraba la posibilidad de caída, originándose una falsa alarma a consecuencia de la pérdida del foco del

elemento de interés. Con estos datos podremos concluir que, si bien no podemos garantizar que la aplicación esté libre de falsas alarmas, sí es cierto que la aparición de éstas es bastante pequeña en casos donde no ha habido situación de error de seguimiento.

Como hemos visto, los resultados obtenidos se aproximan bastante a los objetivos iniciales, sin embargo, el sistema tiene aún numerosas deficiencias y por lo tanto es susceptible a mejoras que detallaremos en el siguiente apartado.

5.3.- Líneas de desarrollo futuras

Como líneas de desarrollo futuras para la aplicación desarrollada en este proyecto fin de carrera vamos a destacar dos ramas fundamentales. Una primera rama centrada en mejorar y eliminar las deficiencias del sistema actual y una segunda rama centrada en añadirle nuevas funcionalidades para que este sistema pueda ser integrado y funcionar en aplicaciones domóticas reales.

5.3.1.- Mejoras a la aplicación actual

Basándonos en las conclusiones del apartado anterior hemos visto que las mejoras que tendríamos que realizar sobre el sistema se basan en aumentar el porcentaje de éxito en la detección de caídas, conseguir un sistema más autónomo en cuanto a su inicialización, gestionar un sistema de detección y recuperación ante fallos y disminuir la ocurrencia de dichos fallos.

5.3.1.1.- INICIALIZACIÓN DEL SISTEMA:

En el capítulo cuarto vimos que actualmente el sistema precisa de una inicialización de los elementos a seguir para poder ponerse en funcionamiento. Como ya sabemos, esta inicialización consiste en marcar mediante el ratón al elemento que queremos seguir, para que el sistema pueda computar el histograma del plano H de los píxeles marcados. Una vez que ya tenemos el histograma el programa se pone en marcha automáticamente, sin precisar ninguna otra actuación para realizar el seguimiento y monitorización correctamente.

Sabemos, por la información extraída del artículo “*Computer Vision Face Tracking For Use in a Perceptual User Interface*” de Gary R. Bradsky donde se explican los fundamentos teóricos del algoritmo Camshift, que el color de la piel de cada persona, sea de la raza que sea, presenta un histograma H muy similar. Gracias a esta característica, la necesidad de realizar un marcado inicial de la cara del sujeto para obtener su histograma desaparecería, pues con un histograma genérico del color de la piel ya funcionaría la aplicación correctamente.

En base a esto, una mejora interesante consistiría en introducir una fase de aprendizaje previa, de tal forma que introduzcamos como histograma inicial aquel proporcionado por la media de varios histogramas de personas distintas (a poder ser con colores de piel distintos). De esa forma, la inicialización de la aplicación consistiría en considerar como ventana de búsqueda la imagen completa y como histograma el almacenado, consiguiendo una inicialización del programa totalmente autónoma.

5.3.1.2.- INCREMENTO DEL PORCENTAJE DE ÉXITO EN LA DETECCIÓN DE CAÍDAS:

La principal causa por la que el éxito en la detección de caídas no es más alto en la aplicación actual es la pérdida momentánea del elemento a seguir. Es decir, que si queremos implementar como mejora un aumento en el porcentaje de la detección de caídas tendremos que disminuir todo lo que podamos los errores de seguimiento.

Los tres errores de seguimiento más comunes que hemos encontrado durante el funcionamiento de la aplicación se detallan a continuación, proponiendo además algunas soluciones que podrán ayudarnos a solventarlos.

A.- TRANSFERENCIA DEL FOCO DE ATENCIÓN DE SEGUIMIENTO DEL ELEMENTO DE INTERÉS A UN ELEMENTO DISTRACTOR.

Este error consiste en que cerca del elemento de interés pueden existir distractores con un histograma del plano H similar al del objeto a seguir, haciendo que la aplicación los confunda y por lo tanto comenzando a seguir al objeto erróneo.

Este tipo de pérdida es la más común y por lo tanto es la más importante de eliminar. Proporcionaremos a continuación algunas mejoras para tratar de evitarla:

- Aplicar otros algoritmos de forma paralela para **dotar de redundancia al sistema**, como por ejemplo la utilización de detectores de caras o el modelado del fondo de la imagen para identificar los elementos que pertenecen a la escena y no al individuo.
- Mantener la proporcionalidad del tamaño de las ventanas de búsqueda, es decir, en caso de que la aplicación detecte dos *blobs* cercanos pero distintos con el mismo histograma H (ej: cuando el individuo pasa el brazo por delante de la cara), elegir aquel *blob* cuyo tamaño de ventana sea igual o proporcional al que se está siguiendo.

Explicado con otras palabras, si nuestro elemento de interés presenta un tamaño de ventana $W_1 \times H_1$ y se le acerca un distractor con un tamaño $W_2 \times H_2$ no proporcional al anterior, en caso de que el sistema tenga que elegir entre ambos, descartaríamos el segundo por no presentar un tamaño acorde con el objeto que estamos siguiendo.

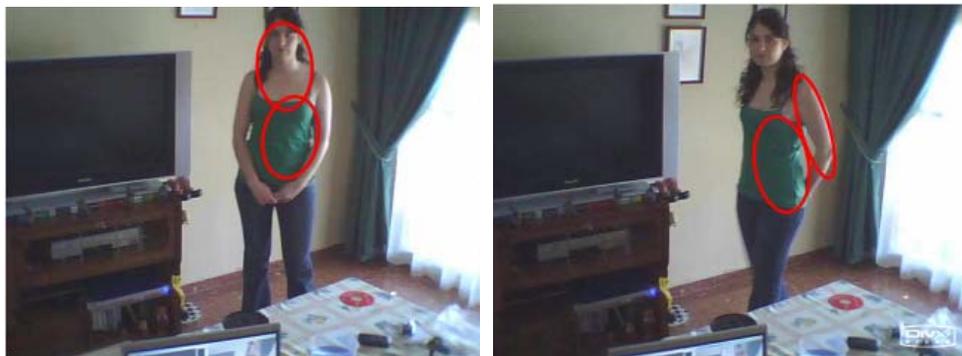


Figura 5.1.- Situación de pérdida del objeto. Cambio de foco

Indicar que consideraríamos como válidas las ventanas con tamaños $a \cdot W_1 \times a \cdot H_1$ ya que el individuo monitorizado puede acercarse o alejarse de la cámara, haciendo que el tamaño de la ventana cambie de forma proporcional.

B.- PÉRDIDA DEL ELEMENTO DE INTERÉS OCASIONADO POR DISCONTINUIDADES.

En este caso lo que ocurre es que debido a una discontinuidad en la trayectoria el sistema no es capaz de seguir al objeto, produciéndose la pérdida del mismo. Esto es así debido a que una de las consideraciones iniciales fue la suposición de procesamiento de movimientos continuos, por lo que el sistema no es capaz de solventar estas situaciones.

Estas discontinuidades se producen principalmente debido a que la velocidad de procesamiento de la aplicación no es lo suficientemente alta. Actualmente hemos alcanzado una velocidad entre los 25 y los 30 frames por segundo, pero si fuésemos capaces de optimizar el código todavía más obtendríamos un rendimiento algo mayor y por lo tanto una reducción de estas discontinuidades. Por lo tanto, la optimización del código sería otra posible mejora.

C.- PÉRDIDA DEL ELEMENTO DE INTERÉS OCASIONADO POR OCLUSIONES.

Existen muchas situaciones en las que el elemento que estamos siguiendo se ve oculto produciéndose por lo tanto su pérdida momentánea o definitiva.

Una mejora que podríamos introducir en la aplicación para eliminar estas pérdidas es la introducción de otra cámara en la habitación, situada diametralmente opuesta a la primera y por lo tanto cubriendo las zonas que se escapan con una sola cámara.

Con la introducción de otra cámara, evitaríamos los problemas que ocurren cuando el individuo da la espalda a la cámara, cuando pasa por detrás de objetos que le ocultan (electrodomésticos, muebles, etc) y además aportaría redundancia al sistema minimizando considerablemente también los otros tipos de pérdidas.

5.3.1.3.- Recuperación ante fallos:

Una de las mejoras más importantes e interesantes que se deben incluir en un futuro para aumentar la autonomía y fiabilidad de la aplicación es la introducción de un método para la detección y recuperación ante fallos.

Ese método consistiría principalmente en un bloque de código que se ejecutaría sólo en caso de detección de fallo y que consistiría en una reinicialización del algoritmo Camshift haciendo que la ventana de búsqueda sea la ventana completa y que de todos los *blobs* que concuerden con el histograma H inicial se tome aquel que siga el patrón de “cara” (utilizando un detector de caras, que es otra de las mejoras ya propuestas).

Una vez encontrada la cara, conociendo su orientación y tamaño, encontrar la situación del torso ya sería una tarea sencilla utilizando los conceptos proporcionados por el *Hombre de Vitruvio* de Leonardo da Vinci, los cuales afirman que si la cabeza tiene una proporción de $H \times H$ (ancho x alto), el torso tendrá un tamaño de $2H \times 3H$.

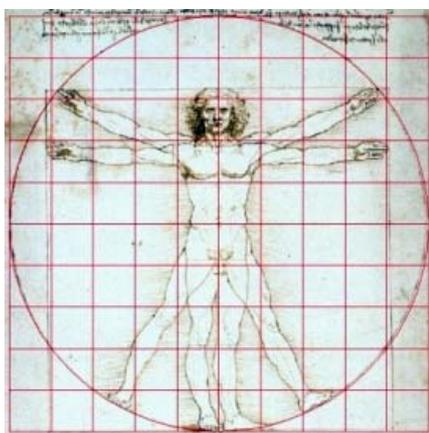


Figura 5.2.- El hombre de Vitruvio

Además, utilizando estos conceptos y sabiendo que la posición de la cabeza está centrada entre los hombros, para casos en los que se produzca la pérdida de la cara y no la del torso su recuperación será sencilla.

5.3.1.4.- Minimización de falsas alarmas:

Para conseguir minimizar las falsas alarmas (detección de caídas cuando realmente no se han producido) propondremos dos posibles mejoras:

- Introducción de un algoritmo de **reconocimiento de posturas**, de forma que si se detecta un movimiento brusco analicemos la imagen y veamos si la posición de la cara y el torso concuerdan con las que tendría alguien que se hubiese caído.
- Introducción de un algoritmo de **análisis de trayectorias**, de tal forma que si la persona monitorizada realmente se cae y

a continuación se vuelve a levantar sin problemas la aplicación sea capaz de detectarlo y no lance la alarma.

5.3.2.- Modificaciones para su introducción en un sistema domótico real.

Las líneas de trabajo vistas hasta el momento se centran en las deficiencias actuales de la aplicación y en intentar solventarlas. En este apartado vamos a mostrar de forma breve tres líneas de investigación futuras centradas en la fase de gestión de las alarmas para poder incluir la aplicación en un sistema domótico real.

Nos hemos querido centrar en esta línea de investigación debido a que en la actualidad el auge de las tecnologías de la información y la evolución e integración de sistemas electrónicos e informáticos está haciendo que el campo de la domótica obtenga un crecimiento exponencial. Además, en lo referente al área de la visión por computador, cada día hay disponibles en el mercado más y mejores cámaras y equipos informáticos, que hacen posible una implantación de estos sistemas más fácil, cómoda y robusta.

Nuestras propuestas son las siguientes:

- Mejora de la interfaz gráfica de usuario.
- Implementar un sistema que, una vez detectada la caída, se encargue de enviar un mensaje a un terminal (teléfono móvil, BlackBerry, etc) para informar de la alarma, mostrando por ejemplo la fotografía capturada por la cámara justo después del instante de la detección de la situación de riesgo.
- Otra posibilidad alternativa es la utilización de la aplicación con una o varias cámaras más sofisticadas que puedan ser controladas de forma remota en base al estándar uPNP desde un terminal, permitiéndose así mover las cámaras y orientarlas en cualquier dirección, aplicar zoom, etc y comprobar así si existe una situación de peligro real. La solución sería más cara pero ampliaría el abanico de posibilidades.