INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

Escuela Superior de Ciencias Experimentales y Tecnología

Curso académico 2007-2008

Proyecto Fin de Carrera

Seguimiento 3D visual de múltiples personas utilizando un algoritmo evolutivo multimodal

Tutor: José María Cañas Plaza

Autor: Sara Marugán Alonso

"No nos atrevemos a muchas cosas porque son difíciles, pero son difíciles porque no nos atrevemos a hacerlas."

Lucio Anneo Séneca

Agradecimientos.

Quiero agradecer a mi familia todo el apoyo que he recibido, en especial a mis padres por permitirme estudiar y darme todo lo que necesito.

A Juan Antonio Morales quiero darle las gracias por su apoyo, ayuda y comprensión durante todo este tiempo, por estar a mi lado y por creer en mí.

Agradezco a mi tutor José María Cañas Plaza el entusiasmo y la dedicación que ha puesto en dirigir este proyecto, su confianza y la formación recibida.

A mis amigos, por el ánimo y el interés mostrado, así como por la ayuda prestada en la realización de experimentos.

Quiero agradecer también a Javier Martín y a José Antonio Santos su ayuda y compañerismo durante el desarrollo de nuestros proyectos.

Finalmente, agradezco a Antonio Pineda su aportación en el comienzo de este proyecto y a la gente del laboratorio de Robótica, el buen ambiente de trabajo.

Resumen.

En los últimos años, los avances en la capacidad de cómputo de los ordenadores personales y el bajo coste de las cámaras han permitido el desarrollo de nuevas aplicaciones dentro de la visión artificial. Las cámaras son sensores que proporcionan gran cantidad de información sobre el entorno donde se utilizan. Extraer e interpretar esa información permite la creación de aplicaciones que realicen tareas útiles para nuestra vida cotidiana.

La finalidad de este proyecto es crear una aplicación capaz de estimar de forma continuada la posición tridimensional de múltiples personas dentro de una habitación de gran volumen. Además, será capaz de aprender de forma automática el color característico de la vestimenta de cada persona localizada.

Para resolver la *localización 3D* se ha diseñado e implementado un *algoritmo evolutivo multimodal* que utiliza dos tipos diferentes de individuos puntuales, unos para la búsqueda de movimiento y otros para realizar el seguimiento sobre cada persona. El algoritmo extrae *información de color y de movimiento* de las imágenes de cuatro cámaras situadas en las esquinas de la habitación, asigna una salud a cada individuo basándose en la información sensorial y genera una nueva población de individuos a través de operadores genéticos.
Tras la detección de un movimiento se dispara el seguimiento de la persona que lo produjo, realizando un *aprendizaje automático del color* de esa persona con el fin de mantener su localización a pesar de que deje de moverse.

La aplicación ha sido desarrollada apoyándose en la *plataforma software Jdec*. El sistema es suficientemente *vivaz* para realizar un seguimiento tridimensional de varias personas en movimiento, ofrece una *precisión centimétrica* y utiliza *hardware convencional*.

Índice general

Índice de figuras

Capítulo 1
Introducción

La vista es el sentido que más utilizamos para captar información de nuestro entorno y consiste en la habilidad de detectar la luz y de interpretarla. Tanto los humanos como los animales poseemos un sistema visual que nos permite crear un esquema de nuestro entorno y conocer detalles de los objetos que nos rodean. Gracias a estos detalles somos capaces de reconocer dichos objetos por su color, por su forma, detectar movimiento en ellos o incluso estimar aproximadamente la distancia que nos separa.

En los últimos años, un área activa de investigación es la reproducción de estas habilidades mediante sistemas informáticos. Las cámaras son sensores de bajo coste cada vez más comunes en nuestra vida cotidiana. Esta proliferación junto con el aumento de la capacidad de cómputo de los ordenadores de sobremesa han reavivado el interés en la visión por computador.

Este proyecto fin de carrera es una aplicación directa de la localización visual, consiste en un sistema automático que permite seguir a varias personas en el interior de una habitación y estimar sus posiciones 3D gracias a la información de color y de movimiento de las imágenes que se reciben.

En este primer capítulo se describe el contexto sobre el que se apoya el sistema: la visión artificial de modo genérico y la localización y el seguimiento visual como contexto particular.

1.1. Visión artificial

La visión artificial o visión computacional es el área de la inteligencia artificial que se dedica a extraer información de las imágenes con el fin de comprender y asimilar lo que en ellas está sucediendo. La visión artificial tiene como objetivo por tanto

interpretar las imágenes analizadas para obtener información relevante sobre elementos que en ellas aparecen.

Los comienzos de la visión artificial se remontan a mediados del siglo XX, cuando aparecen los primeros programas para la detección de tanques u obstáculos. Sin embargo, el procesamiento ágil de las imágenes requiere una gran velocidad de cómputo que hasta la década de 1990 no se comenzó a lograr. A partir de entonces la visión computacional comenzó a emplearse para múltiples tareas y su desarrollo fue concentrándose en problemas de tratamiento de las imágenes como la segmentación, el reconocimiento de formas o el filtrado de bordes.

La utilización de estas técnicas posibilita el desarrollo de sistemas que persiguen objetivos concretos tales como *reconocimientos faciales o de iris* (figura 1.1 (a)), *reconocimiento de objetos* o *seguimiento 2D de objetos*. En robótica la visión puede utilizarse para la navegación del robot así como para la autolocalización o reconstrucción de lo que el robot ve.
Una reciente aplicación utilizada por la conocida marca de coches Nissan (figura 1.1 (b)) es el análisis de la cara del conductor mediante las imágenes que proporciona una pequeña cámara situada cerca del volante. Centrándose más concretamente en los ojos del conductor, permite extraer información acerca de la frecuencia de parpadeo y si los ojos están cerrados o abiertos, este análisis constituye una de las pruebas que realiza el vehículo para detectar si el conductor está en estado de embriaguez.

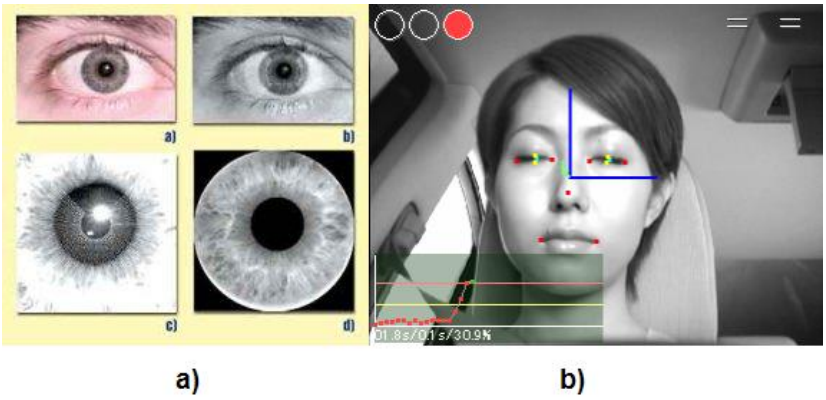

Figura 1.1: Reconocimiento con visión artificial - Iris (a) y Estado conductor (b).

Otra aplicación de gran utilidad es la identificación automática de matrículas (figura 1.2). Situando cámaras en las carreteras o en aparcamientos el sistema permite localizar

e identificar la matrícula en el vídeo y mediante un módulo de reconocimiento de caracteres se extrae el número de matrícula. Este sistema combinado con una base de datos que almacene información respecto a los conductores y vehículos es suficiente para gestionar y controlar el acceso a aparcamientos o incluso al casco urbano de una ciudad. Dos buenos ejemplos de ello son los aparcamientos del aeropuerto de Barajas y la zona centro de Londres.

Figura 1.2: Identificación de matrículas.

En la actualidad, los avances en geometría proyectiva, pares estéreo y autocalibración han abierto la puerta a la extracción de *información tridimensional* de las imágenes. El uso de estas nuevas técnicas permite alcanzar nuevas cotas en el área de la visión artificial, pues la información tridimensional permite conocer mejor y con una mayor precisión el entorno. Por ejemplo es posible hoy día realizar *reconocimiento 3D* de rostros u objetos (figura 1.3), que pueden ser utilizados en distintos ámbitos como la representación de piezas en entornos industriales, el reconocimiento de personas mediante el estudio biométrico en áreas de seguridad, diseño infográfico para la industria del cine o de videojuegos, etc.

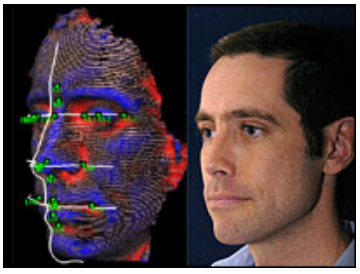

Figura 1.3: Modelo rostro tridimensional.

Otro ejemplo es el sistema *EyeVision*[1], que permite modelar en tiempo real una escena de manera fotorrealista con el fin de obtener nuevas imágenes virtualizadas de la misma. Este sistema ha sido empleado en la Superbowl para la repetición de jugadas desde cualquier ángulo. A través de la colocación estratégica de un determinado número de cámaras alrededor de un estadio es posible obtener la realidad virtualizada de lo sucedido segundos antes. La escena tridimensional se obtiene de la composición de las imágenes 2D que consigue cada una de las cámaras.

Figura 1.4: Sistema de repetición tridimensional Eye-Vision.

1.2. Localización y seguimiento 3D

La localización 3D consiste en determinar la *posición de un objeto* o elemento dentro de un mapa o *entorno tridimensional*. Esta posición habitualmente vendrá especificada como un punto tridimensional de coordenadas (x,y,z), cuya situación estará referida a un origen o centro de referencia previamente establecido.
El seguimiento es la estimación de la posición de un objeto de forma continuada. La posición puede cambiar a lo largo del tiempo, por lo que el proceso de seguimiento debe actualizar dicha posición continuamente, obteniendo la trayectoria del objeto.

Para lograr la localización y el seguimiento 3D es necesario el uso de cámaras u otros sensores que informen de la posición del objeto. En el caso de las cámaras, se necesitan al menos dos imágenes de posiciones diferentes apuntando al mismo objeto para poder conseguir la localización 3D. Mediante dos o más imágenes es posible aplicar una técnica de localización que determine la posición.

Algunas técnicas clásicas de localización son las siguientes.

- *Localización por mínimos cuadrados*: estiman la posición del objeto basándose

[1] http://www.ri.cmu.edu/events/sb35/tksuperbowl.html

en el cálculo del error mínimo de las observaciones realizadas.

- *Localización con filtros de kalman*: tratan de estimar recursivamente la posición de mínima varianza fusionando información parcial e indirecta sobre localización [Maybeck, 1979][Bishop y Welch, 2005]. Su principal limitación es que es una técnica unimodal y exclusivamente gaussiana, por lo que no es capaz de manejar múltiples hipótesis.

- *Localización probabilística*: este tipo de localización consiste en determinar la probabilidad de que el objetivo se encuentre en una determinada posición a través de los datos obtenidos de los sensores [D. Margaritis, 1998]. A cada posible posición se le asocia una probabilidad reflejando la verosimilitud de ser la posición actual del objetivo. Esta probabilidad se va actualizando con la incorporación de nuevas lecturas. La eficiencia de estas técnicas generalmente depende del tamaño del entorno donde se efectúa la localización, siendo más difícil de manejar en espacios muy grandes.

- *Localización con métodos metaheurísticos*: estos filtros contemplan el problema de la localización como una búsqueda de la posición entre un número determinado de soluciones posibles. Los métodos metaheurísticos de localización realizan una búsqueda de la posición de manera algorítmica, no aseguran una solución óptima al problema, sino más bien una aproximación que en ocasiones puede ser la mejor. Constituye un buen ejemplo el filtro de moscas de *Louchet* diseñado inicialmente para el problema de la reconstrucción 3D [Louchet, 2000][Louchet, 2001][Louchet *et al.*, 2002]. El filtro cuenta con una población de N moscas, cada una representa un estado y se le asigna un peso o salud que se actualiza en cada iteración del proceso.

Una famosa aplicación utilizada en distintos ámbitos tales como medicina, deporte o en la industria cinematográfica es el sistema *Vicon*[2] (figura 1.5), que se centra en la captura de movimiento. Mediante el uso de unos marcadores especiales situados sobre el cuerpo de una persona y varias cámaras filmando desde diferentes puntos de vista, un ordenador analiza todas las trayectorias y extrae un esqueleto 3D animado. Los movimientos recogidos servirán para aplicarlos posteriormente a un personaje de animación o para su análisis en el caso de la medicina y el deporte.

[2]http://www.vicon.com/

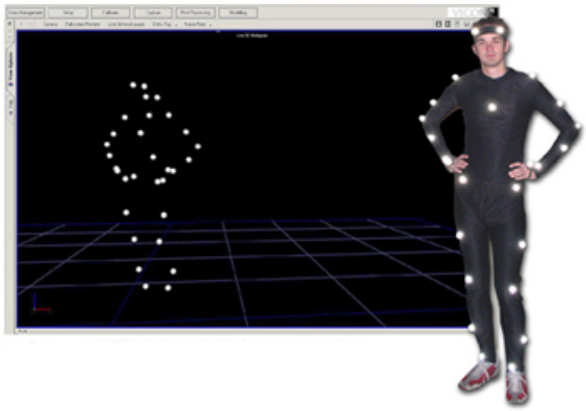

Figura 1.5: Sistema de captura de movimiento de Vicon Peak.

1.3. Seguimiento 3D visual con algoritmo evolutivo

La localización y el seguimiento 3D pueden aplicarse en múltiples escenarios en los que, según la situación, conocer la posición de un objeto o una persona puede servir y ayudar enormemente en tareas muy complejas.

El antecedente de este proyecto es el proyecto fin de carrera *Aplicación de seguridad basada en visión* [Cabello, 2006], que consistió en realizar una aplicación de seguridad que permite localizar dentro de una habitación a un sujeto del cuál se conoce su color y establecer una zona restringida. Si el sujeto se adentra en ella, el sistema avisa con una alarma sonora y visual (figura 1.6).

Figura 1.6: Aplicación de seguridad - Alarma no activa (a) y Alarma activa (b).

Para el desarrollo de esa aplicación se emplearon dos técnicas distintas: un algoritmo

evolutivo monomodal y una técnica probabilística denominada filtro de partículas. La localización y el seguimiento 3D en dicho proyecto fueron aplicados a la *vigilancia y protección de objetos* y al *cuidado de ancianos* utilizando la estimación en la coordenada Z para avisar de la caída de una persona mayor.

Esa aplicación de seguridad constituye el punto de partida de este proyecto, siendo los objetivos principales lograr el seguimiento de *múltiples* sujetos y *aprender automáticamente el color* de cada uno de ellos.

Un *algoritmo evolutivo multimodal* es la técnica empleada para localización y el seguimiento 3D de múltiples personas en este proyecto. Otros trabajos recientes con algoritmos evolutivos multimodales son los proyectos fin de carrera *Localización y construcción de mapas en un robot de interiores* [Ángel Cortés Maya, 2007] y *Reconstrucción 3D visual con algoritmos evolutivos* [Martínez, 2007].

En la línea del seguimiento multiobjeto cabe citar la tesis doctoral *Aplicación de métodos secuenciales de Monte Carlo al seguimiento visual 3D de múltiples objetos* [González, 2007], que utiliza un filtro de partículas similar al utilizado en [Cabello, 2006].

Capítulo 2
Objetivos

Tras haber presentado el contexto general y particular sobre el que se asienta este proyecto fin de carrera, estableceremos en este capítulo los objetivos concretos que se han propuesto al igual que los requisitos que han condicionado el desarrollo del mismo.

2.1. Descripción del problema

El objetivo del proyecto es la localización y seguimiento 3D de múltiples personas mediante técnicas de visión. Con este propósito, se diseñará y se implementará una aplicación que resolverá este problema mediante un algoritmo evolutivo multimodal.

El objetivo principal se ha dividido en varios subobjetivos más específicos.

1. *Diseño e implementación de un algoritmo evolutivo multimodal* para la interpretación de la información obtenida de las cámaras. Este subobjetivo pretende construir una alternativa más potente al algoritmo utilizado en [Cabello, 2006] que adaptó al problema de la localización el algoritmo genético de Louchet [Louchet, 2000]. El algoritmo que se diseñará será multimodal, similar al implementado en [Martínez, 2007] pero enfocado a resolver el problema planteado.

2. *Diseño e implementación del autoaprendizaje de color de las personas.* La aplicación no debe disponer de ningún tipo de información de color acerca de la vestimenta de las personas que se localizarán, sino que aprenderá automáticamente el color característico de cada una de ellas.

3. *Exploración de distintas técnicas para la detección de movimiento en las imágenes.* Para ello se implementarán y contrastarán todas las técnicas con la finalidad de obtener el mejor resultado.

4. *Integración y experimentos.* Con la intención de probar y ajustar el funcionamiento de la aplicación con los todos los subsistemas integrados, se realizarán numerosas pruebas para asegurar y cumplir lo más fielmente posible los objetivos del proyecto.

2.2. Requisitos

Los requisitos propuestos para la aplicación que presenta este proyecto fin de carrera están resumidos en los siguientes puntos:

1. Implementación del sistema apoyándose en la *plataforma software Jdec*[1] sobre el sistema operativo Linux.
2. Utilización del *lenguaje C* para programar la aplicación, que está impuesto por la plataforma.
3. Funcionamiento de la aplicación usando *hardware convencional*.
4. Validación del sistema en una *habitación de gran volumen*, en concreto el Laboratorio de Robótica de la URJC.
5. El software a desarrollar será *vivaz*, será capaz de seguir personas andando dentro de la habitación.
6. Seguimiento de varias personas u objetos con una *precisión centimétrica*.

2.3. Metodología y plan de trabajo

El plan de trabajo utilizado en la realización de este proyecto ha consistido en el *modelo de desarrollo en espiral basado en prototipos* (figura 2.1). Este modelo de desarrollo se basa en la realización de varias subtareas sencillas que conjuntadamente compondrán el comportamiento final del sistema. Usando este modelo se aporta cierta flexibilidad en cuanto a posibles cambios de requisitos.

Este modelo de desarrollo se caracteriza por la realización de las subtareas en un número determinado de ciclos. En cada uno de estos ciclos existen cuatro etapas: *Análisis de requisitos, Diseño e implementación, Pruebas y Planificación del próximo ciclo de desarrollo.*

[1] http://gsyc.escet.urjc.es/ jmplaza/software.html

Durante todo el desarrollo del proyecto se han mantenido reuniones semanales con el tutor para establecer y afinar los puntos que se han llevado a cabo en cada etapa, así como para comentar los resultados de etapas anteriores.

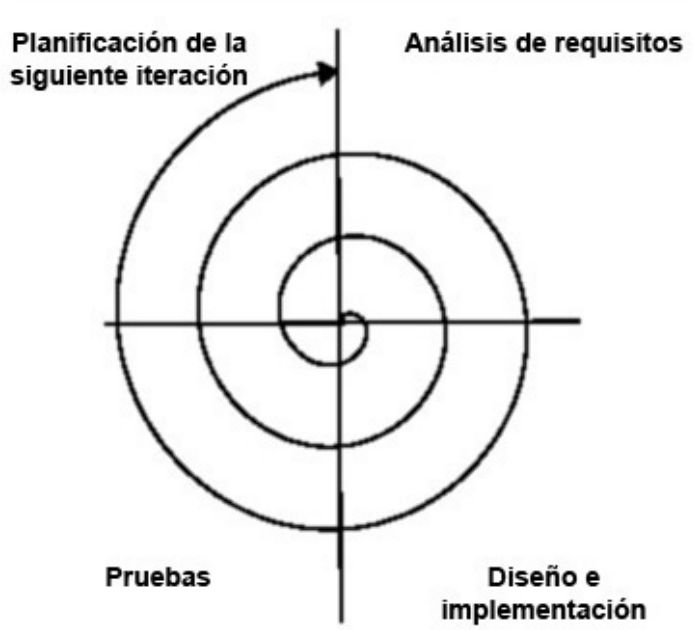

Figura 2.1: Modelo en espiral

Para el desarrollo de este proyecto se han completado los siguientes siete ciclos:

1. *Familiarización con la plataforma software Jdec* [Plaza, 2004] y estudio a fondo de la estructura de la misma. La primera iteración de la espiral consistió en la creación de varios esquemas para comprender el funcionamiento general de *Jdec*.

2. *Estudio de técnicas de visión artificial y otros conocimientos relativos* tales como filtros de color y movimiento, conocimientos de espacios de color, soportes de vídeo, calibración y funcionamiento de controladores de camáras USB y Firewire. Diseño y creación de nuevos esquemas de ejemplo en los que se aplican dichas técnicas.

3. *Implementación de la aplicación de seguridad basada en visión [Cabello, 2006] sobre una nueva versión de Jdec*. Para conocer de forma cercana el problema se reimplementó la aplicación en la cuál se basa este proyecto sobre la versión *Jdec-4.2*, lo que suponía cambios significativos respecto de la 3.4.

4. *Diseño e implementación del algoritmo evolutivo multimodal*. En este ciclo se realizó un estudio e implementación del algoritmo genético adaptado a resolver el problema de la localización 3D de múltiples objetos.

5. *Diseño e implementación de la autodefinición del color de los sujetos.* Mediante el estudio de distintas técnicas se desarrolló un aprendizaje automático del color característico de cada objeto o persona localizada.

6. *Exploración y contraste de distintas técnicas para la detección de movimiento en las imágenes.* Se implementaron distintas técnicas y finalmente se determinó cuál de ellas era más adecuada para este proyecto.

7. *Experimentos.* La última etapa de la espiral consistió en experimentos realizados con las técnicas implementadas integradas, ajustando parámetros para optimizar su funcionamiento.

Capítulo 3
Entorno de desarrollo

En este capítulo se describe la infraestructura software y hardware sobre la que se apoya este proyecto fin de carrera, al igual que las bibliotecas que se han utilizado para su funcionamiento.

3.1. Infraestructura hardware: Cámaras Firewire

Las cámaras son un elemento hardware indispensable en el funcionamiento de este proyecto. Son los sensores perceptivos utilizados por la aplicación de seguimiento para poder estimar una posición 3D en la habitación. Las cuatro cámaras utilizadas son del modelo Apple iSight[1] (figura 3.1). Este modelo de cámara se caracteriza por poder capturar imágenes en color de hasta 640x480 píxeles con un ritmo de actualización de 15 fotogramas por segundo (fps), o bien de 320x240 a 30 fps tal y como se utiliza en este proyecto. Para ello se conecta la cámara a un ordenador através de una conexión de tipo IEEE1394 a 400 Mbps. El ritmo de actualización tan vivaz de los fotogramas se debe a la *utilización de DMA* para acceder a las imágenes capturadas. La cámara utiliza además un sistema de enfoque y apertura del iris automático.

Figura 3.1: Cámara web Apple iSight

Para poder conectar las cámaras firewire y ejecutar la aplicación es necesario

[1] http://www.apple.com/isight

disponer de *al menos un ordenador* de gama media, al que estarán conectadas las cuatro cámaras y que además ejecutará la aplicación de seguimiento. Otra manera de conectar las cámaras es mediante una red local, en la que se necesitarían varios ordenadores funcionando como *servidores de imágenes*.

3.2. Infraestructura software: La plataforma *Jdec*

La aplicación de seguimiento 3D se apoya en la plataforma robótica *Jdec*, que ha sido desarrollada en la Universidad Rey Juan Carlos y es fruto de una tesis doctoral sobre generación de comportamientos autónomos en robots móviles [Plaza, 2003]. Esta plataforma permite desarrollar aplicaciones robóticas para el robot *Pioneer*[2] que tomen decisiones inteligentes de manera autónoma y aplicaciones domóticas en las que intervienen sensores y/o actuadores.

La plataforma plantea las aplicaciones como un conjunto de *esquemas* que se ejecutan simultáneamente en hilos distintos y que realizan tareas sencillas y concretas. Esta ejecución concurrente de los esquemas da lugar a un comportamiento.
Para aplicaciones reactivas sencillas, con un único esquema que tenga muchas iteraciones por segundo resulta suficiente.
Para aplicaciones algo más complejas *Jdec* propone dividir la funcionalidad en esquemas perceptivos y esquemas de actuación concurrentes, todos al mismo nivel. Los esquemas perceptivos tratan los datos sensoriales y los esquemas de actuación recogen la información sensorial o de otros esquemas perceptivos y toman decisiones de actuación.
Para aplicaciones aún más complicadas *Jdec* propone organizar la colección de esquemas en una jerarquía de varios niveles, donde unos activan a otros. Esta división software tiene su sustrato teórico en la arquitectura cognitiva JDE [Plaza, 2003] para generar comportamiento autónomo en robots.

Este proyecto fin de carrera usa un único esquema perceptivo, por lo que sólo hay un nivel y no utiliza jerarquía.

[2] http://www.activrobots.com/ROBOTS/p2dx.html

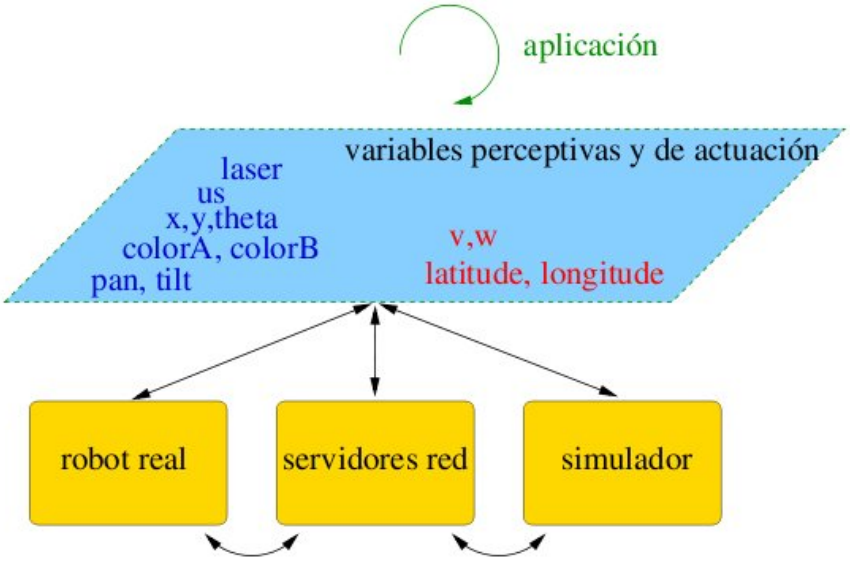

Figura 3.2: *Jdec* básico

La plataforma se ha implementado en una arquitectura software *Jdec* que ofrece una interfaz de variables de percepción y de actuación (figura 3.2). La aplicación robótica o domótica obtiene las últimas observaciones sensoriales leyendo las variables perceptivas, que la plataforma mantiene permanentemente actualizadas. Por otra parte ordena comandos a los motores escribiendo en las variables de actuación, que la plataforma se encarga de materializar y hacer llegar hasta los actuadores correspondientes.

Como ilustra la figura 3.2, *Jdec* permite tres posibles configuraciones típicas:

1. Los sensores y actuadores del robot real pueden estar físicamente conectados al ordenador en el que se ejecuta *Jdec*. Por ejemplo, con la base motora del Pioneer conectada al puerto serie del ordenador portátil, y el par de cámaras a su bus firewire. La plataforma incorpora los *drivers* oportunos para resolver el acceso local a esos dispositivos.

2. Los sensores y actuadores pueden estar simulados y *Jdec* se conecta al simulador.

3. *Jdec* se conecta a los *servidores de red* que a su vez están finalmente enlazados con los dispositivos hardware locales o con un simulador. La ventaja de usar estos servidores es que los dispositivos y la aplicación robótica no tienen que estar necesariamente en la misma máquina.

Si bien la arquitectura software está principalmente orientada a la programación de comportamientos en robot, ha sido adaptada al caso concreto de este proyecto para *aprovechar de ella todos los elementos dedicados a la visión artificial*. Principalmente se utiliza el acceso a las imágenes de las cámaras, que mediante la interfaz proporcionada se actualizan en las variables perceptivas *colorA, colorB, colorC y colorD*.

3.2.1. Drivers Networkclient y Networkserver

La plataforma *Jdec* permite conectar las cámaras mediante una red local, para ello dispone de dos drivers que posibilitan el flujo de datos entre los distintos ordenadores.

El driver *networkserver* permite a *Jdec* servir mediante una conexión *TCP/IP* todos los dispositivos que pueda tener conectados. Entre estos se incluye cualquier cámara o imagen estática configurada, los dispositivos del robot o un cuello mecánico. El acceso a los dispositivos se puede realizar en una red local o a través de internet. Mediante este driver, *Jdec* se puede utilizar como un servidor de imágenes o de dispositivos de un robot.

El driver *networkclient* permite a *Jdec* conectarse a otro *Jdec* que esté usando el driver *networkserver* para servir algún dispositivo.

3.2.2. Driver Mplayer

Otra posibilidad que ofrece la plataforma es obtener las imágenes desde ficheros de vídeo. El driver *Mplayer* puede cargar simultáneamente hasta cuatro vídeos y ofrecerlos de modo que la aplicación trabaje sobre ellos de la misma forma que si trabajara con imágenes en directo. El driver está apoyado en la funcionalidad del reproductor de películas para Linux *mplayer* y el codificador *mencoder*.

3.3. Bibliotecas auxiliares

En conjunción con la plataforma software *Jdec* se han utilizado también tres bibliotecas adicionales en este proyecto fin de carrera. La primera de ellas es la *biblioteca de geometría proyectiva Progeo* utilizada para operar con un espacio tridimensional. La segunda es la *biblioteca de interfaces gráficas XForms*. Y la última es la *biblioteca de primitivas de rendimiento integrado de Intel* (IPP).

3.3.1. Progeo

La biblioteca de geometría proyectiva Progeo es utilizada en la aplicación para relacionar el mundo de las imágenes (2D) con el mundo real (3D). Esta relación se consigue gracias a dos funciones principales:

- *Proyectar*. Esta función permite realizar la proyección óptica de un punto 3D del espacio en el plano imagen de una de las cámaras, obteniendo así un punto 2D perteneciente a ese plano. Con ese punto 2D es posible obtener el píxel de la imagen correspondiente.
- Función *Retroproyectar*. Esta función permite obtener la recta de proyección que une el centro óptico de una cámara con el punto 3D que representa un punto 2D de su plano imagen. Ese punto 2D se corresponde con un píxel en la imagen de esa cámara y mediante esta función se puede llegar a obtener el punto 3D del que es proyección en el plano. A través de la intersección de dos rectas de proyección de un mismo punto 3D es posible obtener dicho punto resolviendo el sistema de ecuaciones.

Progeo se basa en el *modelo de cámara Pinhole* para realizar estas transformaciones, que se puede observar en la figura 3.3.

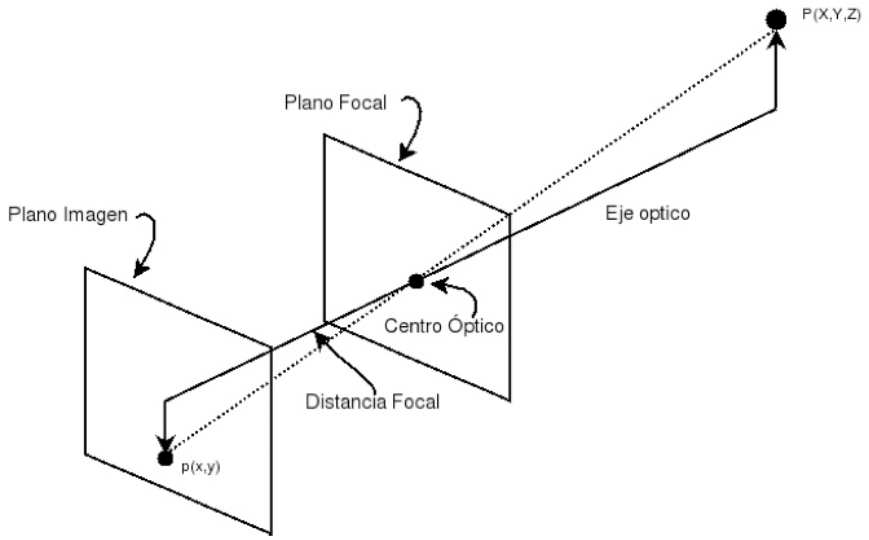

Figura 3.3: Modelo de cámara Pinhole

En este modelo se asume que cualquier punto $P(x, y, z)$ se proyecta en el plano de imagen a través de otro único punto llamado *Centro óptico*. La recta que une el punto P y el centro óptico se denomina *Línea de proyección* e intersecta al plano imagen justo en el píxel $p(x, y)$, que es la proyección de $P(X, Y, Z)$. El centro óptico está situado a la *Distancia focal* del plano imagen. Este modelo lo completan el *Eje óptico*, que es una línea perpendicular al plano imagen y que atraviesa al centro óptico, y también el *Plano focal*, que es el plano perpendicular al eje óptico cuyos puntos no se proyectan en el plano imagen, incluyendo al centro óptico.

Progeo proporciona además los tipos de datos *Punto2D* y *Punto3D* para la representación de puntos en el plano y en el espacio. También proporciona los tipos *CamaraPinhole* y *CamaraStereoPinhole*. Ambos tipos se utilizan para simular el uso de cámaras y pares estéreo en el entorno tridimensional.

Para la generación de imágenes virtuales del mundo 3D en el que trabaja la aplicación es necesario que las cámaras se encuentre calibradas. El proceso de calibración consiste en obtener los *parámetros intrínsecos* y *parámetros extrínsecos* de las mismas. En el caso de los parámetros intrínsecos es necesario conocer la *distancia focal* en cada eje y el tamaño del píxel central del plano imagen. Para los parámetros extrínsecos se necesita la posición 3D de la cámara y la orientación.

3.3.2. XForms

La biblioteca de interfaces gráficas XForms se apoya directamente en el servidor de imágenes X Window. La biblioteca proporciona una herramienta con la que poder crear la interfaz de botones, barras de desplazamiento, menús o cuadros de imágenes llamada *fdesign*.

Con XForms es posible crear ventanas directamente sobre el sistema X Window de todo sistema operativo Linux. No se apoya en ninguna otra biblioteca o motores intermedios. Es por esto que una interfaz en Xforms es prácticamente compatible con cualquier sistema actual basado en unix del mercado, a diferencia de otras librerías más modernas y de diseño más atractivo. Se tiene la garantía de que la interfaz gráfica puede funcionar en todo sistema que incorpore este servidor gráfico (linux, maCOs, freebsd, etc).

3.3.3. Intel Integrated Performance Primitives (IPP)

Intel IPP es una librería multiplatforma que permite escribir aplicaciones altamente optimizadas para maximizar el rendimiento en procesadores Intel. Intel IPP incluye funciones de procesado de señal e imagen, cifrado, manipulación de cadenas de texto y vectores, matemática de matrices, así como sofisticadas primitivas para la construcción de codecs de audio y vídeo.
La versión utilizada de esta biblioteca es Intel IPP 5.1, descargable de su página web[3].

Las funciones usadas son las siguientes.

- *Conversión*. Para la visualización en la interfaz gráfica de las imágenes captadas es necesario convertirlas de tres bytes por píxel a cuatro bytes por píxel.
- *Espejo*. Para la instalación de las cámaras en el techo fue necesario situarlas boca abajo, por lo que las imágenes que se obtienen de ellas están invertidas. Esto supone que la aplicación debe ajustarlas para su correcta visualización mediante esta función.

[3]http://www.intel.com/cd/software/products/asmo-na/eng/perflib/ipp/302910.htm

Capítulo 4
Descripción informática

Una vez presentados los requisitos y las herramientas utilizadas en este proyecto, en este capítulo se describe cómo se ha estructurado e implementado la aplicación. Primero se muestran los fundamentos teóricos de los algoritmos evolutivos, se presenta seguidamente el diseño global del programa así como el algoritmo implementado y por último se detalla la interfaz gráfica de la aplicación.

4.1. Fundamentos teóricos de los algoritmos evolutivos

Los llamados algoritmos genéticos o evolutivos son algoritmos de búsqueda que tratan de hallar una solución óptima al problema planteado. Estos algoritmos combinan las propiedades de las hipótesis en el espacio de soluciones para obtener nuevas generaciones más robustas. Son algoritmos iterativos que guardan cierta similitud con la evolución natural y se utilizan a menudo en tareas de optimización, diseño de controladores borrosos o sistemas de aprendizaje automático.

Estos algoritmos cuentan con un conjunto o población de N individuos donde cada individuo n_i tiene asociado un determinado peso o salud h_i, que indica cómo de buena es esa solución y se actualiza en cada iteración del proceso. Dependiendo del propósito del algoritmo, un determinado individuo puede ser en si mismo una posible solución (enfoque *Pittsburgh*) o bien sólo una contribuyente a la misma (enfoque *Michigan*).

Los algoritmos genéticos se dividen principalmente en dos fases:

1. *Generación de la próxima población.* Se aplican operadores genéticos para obtener la nueva población de individuos.

2. *Computación de la salud.* Se calcula la salud para cada uno de los individuos en función de las observaciones realizadas.

La nueva población se genera gracias a los *operadores genéticos*, utilizando en el proceso a los individuos que reúnen las mejores propiedades de la población anterior. Estas propiedades dependerán del propósito del algoritmo, y los operadores genéticos pueden ser diseñados expresamente para captar estos rasgos de manera más eficiente. Existen varios operadores genéticos clásicos que se suelen utilizar siendo adaptados al contexto específico de cada problema. Algunos de estos operadores son: *Mutación aleatoria*, *Ruido térmico*, *Cruce* o *Repulsión*. La Mutación aleatoria permite generar individuos en cualquier posición del espacio de soluciones. El Ruido térmico genera individuos alrededor de otro individuo determinado. El Cruce permite combinar las características de dos individuos para generar una nueva. La Repulsión evita que dos individuos sean generados en la misma posición del espacio de soluciones.

El cálculo de salud consiste en evaluar a cada uno de los individuos a partir de unos criterios muy dependientes del problema concreto. Cada uno de estos criterios se corresponde con características que determinan la calidad del individuo. Cuantos más criterios satisface un individuo mayor será su valor de salud h_i.

4.2. Diseño general

El objetivo del algoritmo evolutivo implementado es la localización y el seguimiento 3D de múltiples personas a través de varias cámaras en un espacio controlado.
La aplicación se compone de un único esquema sobre la plataforma software *Jdec*. En cada iteración de este esquema se obtiene información sensorial de cada una de las cámaras en forma de imágenes 2D y se realiza una nueva estimación del número de personas y sus posiciones 3D en la sala, representando de forma gráfica y numérica la solución a la localización.

Sobre las imágenes 2D de entrada se extrae información de movimiento y de color para la detección y el seguimiento de las personas. Gracias a la utilización de más de una cámara es posible realizar la estimación 3D de sus posiciones, considerando como espacio de soluciones las posiciones 3D dentro de la habitación en la que se han situado las cámaras.

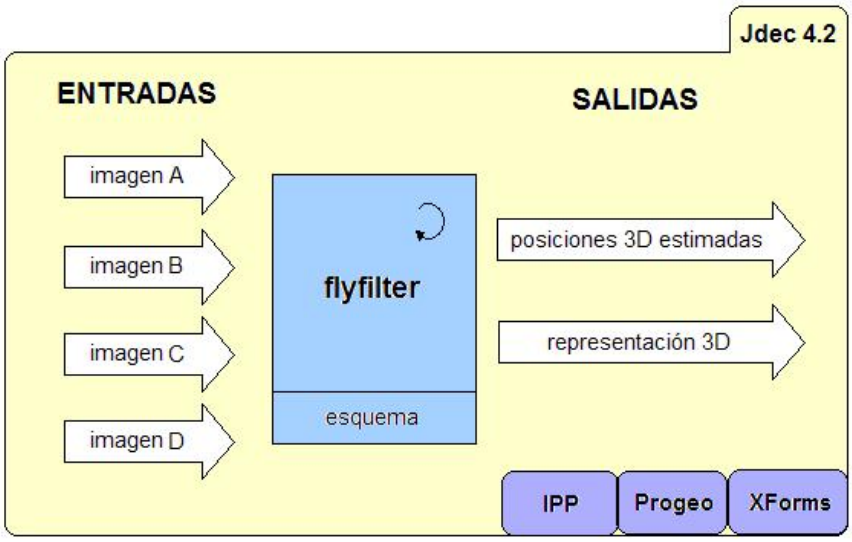

Figura 4.1: Representación del diseño general de la aplicación.

En esta adaptación del algoritmo genético al problema de la localización 3D la primitiva de información utilizada es un conjunto de individuos puntuales y se distinguen dos grupos diferentes: exploradores y explotadores. Sobre cada tipo se utilizan operadores genéticos diferentes y poseen características distintas:

- *Población de exploradores.* Está compuesta por M individuos destinados a explorar todo el espacio de la habitación controlada. Permiten una búsqueda de forma gruesa con el fin de encontrar personas en movimiento. La salud de esta población se asigna únicamente tomando en cuenta la información de movimiento extraída de las imágenes. Un individuo será prometedor si su salud supera un umbral dado, es decir, si ha detectado un movimiento considerable.

- *Población de explotadores.* Esta población se compone de un número de grupos de individuos o razas indeterminado, inicialmente será cero. Cada raza cuenta con N individuos cuya finalidad es explotar solamente una zona concreta del espacio 3D en la que haya aparecido un individuo prometedor de la población de exploradores. Su salud se define a través de la información de color y de movimiento de las imágenes. Las razas explotadoras se crean durante la ejecución del algoritmo, nacen a partir de individuos prometedores de la población exploradora y se extinguen si su salud es baja. Generalmente surgirá una sola raza por persona y a través de ésta se realizará su seguimiento.

En este diseño, cada uno de los individuos explotadores de una raza es una solución en si misma para la localización de la persona que esté siguiendo esa raza (enfoque *Pittsburgh*), por lo que se corresponde con un *punto 3D* de coordenadas (x,y,z). La salud h_i de cada individuo indica cómo de buena es esa solución teniendo en cuenta las observaciones en las imágenes de las cuatro cámaras del sistema.

En la figura 4.2 se muestra el diagrama de flujo que representa el funcionamiento del algoritmo.

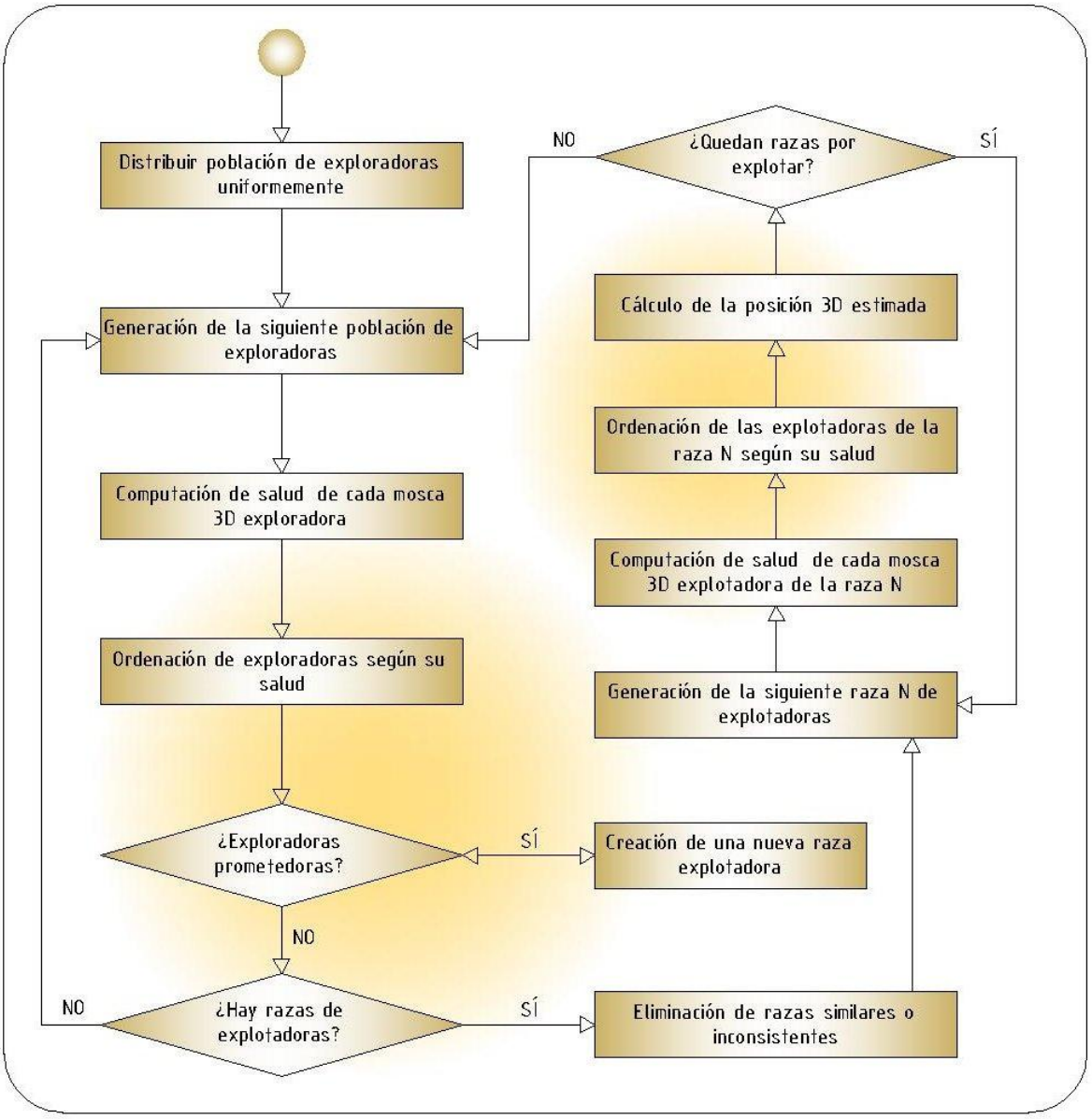

Figura 4.2: Diagrama de flujo del algoritmo.

Por tanto, el seguimiento de una persona dentro de la habitación supone dos etapas: primero la búsqueda y detección de movimiento producido por la persona y segundo su seguimiento.

En la etapa de búsqueda se distribuye la población de individuos exploradores uniformemente por todo el espacio. El objetivo de esto es realizar una búsqueda de movimiento en la sala. Mediante ciertos operadores genéticos que se detallarán más adelante y la información de movimiento extraída de las imágenes, la población de exploradores converge en las zonas 3D de interés, donde se ha detectado movimiento. A cada individuo se le asigna un peso o salud h_i, aquel cuya salud supere un umbral determinado se considera un individuo prometedor y éste provoca la creación de una nueva población de individuos en su posición 3D, esto es, una nueva raza. El proceso de búsqueda está siempre activo con la finalidad de explorar en todo momento el movimiento en la habitación, lo que permite la aparición de nuevos seguimientos.

Tras el nacimiento de una raza comienza la segunda etapa, la de seguimiento, y la zona del espacio que ocupa la persona es analizada extrayendo información de movimiento y de color de las imágenes a través de los individuos de la nueva población. Así, una raza se desplazará siguiendo la trayectoria de la persona localizada. Además, para cada persona localizada se realiza un aprendizaje de color que permite continuar el seguimiento aunque la persona se detenga, caracterizándose cada raza por su color correspondiente.
La raza se mantendrá viva mientras se detecte compatibilidad con las observaciones sensoriales, es decir, si la persona sobre la que se realiza el seguimiento abandona la sala, la raza desaparece. Si no se detectan personas en la habitación, el algoritmo únicamente opera sobre la población de exploradores.
A lo largo de la ejecución de la aplicación las razas nacerán y desaparecerán dinámicamente según el número de personas que se encuentren en la habitación y la información extraída de las imágenes.

El esquema que materializa la aplicación se denomina *flyfilter*. El carácter iterativo del esquema en *Jdec* encaja a la perfección con la propia naturaleza del algoritmo evolutivo. La *primera población exploradora es totalmente aleatoria* e iterativamente irá evolucionando.

4.3. Generación de la siguiente población

En esta etapa del algoritmo se trata de *generar las poblaciones de individuos* gracias a las observaciones realizadas en las imágenes de las cámaras y a la población de individuos en la iteración anterior.

Este procedimiento se efectúa utilizando los *operadores genéticos*. Estos operadores se encargan de captar las mejores propiedades de los individuos de la población anterior de forma que se consiga una nueva que permita una localización más exacta de las personas seguidas.

4.3.1. Generación de la población de exploradores

La población de exploradores se sitúa por todo el espacio de la habitación y su propósito es ubicar individuos en posiciones 3D compatibles con la información de movimiento extraída de las imágenes. Mediante la utilización de los operadores genéticos de abducción y mutación aleatoria se consigue dicho propósito.

A continuación se detalla el funcionamiento de estos operadores:

- *Operador de mutación aleatoria*. Este operador selecciona al azar un individuo con baja salud de entre todos los de la población y lo posiciona aleatoriamente en otro lugar de la habitación. El algoritmo ordena previamente la población de exploradores por salud y para aquellos con menor salud de manera aleatoria determina una nueva posición y los ubica en dicho lugar. Este operador permite *explorar uniformemente el espacio* de la habitación. Sus probabilidades de éxito son bajas pero permite evitar los máximos locales en la búsqueda de la solución.

- *Operador genético de abducción*. Es un operador creado desde la particularidad del problema de la localización 3D desde visión. La abducción permite suponer que si en una imagen de una cámara visualizamos un objeto que se encuentra en movimiento, el objeto en sí estará en algún punto de la línea de proyección que une el centro óptico con el punto donde intersecta al plano imagen. De este modo, es razonable pensar que generando nuevos individuos a lo largo de esa línea de proyección, algunos puedan estar cerca del objeto. Así, siempre que se detecte en cada iteración algún objeto en movimiento, podrá visualizarse un haz de individuos distribuidos a lo largo de las líneas de proyección de cada cámara tal y como puede verse en la figura 4.3 para un objeto situado en el centro de la habitación. Esta búsqueda permite acelerar la convergencia de la población en

las zonas interesantes, por lo que es más eficiente que la simple búsqueda al azar. Este operador precisa que se realice un filtrado completo por movimiento de las imágenes que permita obtener una lista de píxeles relevantes de cada imagen.

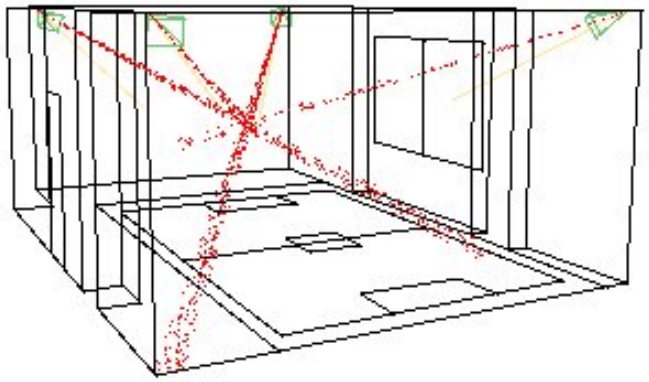

Figura 4.3: Resultado del operador abducción.

El número de individuos que genera cada operador se fija mediante porcentajes, lo que permite dar más o menos importancia a cada operador. Este ajuste determina en gran medida el comportamiento del algoritmo, por lo que es importante mantener un buen equilibrio entre ambos operadores para que la búsqueda de movimiento sea eficaz.

4.3.2. Generación de la población de explotadores

La evolución de esta población se consigue a través de la utilización de los operadores genéticos de elitismo y ruido térmico, que se aplican por cada raza. En este caso la población anterior tiene más peso en la nueva generación que en el caso de los exploradores.

- *Operador genético de elitismo.* Este operador necesita que los individuos estén ordenados por salud y elige los explotadores con mayor salud de la población anterior permitiendo que permanezcan en la nueva población sin que sufran ningún cambio. La idea del operador consiste conservar una estabilidad en la nueva población no olvidando aquellas posiciones que ya eran compatibles en el estado anterior.

- *Operador genético de ruido térmico.* Este operador, al igual que la mutación aleatoria, se encarga de operar con los individuos que poseen menor salud. Sin

embargo en este caso las nuevas posiciones aleatorias son cercanas a las que ocupaban en la población anterior. Permite *explotar una zona* distribuyéndose como ruido térmico en torno a la misma.

Del mismo modo que en la generación de los exploradores, la medida en que se utiliza un operador u otro se determina mediante porcentajes, que fijan el número de individuos que debe generar cada operador.

4.4. Computación de la salud

Este segundo paso del algoritmo consiste en la asignación de salud o peso h_i a cada uno de los individuos generados. Al asignar salud a los individuos se podrán distinguir aquellos que se encuentren más cerca de la persona a seguir de los que estén lejos. Poseerán mayor salud los individuos cuyo grado de compatibilidad con las características buscadas repecto al movimiento y al color sea elevado. La salud, por tanto, modula la evolución de la población entre los individuos propuestos por los operadores en cada generación.

Para el cálculo de la salud se tiene en cuenta la información de color y de movimiento extraída de las imágenes capturadas. En el caso de la población de individuos exploradores, su salud se define únicamente en base a la información de movimiento puesto que el sistema utiliza el movimiento como *disparador* del seguimiento. Una vez iniciado el seguimiento, a través de la población de individuos explotadores surgida se realiza un aprendizaje del color de la vestimenta de la persona. Así, la ecuación de salud de los exploradores estará asociada únicamente a la salud calculada en base al movimiento y en el caso de la población de explotadores, estará asociada tanto a la salud por color como por movimiento.

La ecuación de salud total para un invididuo será:

- Población de exploradores: $h_i = mov_i$
- Población de explotadores: $h_i = (color_i + mov_i)/2$

Donde $color_i$ y mov_i son la salud asociada al color y la salud asociada al movimiento, respectivamente.

4.4.1. Salud asociada al color

Como se ha explicado anteriormente, la aplicación inicialmente no cuenta con ningún tipo de información de color acerca de las personas. A través de un aprendizaje automático el sistema genera un filtro de color para cada raza surgida y gracias a éste se valora la salud asociada al color de una raza (el proceso de este aprendizaje se detallará en la sección 4.7.2).
Por tanto, esta salud permite valorar si el color aprendido para cada raza es compatible con las observaciones sensoriales.

Primeramente se proyecta el punto 3D que representa al explotador en un píxel del plano imagen de la cámara. En este paso se utiliza la *función proyectar* de la biblioteca de geometría proyectiva Progeo (descrita en el apartado 3.3.1). Tras conocer el píxel de la imagen sobre el que proyecta el explotador, para el proceso de filtrado por color se tienen en cuenta sus píxeles vecinos, calculando el porcentaje de píxeles que pasan el filtro entre los píxeles vecinos que junto con el píxel obtenido forman una *ventana de vecindad* de 5x5 píxeles (figura 4.5).

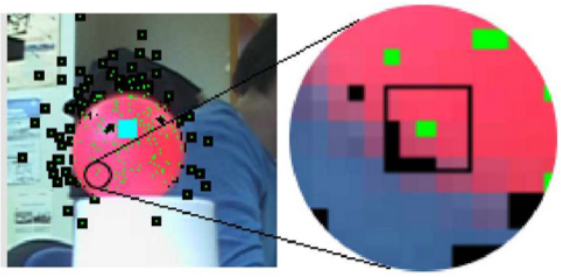

Figura 4.4: Representación de la ventana de vecindad de un píxel.

La proyección del individuo en las cuatro cámaras y posterior análisis de la ventana de vecindad en la imagen de cada cámara permite definir dos criterios sobre los que se basará la ecuación de salud.

- *Porcentaje de píxeles que pasan el filtro de color.* Este criterio mide qué cantidad de píxeles de las vecindades obtenidas superan el filtro de color. Una vez analizadas las cuatro vecindades del individuo sobre el que se evalúa su salud y obtenidos los porcentajes de píxeles para cada vecindad, se realiza un sumatorio de todos los porcentajes. Por tanto este criterio únicamente tiene en cuenta el número de píxeles compatibles con el color buscado. De este modo, los individuos

compatibles con el color de la persona obtendrán mayor salud. La salud máxima se alcanza si todos los píxeles analizados superan el filtro de color.

$$P(\text{color}_i \mid \text{img}_m) \;=\; \frac{\text{k}}{25}$$

k= numero de pixeles que pasan el filtro en la vecindad 5x5 de la imagen m

$$criterio1 \;=\; \sum_{m=1}^{M} f_i * P(\text{color}_i \mid \text{img}_m) \tag{4.1}$$

$$f_i \;=\; \frac{M}{n_i}\ si\ \text{n}_i > 1$$
$$f_i \;=\; 0{,}5\ si\ \text{n}_i = 1$$

n_i = numero de cámaras en las que el individuo ha proyectado bien

En el sumatorio se incluye un factor multiplicativo f_i que permite no penalizar la salud de aquellas razas que siguen a personas que se visualizan en dos o tres cámaras frente a las que se ven en las cuatro, por tanto este factor depende del número de cámaras en que proyecta bien el individuo, siempre que proyecte en más de una. Si una persona se ve bien en el máximo de cámaras posibles para su posición, podrá obtener un valor de salud máxima asociada al color.

- *Porcentaje de coindicencia entre las vencindades.* Mediante este criterio se pretende medir el parecido entre las cuatro vecindades halladas tras la proyección del individuo en cada una de las cuatro imágenes. El parecido se evalúa tomando en cuenta la posición de los píxeles que pasan el filtro de color en la ventana de vecindad, si en cada imagen el píxel de determinada posición supera el filtro significa que en esa posición existe coincidencia. Comprobando todas las posiciones de la vecindad para un individuo se obtiene el porcentaje de coincidencia entre las distintas vecindades.

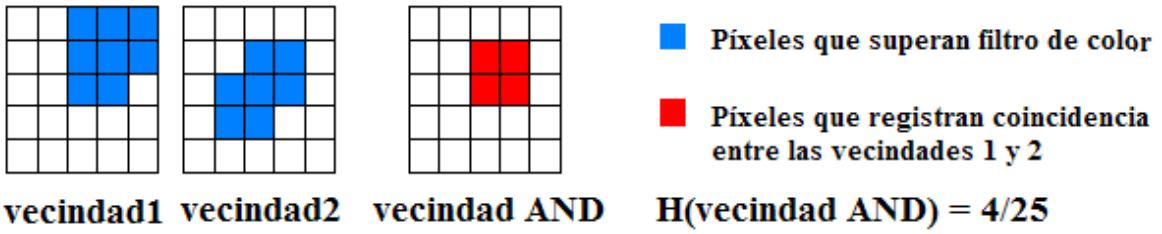

Figura 4.5: Criterio 2 - Ejemplo de cálculo con dos vecindades.

$$vecindad_{AND} = \prod_{m=1}^{M} (\text{vecindad-color}_i \mid \text{img}_m)$$

Sobre la *vecindad AND* se calcula el procentaje de las posiciones en las que se ha hallado coincidencia mediante la función H.

$$criterio2 : H(vecindad_{AND}) = \frac{\text{s}}{25} \tag{4.2}$$

s= numero de posiciones de la vecindad que han registrado coincidencia.

En el caso de que un individuo sólo proyecte sobre una de las imágenes, como es lógico este criterio no sumará nada a la función salud. La utilización de este criterio posibilita que obtengan mayor salud aquellos individuos que se encuentren más cerca espacialmente del objeto, descartando en gran medida los que se hallen en las líneas de proyección sobre cada imagen y estén lejos de la persona, ya que éstos obtienen buena salud sólo a través de una o dos de las cámaras. Esto es de gran importancia para la estimación de la posición 3D real del objeto.

Por tanto, para un determinado individuo su salud asociada al color se calcula mediante la siguiente ecuación, donde alpha y beta permiten la ponderación de los criterios.

$$color_i = \alpha * criterio1_{color} + \beta * criterio2_{color} \tag{4.3}$$

4.4.2. Salud asociada al movimiento

La salud asociada al movimiento se calcula de forma idéntica a la salud definida por color, aplicando esta vez un filtro de movimiento.

$$mov_i = \alpha * criterio1_{mov} + \beta * criterio2_{mov} \tag{4.4}$$

4.5. Dinámica de razas

Una vez descrito cómo se genera cada problación de individuos y se asigna salud a cada uno de ellos, en esta sección se explica cómo se crean las razas y cómo evolucionan, así como las distintas operaciones que se realizan sobre ellas.

4.5.1. Creación de una raza explotadora

Inicialmente el algoritmo utiliza la población de individuos exploradores para detectar movimiento en la habitación. Gracias al operador abducción explicado anteriormente esta población converge en las zonas 3D donde se registra movimiento. Tras la asignación de salud a cada individuo explorador se evalúa si alguno de ellos es prometedor, es decir, si su salud supera o iguala un umbral establecido. Cuando un individuo es prometedor forma una nueva raza de explotadores o se incorpora a otra raza cercana espacialmente a él. Se comprueba si ya existe una raza en esa zona del espacio para aumentar la eficiencia del algoritmo, ya que cada raza consume ciertos recursos y tiempo de análisis.

Una raza inicialmente se compone de individuos idénticos al explorador prometedor que ha provocado su creación. En iteraciones posteriores se opera sobre éstos mediante los operadores genéticos de elitismo y ruido térmico como se explicó en el apartado 4.3.2.

Cada raza tiene varios atributos:

- *Color característico.* Cada raza tiene un color genuino aprendido. El aprendizaje de color se explicará más adelante en la sección 4.7.2.
- *Salud acumulada.* En cada iteración se calcula una salud en base a las observaciones en ese instante de tiempo, sin embargo, una raza puede estar realizando un buen seguimiento pero poseer poca salud instantánea debido a

una oclusión temporal de la visión de la persona seguida en algunas cámaras y esto podría provocar su eliminación. La salud acumulada es un índice que da una idea de qué salud ha ido obteniendo una raza a lo largo de su vida. Esto permite que aunque una raza posea una salud instantánea muy baja, su eliminación no sea inmediata y si se trata de una oclusión temporal la raza recupere salud al visualizarse de nuevo la persona.

- *Tiempo de vida.* Este atributo indica cuánto tiempo lleva existiendo una raza. Permite diferenciar entre razas jóvenes que aún no han definido muy bien su color de aquellas que ya han sido evaluadas muchas veces y por tanto se consideran razas estables.

- *Posición 3D estimada.* Es la media ponderada por salud de las posiciones 3D de todos los individuos de la raza.

4.5.2. Eliminación de razas inconsistentes

En esta etapa del algoritmo se realiza una comprobación sobre las razas con el objetivo de reducir el número de ellas al mínimo necesario para el seguimiento de las personas en la habitación, lo que agiliza el coste computacional del algoritmo.

Se pretende eliminar razas que han perdido consistencia, es decir, razas cuyo valor de salud acumulada está por debajo de un umbral definido. Esto ocurre cuando una raza pierde a la persona que estaba siguiendo, por ejemplo cuando ésta abandona la habitación. Entonces los individuos de su raza correspondiente pierden salud rápidamente al no obtener información sensorial compatible con el color y el movimiento. Otro caso de inconsistencia menos común consiste en que la raza nada más nacer no aprende bien el color característico de la persona, por lo que su salud no aumenta y es eliminada para que otra raza que surja sí pueda realizar el seguimiento de dicha persona.
En el caso de una oclusión temporal, al volver a visualizarse la persona en las imágenes los individuos recuperarían salud y por tanto la salud acumulada de la raza aumentaría sin llegar a descender por debajo del umbral.

4.5.3. Fusión de razas similares

Por otra parte se evalúa la similitud entre razas de acuerdo a: su situación espacial y su color relevante. Al comienzo de cada iteración del algoritmo, si existen poblaciones explotadoras, se comprueba si son semejantes entre ellas. Mediante las

posiciones 3D estimadas almacenadas como atributo para cada raza se halla la distancia espacial entre dos razas, si es menor que una tolerancia establecida se comparan los colores característicos. Si entonces superan un umbral de semejanza, se fusionan.
Por tanto, cuando dos razas se encuentran cerca espacialmente y su color característico es similar se elimina la raza con menor salud acumulada, que de forma general será la menos estable de las dos.

4.6. Tratamiento de la información de movimiento

Tras explicar el algoritmo completo, en este apartado se detalla cómo se extrae la información de movimiento de las imágenes, que junto con el color son las características fundamentales en las que se apoya el cálculo de salud (ver 4.4).

Para la identificación de movimiento dentro de las imágenes se pueden utilizar distintas técnicas. El método más común es el *filtrado de movimiento utilizando dos fotogramas consecutivos* y detectando los cambios de color en la imagen debido al desplazamiento de los objetos. La aplicación de este filtro consiste en restar a cada píxel su valor anterior de RGB para cada una de las imágenes recibidas. Para ello, es necesario almacenar la imágen anterior de cada una de las cámaras que se vayan a analizar, de forma que se pueda contrastar cada píxel con él mismo en el instante anterior. Se establece una tolerancia de movimiento y aquellos píxeles cuya diferencia supere la tolerancia se considera que han sufrido variación debido a un movimiento.

Figura 4.6: Filtro de movimiento - Imagen sin filtrar (a) Imagen filtrada (b).

Un segundo método es la obtención de la *diferencia entre el fotograma actual y de una imagen de fondo aprendida*. Para el aprendizaje del fondo de la habitación en cada cámara se toma una imagen como muestra inicial y sobre ella se van sumando

fotogramas cada ciertas iteraciones, así de forma acumulativa se obtiene una imagen semejante a la que se vería si no apareciera ninguna persona en la habitación, es decir, una imagen de fondo. Como la acumulación continua de fotogramas puede provocar que las imágenes de fondo se distorsionen demasiado, el aprendizaje de fondo se resetea cada aproximadamente cuatro minutos.
La diferencia entre la imagen actual y la de fondo aprendida se genera del mismo modo que en el método anterior.

El primer método permite distinguir muy bien dónde se ha producido movimiento, considerando zonas en las que irrumpe un objeto como aquellas de las que desaparece. El hecho de que esta técnica filtre píxeles pertenecientes tanto a la persona en movimiento como al fondo de la habitación dificulta el aprendizaje de color basado en los píxeles que superan el filtro de movimiento, como se verá en el capítulo de experimentos.
Con el segundo método el resultado será mejor o peor en la medida en que la imagen de fondo aprendida sea buena o no. En general, los píxeles cuyo valor de RGB se asemeje al del fondo serán descartados mediante este filtro.

En este proyecto se ha considerado como píxeles que registran movimiento aquellos que superan *ambas diferencias de forma simultánea*. Por ello se aplica los dos métodos explicados en la detección de movimiento. En el capítulo de experimentos se detallarán los motivos de esta elección y se mostrarán otras implementaciones alternativas.

4.7. Tratamiento de la información de color

Como se ha indicado con anterioridad, una de las dos fuentes de información usada por el algoritmo de seguimiento 3D es el color, que es necesario para el cómputo de salud. En este apartado se muestra el modelo de color utilizado y se describe el proceso de aprendizaje automático del color de las personas localizadas.

4.7.1. Modelo de color HSI

El modelo de color utilizado se basa en el espacio de color HSI que es considerado más robusto frente a cambios de iluminación que el espacio de color RGB, en el que vienen representadas las imágenes de entrada.

Un color en este espacio se representa mediante tres componentes:

- *Tinte* (Hue). Es el color puro propiamente dicho y se corresponde con el ángulo de la figura 4.7.
- *Saturación* (Saturation). Muestra la viveza de un color determinado y se representa en la figura 4.7 como el radio dentro del cono.
- *Intensidad* (Intensity). Es la iluminación del color y se representa como la altura dentro del cono de color.

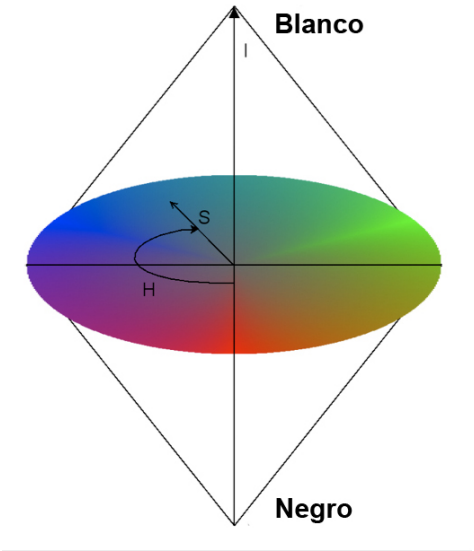

Figura 4.7: Espacio de color HSI.

La relación existente entre RGB y HSI se puede observar en las siguientes ecuaciones:

$$H = cos^{-1}\left(\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right) \tag{4.5}$$

$$S = 1-\frac{3}{(R+G+B)}min\,(R,G,B) \tag{4.6}$$

$$I = \frac{1}{3}(R+G+B) \tag{4.7}$$

Donde R, G, B son la cantidad de rojo, verde y azul, respectivamente, de la imágen con valores entre 0 y 1. H nos proporciona el tinte expresado como un ángulo entre 0 y 2π, S la saturación con valores entre 0 y 1, y la intensidad luminosa I entre 0 y 1.

Por último, para realizar el filtro de color HSI, primero se descartan los píxeles con valores de saturación S por debajo de un umbral y aquellos cuyo valor de iluminación I

se encuentre entre un umbral mínimo y un umbral máximo (establecidos desde el GUI). A continuación, de los píxeles que hayan superado los umbrales anteriores se obtiene la componente H y se comprueba qué probabilidad de ser un color característico de la raza tiene asignada ese matiz según el aprendizaje de color realizado (descrito en el apartado siguiente). Si esa probabilidad es mayor que la probabilidad mínima fijada, entonces el píxel supera el filtro de color. Los píxeles que no superen alguno de los umbrales serán redibujados en escala de grises en la interfaz gráfica de la aplicación o bien, no se dibujarán (ver figura 4.8).

Figura 4.8: Filtro de color para rojo - Imagen sin filtrar (a) Imagen filtrada (b).

4.7.2. Autoaprendizaje de color

El motivo de realizar un aprendizaje de color es acercar el proyecto a una aplicación real. No tener que indicar el color de la ropa de las personas a seguir dota al programa de mucha más autonomía.

Cada raza se caracteriza por un color. El sistema es capaz de aprender el color característico de la vestimenta de cada persona en la habitación. Para ello la aplicación genera y almacena un filtro de color por cada raza basado en la información de color y de movimiento extraída de las imágenes. El movimiento producido por la persona dispara tanto su seguimiento como su aprendizaje de color, es decir, el aprendizaje se realiza sobre los píxeles de movimiento que detecte cada raza.

Como ya se ha comentado, para realizar el aprendizaje de color se utilizan los individuos pertenecientes a la raza de la cuál se quiere definir su color. Éstos permiten el análisis de las imágenes de forma local a la persona que se está siguiendo mediante dicha población.

El método empleado para obtener el color relevante de las personas es la generación de un histograma de la componente H por cada raza, basándose en el modelo de color HSI. La componente H o matiz toma valores entre 0 y 360 e indica el tono de color. Los 360 posibles valores de matices se cuantifican en varios intervalos regulares, en este caso se han tomado 30 intervalos distintos ya que supone un buen compromiso entre rendimiento y eficiencia respecto a los filtros de color.

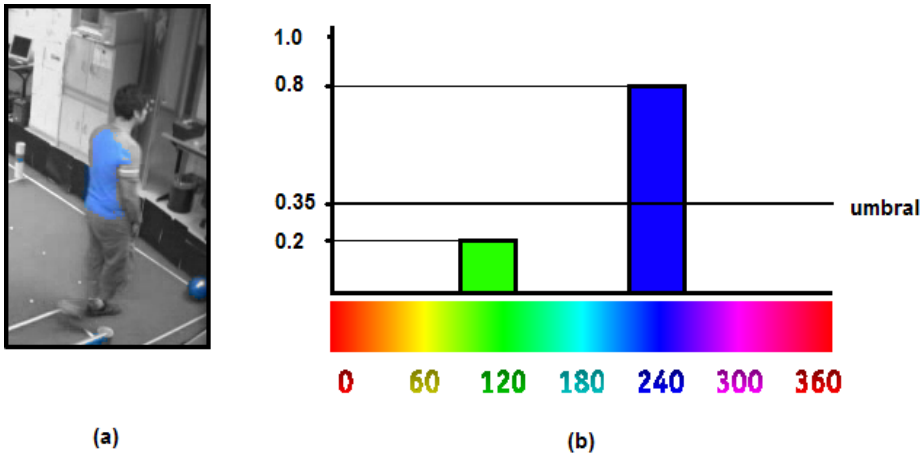

Figura 4.9: Aprendizaje de color - Filtro de color para azul (a), histograma de tintes (b).

Para la generación del histograma que se utilizará en el filtrado se toman muestras en cada iteración, construyendo un histograma parcial en cada una y añadiendo la nueva información al histograma principal de forma acumulativa. Para obtener las muestras del histograma parcial en una iteración se proyectan los individuos de la raza de forma idéntica a como se explicó en el apartado 4.4.1 y se realiza un filtrado por movimiento de aquellos píxeles sobre los que han proyectado. De los píxeles que superen el filtro de movimiento se obtiene su valor HSI de color y se descartan aquellos con saturación muy baja o con un valor de iluminación muy extremo, lo que ayuda a que el filtro generado sea más exigente y rígido, mejorando el seguimiento. El resto de píxeles constituirán las muestras para el histograma parcial de dicha iteración, que se añadirá de forma ponderada al histograma total. De este modo, en cada iteración se obtiene nueva información que se acumula para generar el histograma total. El histograma total se normaliza asignando a cada intervalo un valor entre 0 y 1, que indica con qué probabilidad aparece ese grupo de matices en las imágenes.

$$histograma_{total} = \alpha * histograma_{total} + (1 - \alpha) * histograma_{parcial} \tag{4.8}$$

A través del histograma podemos apreciar fácilmente qué color o colores aparecen con mayor frecuencia. Utilizando un umbral entre 0 y 1 se descartarán aquellos colores que no registren demasiadas muestras en el histograma a modo de filtro de color. La elección del umbral sobre el histograma permitirá decidir si se quiere un filtro más o menos exigente. Esta técnica permite la generación de filtros rígidos y poco flexibles a pequeños cambios. Por ejemplo, si la persona seguida desaparece, el filtro de color no se adapta para que la raza permanezca, sino que ésta pierde salud y se extingue. Además, este método posibilita establecer más de un color característico, lo que resulta muy interesante debido a que las personas no visten de un único color.

El aprendizaje se divide en dos fases que se distinguen en función del tiempo de vida de la raza. Mientras la raza se considera joven, es decir, que se ha creado pocas iteraciones atrás, el aprendizaje de color se realiza en todas las iteraciones. Una vez que la raza se considera estable, la actualización del color se hace cada ciertas iteraciones ya que no es necesario mejorar enseguida el filtro generado.

El autoaprendizaje de color es uno de los objetivos importantes de este proyecto, por lo que se probaron varias alternativas antes de llegar a la solución final que se ha descrito aquí. En el capítulo de experimentos se detallarán las implementaciones alternativas junto con su eficacia y rendimiento.

4.8. Interfaz gráfica de la aplicación

La interfaz gráfica es un elemento importante para la aplicación ya que ha permitido la depuración de errores y se utiliza para la visualización de los resultados de la localización y el seguimiento 3D. La interfaz se presenta en la figura 4.10, distinguiendo cinco partes principales que se explican a continuación. Incluye diferentes elementos para modular y parametrizar el algoritmo así como las herramientas de depuración.

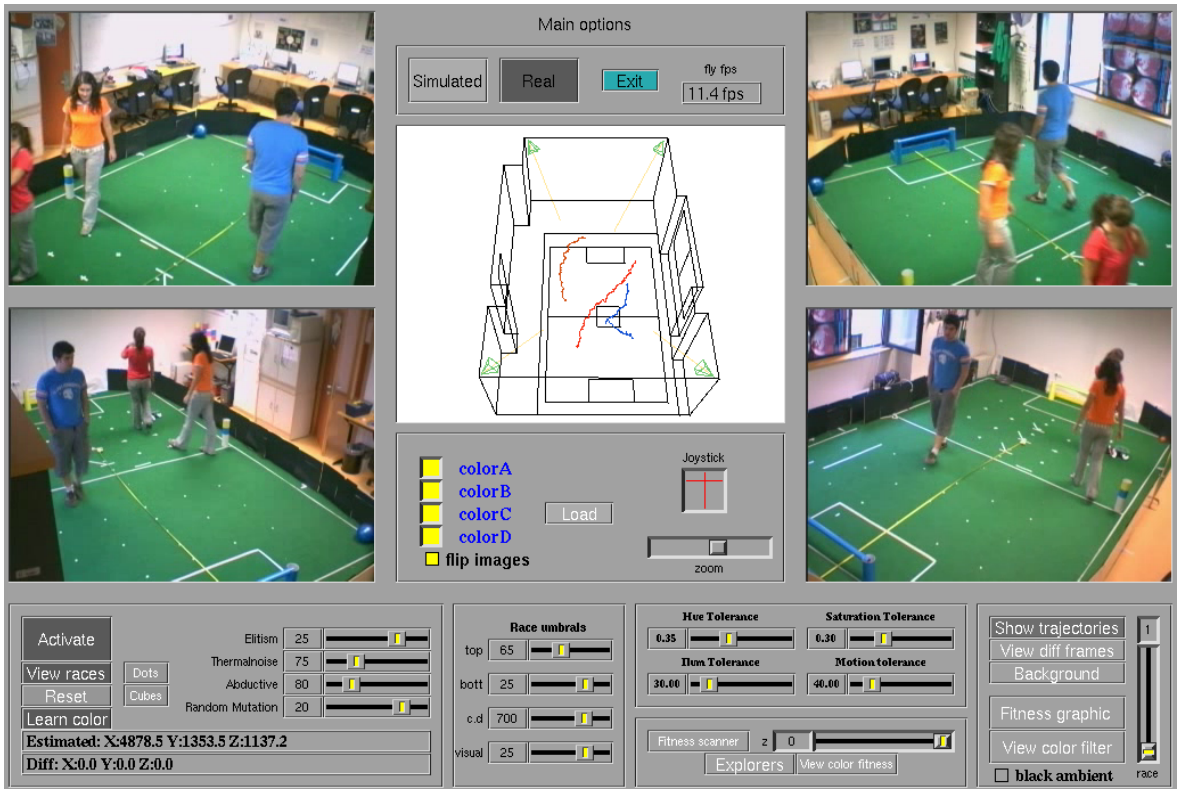

Figura 4.10: Interfaz gráfica de la aplicación de seguimiento visual 3D.

1. *Imágenes de entrada e imagen de la cámara virtual.* Las imágenes de entrada pueden ser reales o simuladas. Las imágenes simuladas se generan externamente a la aplicación a través de otro esquema llamado *Simulator*, que se desarrolló como herramienta de depuración para este proyecto (ver apartado 5.1.1).
   En la imagen de la cámara virtual se representa el estado del seguimiento.

   - *Opciones principales.* Los botones *Real* y *Simulated* permiten seleccionar que las imágenes de entrada sean reales o simuladas. Pulsando *Exit* termina la ejecución de la aplicación y en el campo de texto *fly fps* se puede ver la velocidad del algoritmo en fps.
   - *Imágenes A, B, C y D.* Visualizan las imágenes 2D que capturan las cámaras en la habitación o las imágenes generadas a través del esquema Simulator.
   - *Botones de selección de imágenes y su orientación.* Los botones *colorA, colorB, colorC* y *colorD* por defecto aparecen activados ya que la aplicación presupone que se utilizan cuatro imágenes de entrada, sin embargo es posible desactivar las imágenes deseadas a través de estos botones. La opción *flip images* permite voltear las imágenes de entrada.

- *Imagen de la cámara virtual y su movimiento.* Desde la cámara virtual puede visualizarse una representación 3D de la habitación. A través del controlador *Joystick* o pulsando con el ratón sobre la imagen es posible modificar el punto de vista de la cámara virtual. Mediante el slider *zoom* o utilizando el scroll del ratón sobre la imagen puede alejarse y acercarse la cámara virtual. Por último, el botón *load* permite cargar la configuración de los parámetros por defecto de la cámara, restaurando la imagen virtual inicial.

2. *Control del algoritmo.*

   - Botón *Activate*. Activa o desactiva la ejecución del algoritmo sobre las imágenes de entrada.
   - Botón *View races*. Permite la visualización de la población de explotadores o razas. Los botones *Dots* y *Cubes* activan la visualización de representaciones alternativas de las razas, el primero supone que los individuos se pinten también sobre las imágenes 2D y el segundo provoca que las razas se representen mediante cubos.
   - Botón *Reset*. Pulsando este botón se borra el estado actual del seguimiento y se elimina la población de explotadores, es decir, todas las razas.
   - Botón *Learn color*. Por defecto está activado. Su desactivación implica que la aplicación no utilice la información de color de las imágenes y únicamente funcione en base a la información de movimiento.
   - *Sliders genéticos.* Estas barras de desplazamiento sirven para establecer el porcentaje de individuos que debe someterse a cada operador genético. En el caso de la población de exploradores los porcentajes para *Abductive* y *Random mutation*, y para los explotadores los sliders *Elitism* y *Thermalnoise*.
   - *Posición estimada.* En el primer campo de texto se podrá visualizar la posición instantánea estimada para cada seguimiento. Si se está ejecutando sobre imágenes simuladas puede obtenerse la diferencia entre la posición estimada y la simulada, mostrándola en el segundo campo de texto.
   - *Umbrales de las razas.* Mediante estos sliders se pueden modificar los distintos umbrales que determinan el comportamiento de la dinámica de razas.

3. *Ajuste de los filtros.*

- *Tolerancias del filtro de color*. La interfaz da la posibilidad de ajustar las tolerancias en las componentes HSI del filtro de color.
- *Tolerancia de movimiento*. Ésta fija el umbral para que la diferencia entre dos píxeles se considere como movimiento.

4. *Visualización de exploradores.*

   - *Escáner de salud*. El botón *Fitness scanner* permite que la aplicación entre en modo de depuración. El algoritmo en este modo únicamente distribuye aleatoriamente individuos sobre un plano paralelo al suelo y de altura indicada por el slider z con el fin de evaluar su salud. Si se pulsa además el botón *View fitness* los individuos se pintan de un color en función de su salud, lo que posibilita comprobar el resultado de la función salud fácilmente.
   - *Explorers*. Este botón sirve para visualizar la población de exploradores. También pueden pintarse según su salud pulsando además el botón *View fitness*.

5. *Visualización de trayectorias, filtros de color y movimiento, imágenes de fondo y gráfica de salud de las razas.*

   - *Show trajectories*. Pulsando este botón en la imagen virtual se podrán visualizar las trayectorias que generan los distintos seguimientos que se estén realizando, dibujados con su color característico.
   - *View diff frames*. Este botón permite que en vez de las imágenes de entrada se visualice el resultado del filtro de movimiento en cada una de ellas.
   - *Background*. Las imágenes resultado del aprendizaje del fondo de la habitación se podrán ver en lugar de las imágenes de entrada.
   - *Slider race*. Mediante esta barra de desplazamiento se puede seleccionar la raza de la cual se quiere obtener información como su posición estimada, su salud o su filtro de color.
   - *Fitness graphic*. Al pulsar este botón aparece una nueva ventana que contiene una gráfica de la evolución de la salud de la raza seleccionada.
   - *View color filter*. Sirve para visualizar el resultado del filtro de color sobre cada imagen. Por defecto los píxeles rechazados se pintan en escala de grises, sin embargo pulsando el botón *Black ambient* esos píxeles se dibujan en negro.

Capítulo 5
Herramientas específicas y experimentos

Los experimentos, ajustes y observaciones realizados durante todo el proceso de creación de este proyecto se presentan en este capítulo. Una gran cantidad de pruebas y experimentos fueron necesarios para ajustar los parámetros de funcionamiento del algoritmo de seguimiento así como para comparar las distintas implementaciones alternativas. Asímismo, permitieron validar el buen funcionamiento del sistema final.

En primer lugar se describen las herramientas generadas para la depuración y ajuste del algoritmo. A continuación se muestra la ejecución típica y posteriormente los experimentos realizados tanto con color predefinido como aprendido y las pruebas con distintas fuentes de información para detectar movimiento. Por último se muestra una ejecución con los parámetros del algoritmo ajustados para objetos pequeños.

El escenario de pruebas es idéntico al que se utilizó en [Cabello, 2006]. El entorno concreto es el Laboratorio de Robótica del Edificio Departamental II de esta universidad y el montaje físico consiste en la instalación de cuatro cámaras web adheridas al techo de la habitación conectadas cada una a un ordenador formando todos una red local.

Figura 5.1: Escenario de pruebas - Equipo principal (a) y cámara Apple iSight (b).

La aplicación fue ejecutada en otro equipo del laboratorio conectado a esta misma

red. Las características de este equipo son: Pentium 4 con función de Hyperthreading a 2,6 Ghz, 512 MB de RAM y sistema operativo GNU/Linux (figura 5.1 (a)).

5.1. Herramientas de depuración

Durante todo el proceso de desarrollo de la aplicación se crearon herramientas para el ajuste y la depuración del algoritmo, que se detallan a continuación.

5.1.1. Esquema simulador

Este esquema simula cuatro imágenes de la misma escena 3D emulada, por lo que permite probar el sistema en un entorno simulado. La aplicación es capaz de recoger las imágenes generadas por este esquema e interpretarlas del mismo modo que si fueran de cámaras reales. Estas imágenes (figura 5.2 (a)) representan esquemáticamente la habitación, de forma similar a la cámara virtual del GUI. Los objetos simulados son cubos de diferentes colores (figura 5.2 (b)) cuyas posiciones se pueden controlar desde la interfaz o aplicarles un movimiento automático continuado trazando, por ejemplo, una elipse.

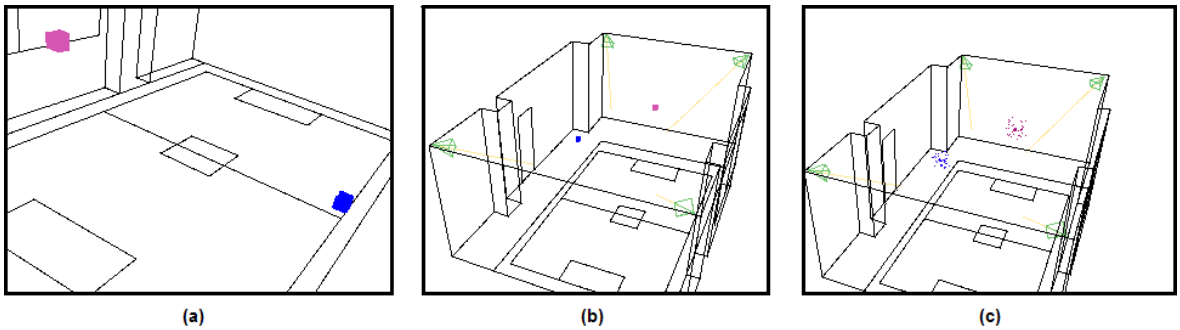

Figura 5.2: Imágenes del simulador - Imagen cámara D (a), objetos simulados (b) y representación de la solución proporciondada por la aplicación de seguimiento (c).

El esquema proporciona las posiciones 3D exactas de los objetos simulados y las cámaras perfectamente calibradas.
La utilización de este esquema resultó de gran ayuda para la depuración en el inicio de la implementación del algoritmo de seguimiento. Además, permitió realizar un análisis de la precisión del sistema mediante la comparación entre las posiciones simuladas y las estimadas, que se presentará más adelante.

5.1.2. Visualización de individuos exploradores y escáner

Visualizar la población de individuos exploradores tanto en la cámara virtual como proyectada en las imágenes fue muy importante para la depuración. Esta población permite realizar una búsqueda de las zonas donde se produce movimiento (figura 5.3), por lo que su comportamiento influye directamente en el funcionamiento de la aplicación.

Una forma de mostrar fácilmente la salud que obtiene cada individuo de esta población fue dibujarlas en pantalla de un color u otro según la salud que poseían. Así, se eligieron tres rangos de salud y tres colores para diferenciarlos. El objetivo era comprobar la eficacia de la función salud, por lo que en las zonas donde se produjera movimiento los individuos debían ser del color que representa una salud elevada. Esto sirvió de gran ayuda para ajustar la función salud y depurarla.

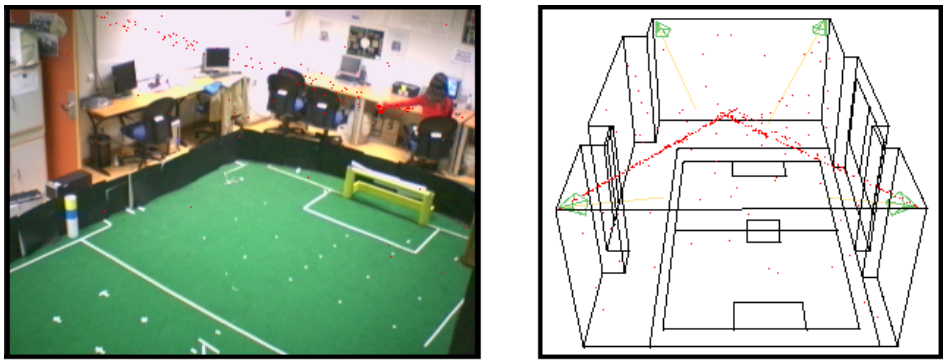

Figura 5.3: Visualización de exploradores - Imagen cámara A (a) y población de exploradores (b).

Otro método que permitió mejorar la función salud fue realizar un escáner. El escáner consistía en distribuir aleatoriamente una cantidad elevada de individuos sobre un plano paralelo al suelo de la habitación, pudiendo seleccionar la altura de este plano. Visualizando los individuos según su salud se pudo observar de forma más clara la correcta asignación de salud a los individuos tras producirse un movimiento en la sala.

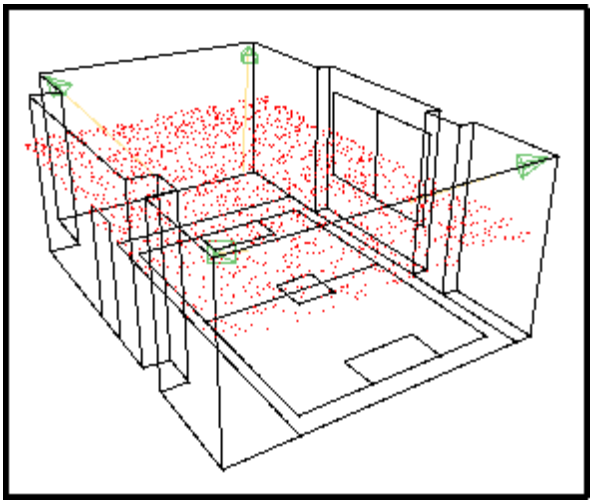

Figura 5.4: Visualización del escáner en la cámara virtual.

5.2. Ejecución típica

En la ejecución típica se puede comprobar que el sistema es capaz de seguir a varias personas dentro de la habitación con vivacidad y precisión, sin conocer ningún tipo de información inicial sobre éstas y realizando un aprendizaje de color para cada una.

Utilizando las cuatro cámaras de la habitación y procesando las imágenes para extraer información de color y de movimiento, se estableció una población de 400 individuos exploradores y 50 individuos explotadores por cada raza surgida. Inicialmente se distribuye uniformemente la población de individuos exploradores por todo el espacio de la habitación, en posteriores iteraciones esta población converge en aquellas zonas donde se detecta movimiento y nacen las distintas poblaciones de individuos explotadores o razas. Mediante las razas se realiza el seguimiento en tres dimensiones y se recoge información de color sobre las personas a las que siguen, aprendiendo automáticamente el color de cada una. Si una persona abandona la habitación, la raza que está realizando su seguimiento se elimina. Así, las razas dinámicamente surgen y desaparecen de acuerdo con las observaciones en las imágenes.

En la figura 5.5 se ve a tres personas caminando por la sala y la representación del estado del seguimiento en la cámara virtual. Cada nube de puntos representa una raza así como una persona localizada y además se dibujan del color aprendido.
Otra representación del seguimiento se realiza mediante trayectorias. El sistema

almacena un pequeño histórico de las posiciones que ha ido calculando para cada persona.

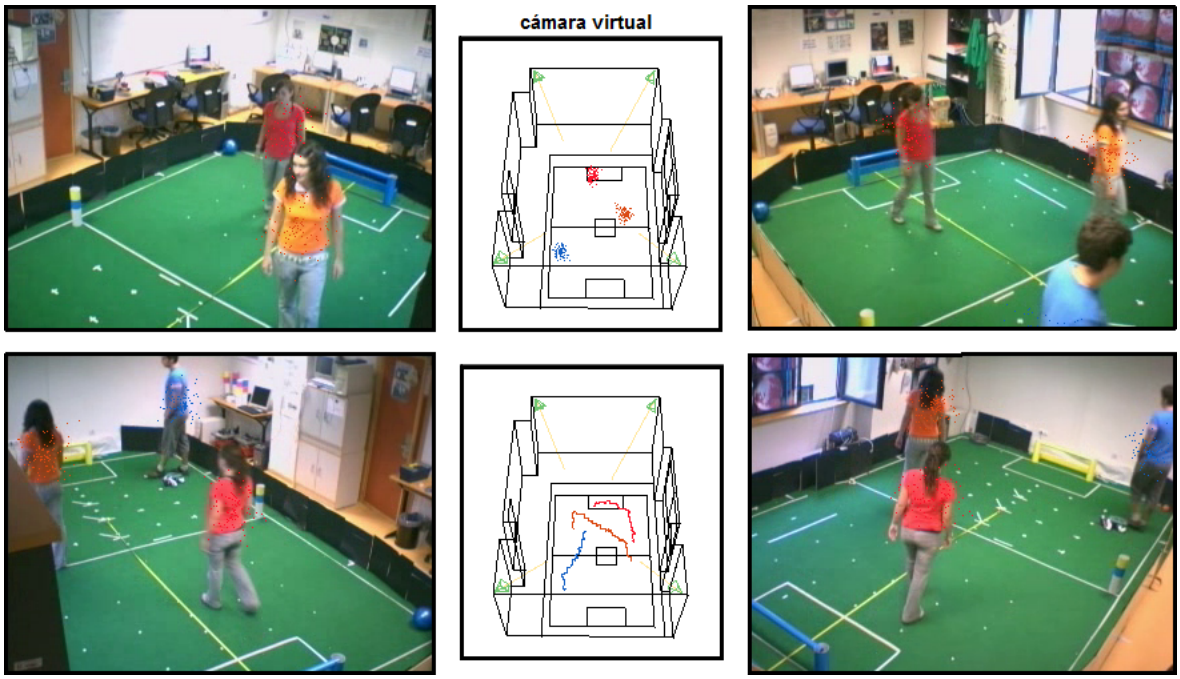

Figura 5.5: Ejecución típica.

La velocidad del algoritmo en la ejecución típica es de 12 ips aproximadamente, funcionando con cámaras a 30 fps. Esta velocidad es suficiente para seguir a personas moviéndose naturalmente dentro de la habitación.

La vista a modo de planta de la habitación permite comprobar que la estimación de la posición es bastante buena. En la siguiente sección se presentan las pruebas realizadas para conocer la precisión de la aplicación.

5.2.1. Precisión

Para caracterizar la precisión del sistema se utilizó el esquema simulador y se realizó una gráfica que representa la evolución en el tiempo de la posición simulada frente a la estimada, es decir, el error cometido. En las pruebas se tomaron datos con un único objeto tanto en movimiento como quieto y visualizándose en distinto número de cámaras.

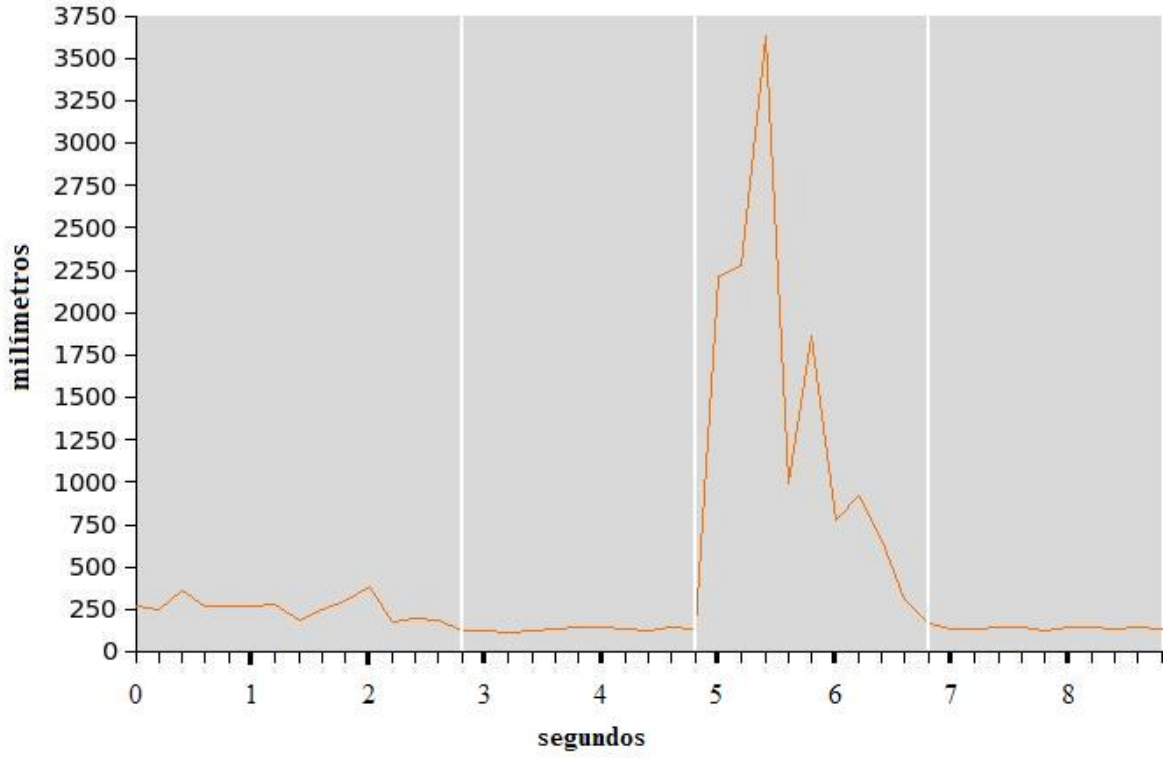

Figura 5.6: Gráfica del error cometido en la estimación de la posición simulada con 4 cámaras.

La gráfica muestra que la precisión en la localización es de orden centimétrico. Inicialmente el objeto está en movimiento (segundos 1 a 2,8) y la diferencia entre la posición estimada y la simulada se mantiene alrededor de los 200mm, lo que es lógico puesto que esta técnica de localización no es predictiva y la posición estimada en movimiento siempre irá algo retrasada respecto a la real. Una vez parado el objeto (segundo 2,8), el error desciende y se mantiene aproximadamente entre 50 y 100mm. Esta medida no cambia si se utilizan dos o tres cámaras en lugar de cuatro. A continuación (segundo 4,8) se colocó el objeto en otra posición diferente de modo repentino con el fin de observar el tiempo de reacción del algoritmo para estimar la nueva posición. En la gráfica se puede apreciar que el error aumenta considerablemente hasta que tras dos segundos encuentra la nueva posición y el error disminuye a niveles normales.

Una segunda prueba de precisión se realizó en el laboratorio real. Se tomaron dos medidas a diferentes distancias del origen de coordenadas para contrastar las posiciones 3D reales y teóricas, expresadas en milímetros. Para ello se utilizó a una persona con camiseta de color rojo y una camiseta de color naranja, que una vez conseguida

su localización mediante movimiento se dejó en el suelo. Como la persona ocupa un volumen elevado, se tomó como punto de referencia el centro de su torso. Para una posición real de la camiseta en P1(x,y,z) = (2830, 2000, 0) se obtuvo una posición estimada P1'(2750, 2140, 58) y para la posición de la persona P2(4900, 3320, 1200) se obtuvo P2'(5067, 3375, 1242).
Estos resultados indican un error cometido del orden de centímetros. Se debe contemplar que siempre se introduce un cierto error de precisión al realizar una calibración manual de las cámaras y al considerar que los objetos no tienen tamaño. A pesar de ello, se confirma que *la técnica es suficientemente robusta para detectar a personas con gran precisión*. Además, se ha observado que la precisión es la misma en diferentes zonas de la sala donde hay cobertura de al menos dos cámaras.

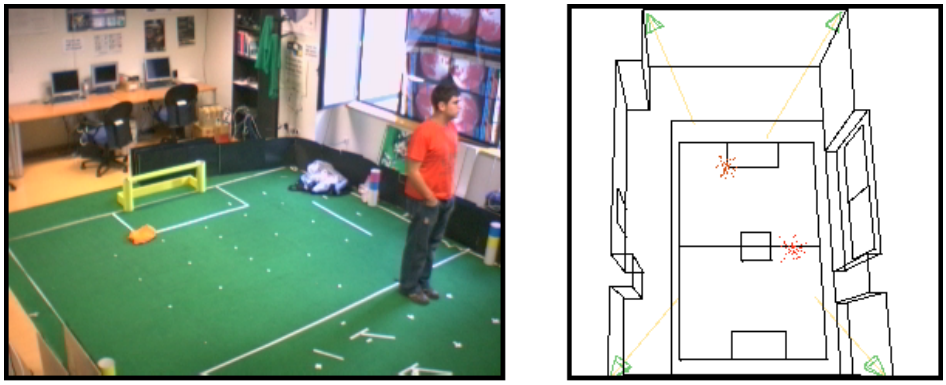

Figura 5.7: Prueba de precisión - Imagen cámara B (a), representación de la solución (b).

5.3. Seguimiento utilizando sólo movimiento

En este experimento se quiso contrastar el funcionamiento del programa con y sin información de color.

Eliminando el color como fuente de información, el algoritmo únicamente utiliza el movimiento para localizar y realizar un seguimiento sobre las personas. El experimento consistió en que una persona anduviera por la habitación y se detuviera en varias ocasiones. Como se esperaba, la aplicación conseguía mantener localizada a la persona mientras que ésta estaba desplazándose, sin embargo la localización se perdía al pararse la persona (figura 5.8).

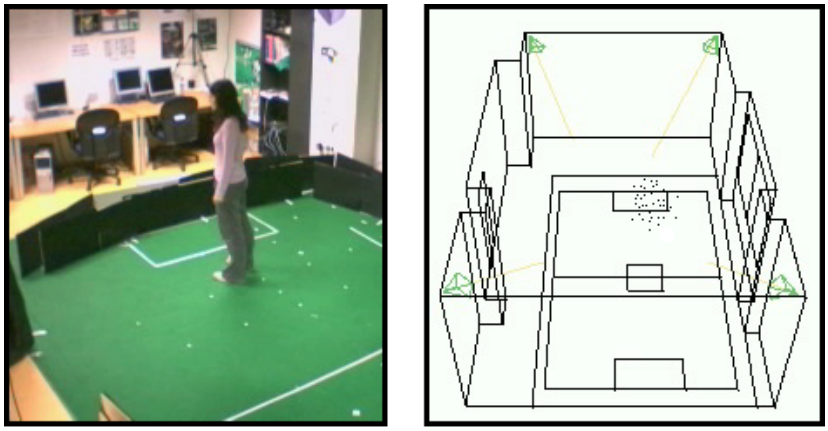

Figura 5.8: Ejecución sin aprendizaje de color - Persona quieta (a), raza desapareciendo (b)

Al realizar el mismo experimento utilizando el color y el movimiento como fuentes de información el resultado fue distinto. Mientras se movía la persona por la habitación la aplicación mantenía su seguimiento aprendiéndose el color y al detenerse la localización continuaba gracias a la información de color almacenada. Por tanto, la incorporación de la información de color resulta de gran necesidad para este proyecto, pues permite continuar el seguimiento de personas aunque éstas se detengan.

5.4. Ajustes del algoritmo de seguimiento

La realización de las pruebas que se presentan a continuación permitieron comprobar el funcionamiento del algoritmo evolutivo sin aprendizaje de color, puesto que son problemas distintos y tanto su evalución como su ajuste puede realizarse por separado.

Para el ajuste incial del algoritmo se probó la aplicación indicando el color de la persona o personas a seguir. Lo que se pretendía era ajustar algunos parámetros del algoritmo específicos del seguimiento 3D sin la intervención del aprendizaje de color, como por ejemplo los porcentajes asignados a cada operador genético y ciertos umbrales referentes a la salud.

La configuración para la que se obtuvo mejor resultado fue asignando en la población de exploradores un 80 % al operador abducción y un 20 % al operador mutación

aleatoria. Esto significa que un 80 % de la población de individuos exploradores son generados mediante el operador abducción. Experimentalmente se comprobó que este operador agiliza notablemente la búsqueda de movimiento en la habitación y provoca la convergencia inmediata de la población en las zonas de movimiento. El resto de los individuos se generan de forma aleatoria, lo que permite no dejar de explorar todas las zonas en todo momento.
En la población de explotadores se fijó un 25 % para el operador elitismo y un 75 % para ruido térmico. En este caso es importante la exploración local que realiza el 75 % de los individuos para conseguir realizar un buen seguimiento de una persona desplazándose.

Otros parámetos ajustados fueron las tolerancias tanto para el filtro de color como para el filtro de movimiento. Observando una persona moverse por las distintas zonas de la habitación se puede comprobar que los cambios de iluminación son considerables. No obstante se decidió no utilizar focos sino únicamente la iluminación de la sala y la luz que entraba por la ventana como escenario típico para la ejecución de la aplicación. Por este motivo la tolerancia del filtro en la componente de iluminación del modelo de color HSI empleado es elevada.

Tras realizar estos ajustes, se comprobó el funcionamiento del algoritmo implementado observando el correcto seguimiento de dos personas vestidas de color rojo (figura 5.9) y se realizaron pruebas para evaluar el funcionamiento de la dinámica de razas: creación, fusión y eliminación de razas.

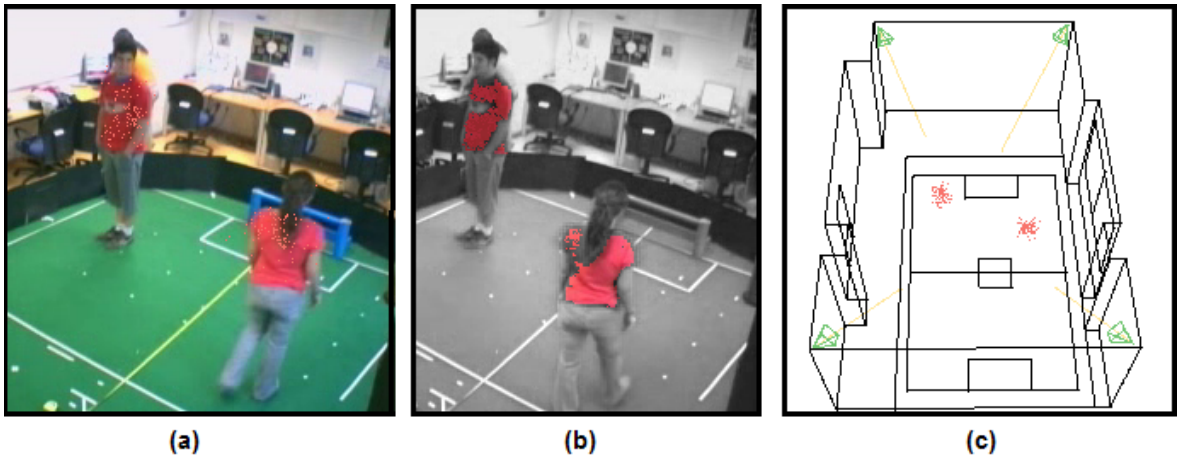

Figura 5.9: Experimento con color predefinido - Imagen sin filtrar (a), imagen filtrada (b) y representación de la solución (c).

La dinámica de razas está muy determinada por ciertos umbrales relacionados con la salud de las poblaciones de individuos. Primero se ajustó el umbral de salud que al ser superado indica que un individuo explorador es prometedor. Al disponer de información de color a priori la salud de la población de exploradores se definió en base al color y al movimiento, por lo que este umbral se fijó a un 85 % de la salud máxima posible. Seguidamente se ajustó el umbral que indica el mínimo de salud que debe mantener una raza para que no sea eliminada, en este caso se determinó a un 50 % de la salud máxima.

Estos umbrales influyen notablemente en el funcionamiento del algoritmo. Si se fijan a unos valores altos, no surgen razas debido a que los individuos exploradores no alcanzan la salud prometedora. Si se dejan estos umbrales demasiado bajos, aparecen muchas razas y la mayoría son redundantes, lo que empeora el seguimiento.

La distancia espacial entre razas para determinar si son similares también es un umbral importante, en esta prueba se fijó en 500mm, lo que supuso mejorar el funcionamiento del algoritmo en general ya que eliminaba las razas similares y esto elevaba el rendimiento de la aplicación.

Por último, un ajuste de gran importancia para el funcionamiento del algoritmo es la ponderación de los dos criterios utilizados en la función salud (ver 4.4), que se realizó mediante pruebas y se determinó que el criterio que mide el parecido entre vecindades (ecuación 4.2) debía tener más peso que el criterio que calcula la cantidad de píxeles relevantes (ecuación 4.1), concretamente el 75 % de la salud tanto por color como por movimiento. Si en las vecindades se halla un parecido considerable hay más probabilidad de que los individuos estén espacialmente situados cerca de la posición real de la persona, lo que implica una mejor estimación de su posición y por tanto un mejor seguimiento.

5.5. Experimentos con autoaprendizaje de color

Una vez ajustado el algoritmo de seguimiento y verificado su correcto funcionamiento se procedió a evaluar el aprendizaje de color y su influencia en la ejecución del algoritmo. El objetivo era aprender el color de la vestimenta de cada persona sobre la que surge un seguimiento, para ello se exploraron varias técnicas que

se detallan a continuación.

5.5.1. Aprendizaje a través de media RGB

En primer lugar se implementó un aprendizaje de color que consistió en realizar una media RGB del valor de los píxeles interesantes. La principal dificultad residía en averiguar qué píxeles debían considerarse interesantes.
Inicialmente se emplearon los píxeles cuya diferencia entre fotogramas consecutivos superaba una cierta tolerancia. El color resultante era semejante al que vestía la persona pero el filtro generado era muy amplio y filtraba parte del fondo, lo que era lógico ya que se producía una mezcla entre los valores de píxeles del fondo y los pertenecientes a la persona. Este método mostrabra problemas si la persona vestía de un color similar a ciertas zonas del fondo de la habitación, en este caso poblaciones de individuos explotadores permanecían en esas zonas aunque la persona las hubiera abandonado debido a que el color encajaba bien con el filtro de esas razas.

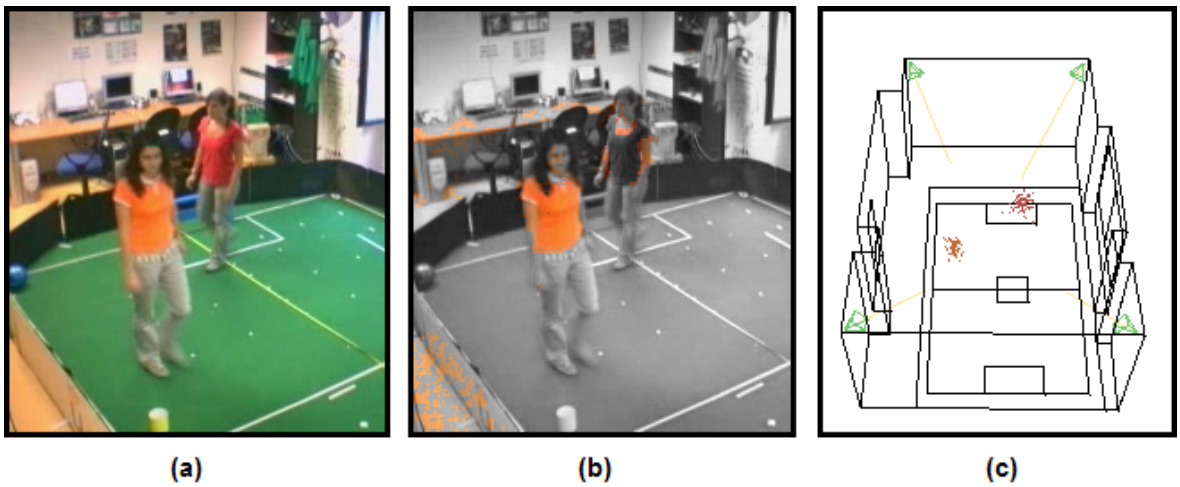

Figura 5.10: Filtro con media RGB - Imagen sin filtrar (a), imagen filtrada (b) y representación de la solución (c).

Se reconsideró sobre qué píxeles aprender el color, considerando esta vez aquellos que superaran la diferencia respecto al fotograma anterior y respecto al fondo, tal y como se describe en la sección 4.6. Los resultados obtenidos al utilizar este método de aprendizaje del color eran aceptables y el seguimiento de múltiples personas se realizaba con éxito. Sin embargo, la inestabilidad del filtro generado realizando una media RGB provocó que se decidiera probar otro método de recolección de información de color.

5.5.2. Aprendizaje mediante histograma de matices

Un histograma de la componente H en el modelo de color HSI sobre los píxeles interesantes fue la otra técnica planteada y finalmente la solución adoptada.

Las mejoras de esta técnica respecto al método anterior son notables. Esta forma de acumular el color recogido de los píxeles interesantes no mezcla los matices de color sino que asigna una probabilidad a cada uno y se selecciona el que tenga la más alta, lo que implica mayor rigidez en el filtro. Este método es menos sensible ante cambios espúreos de color que la media RGB, dotando de mucha más estabilidad al filtro de color aprendido. Al visualizar el filtro de color de cada raza en la interfaz se aprecia fácilmente que el color filtrado está muy definido y no pasan el filtro colores del fondo que no sean realmente similares al de la persona. En la figura 5.11 se puede observar el resultado del filtro para una raza que sigue a una persona vestida de color naranja.

Figura 5.11: Filtro mediante histograma de H - Imagen sin filtrar (a) y filtrada (b).

El número de intervalos del histograma es 30, por lo que se consideran 30 rangos de matices como posibles colores característicos. Esta cantidad de intervalos permite diferenciar muy bien los matices de color, agrupando aquellos que están cercanos en el espacio de H. Este número repercute directamente en cómo de tolerante es el filtro de color, por lo que interesa llegar a un equilibrio.

Eligiendo un umbral de probabilidad adecuado sobre el histograma se decide a modo de filtro qué colores aparecen más frecuentemente, es decir, se determina la probabilidad mínima que debe presentar un matiz para considerarse color característico de la persona. El umbral escogido es 0.35 sobre 1, ya que elimina los colores con baja probabilidad.

Otra de las ventajas de este método es que mediante el histograma se puede definir más de un color característico para una raza, esto es así ya que cada intervalo de valores de H registra un número determinado de muestras y si dos intervalos diferentes registran un elevado número de ellas, se considera que esa población se identifica con dos colores distintos. Esto es muy interesante para la aplicación debido a que las personas no suelen vestir de un único color.

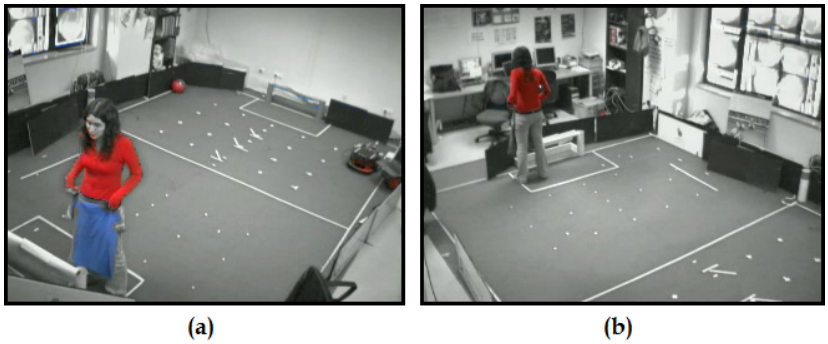

Figura 5.12: Filtro para dos colores distintos - Imagen filtrada cámara D (a), imagen filtrada cámara B (b).

La desventaja de esta técnica es que no aprende colores claros ni oscuros. Al realizar el histograma únicamente sobre la componente H del modelo HSI, los colores blanco y negro por ejemplo no se corresponden con ningún rango específico del histograma, por lo que son descartados.

El rendimiento computacional de la aplicación utilizando esta técnica de aprendizaje es menor debido a que es más costoso computacionalmente realizar un histograma por cada raza surgida que una media RGB. Sin embargo, los resultados obtenidos son considerablemente mejores, por lo que compensa utilizarla. Concretamente la velocidad del algoritmo desciende entre 1 y 2 ips.

El incluir el aprendizaje de color en el algoritmo supuso tener que reajustar ciertos parámetros del algoritmo de seguimiento. Uno de ellos fue el umbral de salud mínima para considerar que un individuo es prometedor. Al no disponer de información a priori de color la salud de los individuos exploradores únicamente se mide en referencia a la información de movimiento percibida en las imágenes, por lo que es lógico que al tener menos información inicial la salud sea menor y por tanto este umbral deba ser más bajo. Tras realizar varias pruebas se fijó a un 65 % de la salud máxima total, algo

menor que en el caso de conocer el color a priori.
Otro umbral muy relacionado con el anterior que hubo de reajustarse es la salud mínima que debe tener una raza para no ser eliminada, que lógicamente también disminuye ya que si no fuera así nada más nacer la salud de una raza estaría por debajo de este umbral. Se fija a un 25 % de la salud máxima.

La estabilidad y la rigidez del filtro de color eran los objetivos buscados, mediante esta técnica se logró un compromiso entre la eficiencia y la estabilidad. Esto repercute directamente en el seguimiento, suponiendo una mejora considerable.

El aprendizaje del color relevante de las personas dota a la aplicación de gran autonomía ya que éstas no deben vestir de un color determinado y esto compensa la ligera disminución del rendimiento computacional del algoritmo.

5.6. Técnicas para detectar movimiento

El movimiento extraído de las imágenes es de gran importancia para la aplicación de seguimiento tanto en la asignación de salud de los individuos como en el autoaprendizaje de color. Las principales características que se evualuaron en cada técnica fueron el coste computacional requerido y la eficacia en los resultados tanto para la salud como para el aprendizaje de color. A continuación se describen los diferentes métodos probados.

5.6.1. Diferencia entre fotogramas consecutivos

Esta técnica es la que se utilizó inicialmente debido a su sencillez en la implementación y los buenos resultados obtenidos para la función salud.
Como anteriormente se explicó (ver sección 4.6), el método consiste en hallar la diferencia absoluta entre una imagen y la inmediatamente anterior en las componentes RGB de cada píxel. Si la diferencia entre dos componentes simultáneamente superan el umbral establecido como mínimo movimiento, se considera que ese píxel pasa el filtro de movimiento.

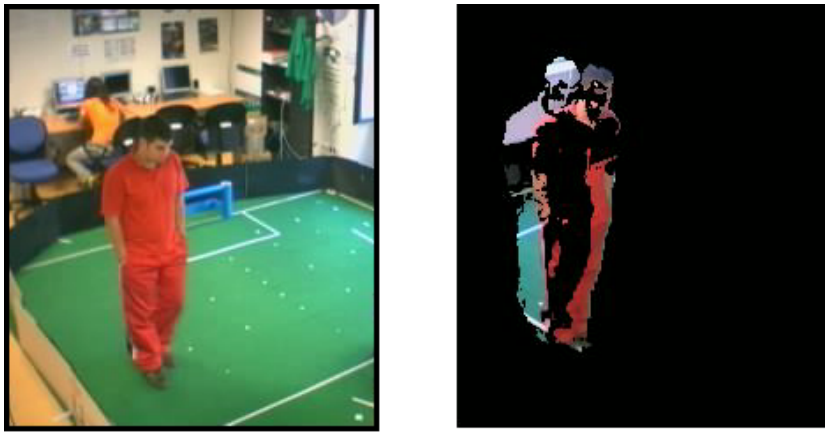

Figura 5.13: Filtro por diferencia entre fotogramas - Imagen sin filtrar (a), imagen filtrada (b)

Esta técnica no supone un alto coste computacional y los resultados en la detección de movimiento para la salud son muy buenos, sin embargo para el autoaprendizaje de color no es eficaz. Considera que se ha producido movimiento tanto en las apariciones como en las desapariciones de una persona, es decir, el filtro es superado tanto por el borde de la persona en la dirección en que avanza como por el borde que deja atrás. En este último los píxeles no pertenecen a la persona sino al fondo, por lo que utilizándolos en el aprendizaje de color distorsionan el color original de la persona. Por ello, esta técnica se descartó debido a que no era eficaz para el aprendizaje de color.

5.6.2. Diferencia respecto al fondo de la habitación

Con la finalidad de solucionar principalmente el problema del aprendizaje de color se decidió almacenar información sobre el fondo de la habitación. Aprendiendo a diferenciar los píxeles que pertenecen al fondo, se pueden descartar directamente y no tenerlos en cuenta para el aprendizaje de color.

El método empleado es el aprendizaje de cuatro imágenes de fondo, una por cada cámara. Tomando una imagen inicial, de forma acumulativa se va sumando ponderadamente a esta imagen otra captada cada ciertas iteraciones. Así se consigue obtener una imagen parecida a la que se vería si no hubiera nadie en la habitación. En la figura 5.14 (b) se puede apreciar cómo en la imagen aprendida aparecen personas difuminadas, resultado de la suma de fotogramas, sin embargo la imagen de fondo se asimila a la que se obtendría si no hubiera nadie en la habitación.

La calidad de las imágenes aprendidas depende de la imagen inicial captada, de la ponderación en la suma de las imágenes y de las iteraciones escogidas como intervalo para añadir nueva información. En esta implementación se ha establecido un peso de un 10 % para la imagen nueva en la suma con la imagen de fondo almacenada (ecuación 4.8) y que se incorpore una nueva imagen al fondo aprendido cada 150 iteraciones.

Figura 5.14: Fondo aprendido-Imagen instantánea (a), imagen de fondo aprendida (b)

En esta técnica de detección de movimiento la diferencia se realiza entre la imágenes actuales y las imágenes de fondo aprendidas, de manera similar que entre fotogramas consecutivos.
En el resultado de esta diferencia se distingue muy bien a las personas en la habitación y en general muchos más píxeles superan el filtro, no sólo los bordes de las personas.

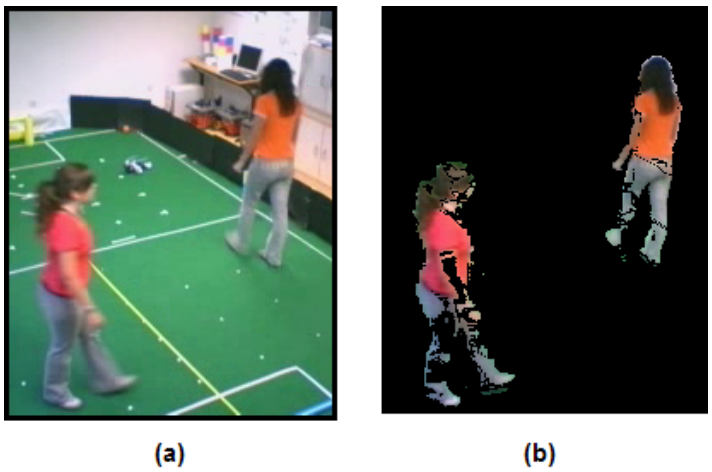

Figura 5.15: Diferencia con el fondo - Imagen sin filtrar (a) y filtrada (b)

El aprendizaje de estas imágenes de fondo se lleva a cabo aprovechando que ya se realiza un filtrado completo por movimiento de las imágenes para asignar salud a la población de individuos exploradores. Esto supone que la velocidad del algoritmo apenas disminuya levemente. Este incremento del coste se compensa debido a que mejora los píxeles interesantes sobre los que se aprende el color.

En referencia a la salud, se produce un incremenento en los valores asignados en la salud por movimiento. Esto es provocado por el aumento del número de píxeles que superan el filtro. Con esto el algoritmo de seguimiento tiene mayor facilidad en la creación de razas, por lo que hubo que reajustar los umbrales relacionados con la salud.

El aprendizaje de color es así más robusto puesto que se utilizan en mayor medida los píxeles correspondientes al interior de las personas y no sólo a sus bordes, descartando los que se identifican con el fondo.

Sin embargo, con esta técnica el algoritmo depende excesivamente de la calidad del fondo aprendido, lo que es muy arriesgado. Realizando pruebas se ha podido comprobar que si una persona aparece en la imagen inicial para el aprendizaje o se queda quieta durante mucho tiempo en una zona, la aplicación toma a esa persona como parte del fondo. Así, al retirarse de esa posición la diferencia respecto al fondo va a perdurar de forma prolongada hasta que lentamente vaya desapareciendo según se recoge nueva información. En esta situación se generaría una raza en la posición que la persona había abandonado, lo que no es correcto. Por este motivo se descartó esta técnica.

5.6.3. Diferencia simultánea con el fotograma anterior y el fondo

La combinación entre las dos técnicas anteriores (5.6.1 y 5.6.2) permite extraer los píxeles que superen el umbral de movimiento entre dos fotogramas consecutivos y descartar aquellos que se asemejen al fondo de la habitación.

La forma de implementar este método es similar a las anteriores y utilizando como criterio del filtro ambas diferencias a la vez. La carga computacional es semejante al caso anterior, aunque en este caso se trabaja a la vez con tres imágenes distintas: la imagen actual, la imagen anterior y la de fondo aprendida.

En cuanto a la función salud los resultados son buenos, en este caso no se necesita un reajuste de parámetros.

Respecto al aprendizaje de color, mediante esta técnica se obtienen los mejores resultados. A pesar de que es menor el número de píxeles que superan el filtro en comparación con los otros métodos, éstos pertenecen con gran seguridad a las personas que se pretende localizar. Por ello esta técnica es la que finalmente se ha escogido una vez se ha experimentado con todas y comparado sus rendimientos.

Figura 5.16: Filtro combinación - Imagen sin filtrar (a), imagen filtrada (b).

5.6.4. Detección de movimiento por flujo óptico

Otro de los experimentos realizados con el fin de analizar y extraer movimiento de las imágenes fue mediante la utilización de flujo óptico.

Las técnicas de flujo óptico estiman el movimiento a través de los cambios de intensidad en una secuencia de imágenes. El flujo óptico hace corresponder a un píxel del fotograma actual otro píxel en el fotograma del instante posterior, si ha detectado un cambio de intensidad. El proyecto fin de carrera *Cálculo y aplicaciones del flujo óptico en tiempo real* [Cadenas, 2007] materializa esta técnica.

Aplicado a este problema, el flujo óptico mide el movimiento generado por una persona en la habitación. Esto permite, al igual que en las otras técnicas, conocer en qué píxeles se ha detectado movimiento, pero además proporciona información acerca de la magnitud de movimiento medida y la dirección de éste. En este proyecto sólo

interesaba obtener la lista de píxeles en los que ha habido movimiento.

La implementación consistió en utilizar el esquema (figura 5.17) de [Cadenas, 2007] que calcula el flujo óptico. Generando cuatro instancias de este esquema, una por cada cámara, nuestra aplicación de seguimiento importaba los resultados de esos esquemas y los analizaba.

Figura 5.17: Esquema opflow.

Inicialmente se tomaron en cuenta para el algoritmo únicamente las mediciones realizadas por esos esquemas. Tras analizar los resultados se determinaban los píxeles interesantes, sin embargo la cantidad de estos píxeles era excesivamente pequeña, por lo que el algoritmo no funcionaba correctamente. El movimiento en esta aplicación es el disparador del seguimiento, por lo que al no detectarse con facilidad, el seguimiento resultaba imposible.

Para solucionar este problema se combinó el cálculo del flujo óptico con la primera técnica expuesta, la diferencia absoluta entre fotogramas consecutivos. De este modo, se consideraba como píxeles interesantes los que registraran una diferencia considerable entre fotogramas y además estuvieran cercanos a los píxeles indicados por el cálculo del flujo óptico, concretamente que pertenecieran a la ventana de vecindad de 7x7. Mediante esta combinación el algoritmo disponía de suficiente información de movimiento, por lo que el seguimiento podía realizarse.

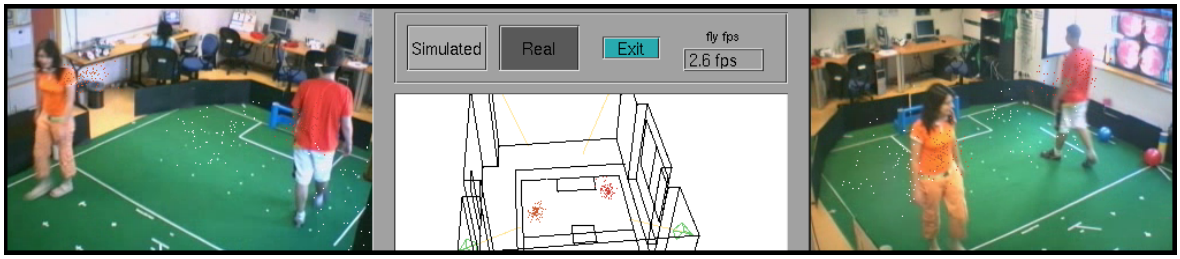

Figura 5.18: Ejecución utilizando flujo óptico.

No obstante, el seguimiento se realizaba con mayor dificultad ya que la información de flujo óptico no se generaba en todas las iteraciones, sino cada tres o cuatro, debido a su elevado coste computacional. Esto supuso que los valores de salud asociada al movimiento descendieran levemente, teniendo que ajustar ciertos parámetros relacionados con la salud.

En cuanto al aprendizaje de color, éste era lo suficientemente bueno para que las personas permanecieran localizadas incluso al detenerse. Sin embargo, al no descartar explícitamente píxeles pertenecientes al fondo, los filtros de color generados eran más tolerantes de lo que se deseaba.

En conclusión, se comprobó que mediante la utilización de flujo óptico el algoritmo funcionaba con dificultades, entre las cuales destacaba la gran carga computacional añadida a la aplicación y que suponía el descenso de la velocidad del algoritmo a 3fps aproximadamente, lo que es incompensable.

5.7. Seguimiento de objetos pequeños

El objetivo de este experimento fue observar el funcionamiento del algoritmo para el seguimiento de objetos de menor tamaño que las personas, como por ejemplo pelotas del tamaño de balones de fútbol.

Hasta ahora, los parámetros del algoritmo han sido ajustados experimentalmente siempre realizando el seguimiento sobre personas. Al probar la aplicación sobre pelotas se comprobó que no funcionaba bien la localización ni el seguimiento de estos objetos. El problema era que no surgían razas, lo que significa que no aparecía ningún individuo prometedor dentro de la población de exploradores. Esto está directamente relacionado con los parámetros de salud del algoritmo.

Los objetos pequeños generan mucho menos movimiento que una persona, es decir, la cantidad de píxeles que supera el filtro de movimiento es pequeña. La salud de los individuos exploradores se asigna únicamente en base al movimiento, por lo que si esta salud es baja debido a la poca cantidad de píxeles, estos individuos no superan el umbral de salud mínima para crear una raza y comenzar el seguimiento.

Con el objeto de probar la versatilidad del algoritmo evolutivo de seguimiento 3D se realizó un reajuste de parámetros. Se disminuyó a un 30 % la salud mínima para considerar a un individuo explorador prometedor y la salud también mínima a un 15 % para que una raza no sea eliminada. De este modo, el algoritmo sí generaba razas para el seguimiento de las pelotas aunque éste se producía con mayor dificultad comparado con el seguimiento de personas.

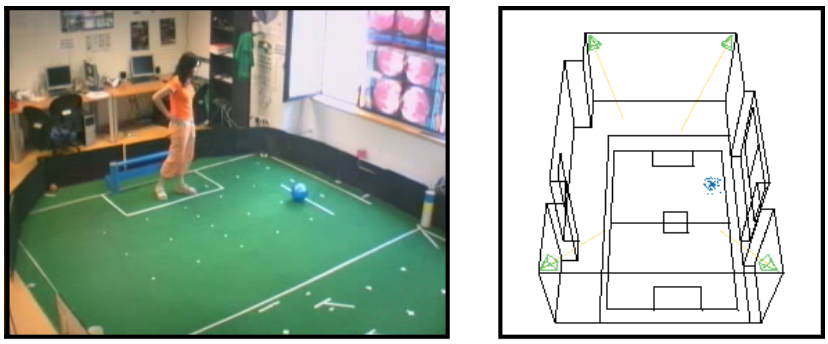

Figura 5.19: Seguimiento de objetos pequeños - Imagen real (a), representación 3D (b)

Probando el algoritmo ajustado para objetos pequeños sobre personas se observó que aparecían demasiadas poblaciones de explotadoras para cada persona, sobre todo a lo largo de las líneas de proyección entre la persona y las cámaras. Esto provocaba que la velocidad de la aplicación disminuyera y que la representación de la solución a la localización no se apreciara claramente.

Por tanto, se concluyó que deben ajustarse los parámetros de salud del algotimo según el tamaño de los objetos a seguir para su adecuado funcionamiento, concretamente el umbral de salud mínima para el nacimiento de una raza y el umbral de salud mínima para que una raza no pierda salud acumulada hasta desaparecer.

Capítulo 6
Conclusiones y trabajos futuros

En los capítulos anteriores se han mostrado los procedimientos utilizados para resolver el problema de la localización y el seguimiento 3D de personas. Además se expusieron los experimentos realizados más relevantes y los resultados obtenidos. En este capítulo se resumen las conclusiones obtenidas y las posibles vías de continuación para la investigación.

6.1. Conclusiones

El objetivo principal de este proyecto era realizar un seguimiento visual 3D de múltiples personas. Para conseguirlo se ha diseñado una aplicación que dentro de una habitación de gran volumen estima continuamente la posición 3D de las personas utilizando un algoritmo evolutivo multimodal.

En primer lugar es necesario destacar que los cuatro subobjetivos en los que se divide este proyecto se han logrado con éxito y los requisitos planteados se han satisfecho, tal y como hemos visto en los capítulos 4 y 5.

El primer subobjetivo consistía en la implementación de un algoritmo evolutivo multimodal que utiliza individuos puntuales como hipótesis. Este algoritmo usa la información de color y de movimiento que extrae de las imágenes recibidas a través de la plataforma *Jdec*, como se ha descrito en los apartados 4.6 y 4.7. Mediante esta información y haciendo uso de la biblioteca Progeo (ver 3.3.1), el sistema es capaz de realizar una estimación continuada de la posición 3D de las personas que se encuentren en la habitación. Además muestra el estado del seguimiento en la interfaz gráfica a través de una representación 3D de la habitación y el valor numérico de la posición estimada en cada momento.
El algoritmo asigna a cada individuo una salud basándose en la compatibilidad con la información sensorial obtenida de las imágenes (ver sección 4.4). Utilizando

una población de individuos exploradores la aplicación busca en 3D las fuentes de movimiento producido en la habitación. Al detectar movimiento y aparecer individuos con salud elevada se dispara el seguimiento de las personas, el cuál se realiza mediante otra población distinta de individuos, las razas explotadoras. Surgiendo habitualmente una raza por persona, se encargan de estimar sus posiciones 3D en cada momento y de obtener información de color sobre la vestimenta de las personas.
Para probar y ajustar el algoritmo de seguimiento primero se realizaron experimentos con color predefinido y resultaron satisfactorios (ver sección 5.4). De este modo se comprobó el correcto funcionamiento del algoritmo evaluándolo sin mezclarlo con el aprendizaje de color. La velocidad de la aplicación alcanzaba aproximadamente 14 ips utilizando las cuatro cámaras de la habitación, lo que supone suficiente vivacidad para el seguimiento de personas moviéndose naturalmente por la habitación.

El segundo subobjetivo era el autoaprendizaje del color característico de la vestimenta de las personas. Se añadió al algoritmo la capacidad de generar dinámicamente tantos filtros de color adecuados como personas se estuvieran siguiendo. Esto permite que la aplicación no necesite ningún tipo de información de color previa. Se probaron dos técnicas distintas: media RGB para caracterizar el color de cada persona e histograma de matices, descrito en el apartado 4.7.2. Gracias a la posibilidad de visualizar los filtros en la interfaz gráfica se pudo ajustar el aprendizaje y comprobar que el color filtrado era el adecuado. De este modo, aunque una persona se detenga, el sistema es capaz de continuar su seguimiento.
En cuanto al rendimiento de la aplicación, incluir un aprendizaje de color mediante histograma para cada persona supuso que la velocidad del algoritmo disminuyera a 12 ips. Sin embargo el algoritmo sigue siendo suficientemente vivaz para realizar el seguimiento de múltiples personas.

El tercer subobjetivo consistía en realizar una exploración y análisis de distintas técnicas para la detección de movimiento en las imágenes: diferencia RGB con el fotograma anterior, diferencia RGB con el fondo aprendido, combinación de las anteriores y flujo óptico (ver sección 5.6). La información de movimiento para este sistema es de gran importancia puesto que a partir de ella surge el seguimiento y también constituye la base para el aprendizaje de color y el cálculo de la salud.
La manera de extraer esta información repercute directamente en la calidad del seguimiento, por ello se plantearon distintas técnicas y se contrastaron con el fin de seleccionar la mejor para este problema. En el análisis se tomó en cuenta los valores

de salud de los individuos, la eficacia de los filtros de color generados y el coste computacional requerido.
La técnica finalmente escogida fue la combinación de dos distintas: diferencia respecto al fotograma anterior y diferencia respecto al fondo aprendido. Ésta no suponía una gran carga computacional y los valores de salud eran adecuados. Sin embargo, la decisión se tomó en gran medida por la notable mejora en el aprendizaje de color.

Evaluar la aplicación con todos los subsistemas integrados era la finalidad del cuarto subobjetivo. A través de los experimentos y las pruebas descritas en el capítulo 5, se ajustó el algoritmo de seguimiento así como el aprendizaje de color con el fin de que el sistema final cumpliera de la manera más fiel posible los objetivos propuestos. Se comprobó que el funcionamiento de la aplicación final es satisfactorio.
No obstante, la aplicación posee algunas limitaciones. Por ejemplo, no aprende colores claros ni oscuros, el número máximo de personas a seguir está limitado por cuestión de recursos y puede fusionar los seguimientos de personas que visten de color semejante y que se encuentren cerca espacialmente.

En cuanto a los requisitos, la aplicación cumple todos los planteados en el capítulo 2. La aplicación utiliza únicamente *hardware convencional*, funciona correctamente en una *habitación de gran volumen* y es capaz de seguir a varias personas de forma *vivaz y precisa*. Para la comprobación de este último requisito se realizaron pruebas de precisión tanto en la habitación real como con la herramienta de simulación, obteniendo errores siempre inferiores a 20cm (ver apartado 5.2).

Para concluir, conviene recordar que el punto de partida fue el proyecto fin de carrera *Aplicación de seguridad basada en visión* [Cabello, 2006], en el que el seguimiento se realizaba sobre un único objeto o persona y era necesario indicar a la aplicación el color específico a seguir. En este proyecto se ha avanzado consiguiendo la localización y el seguimiento de *múltiples* personas sin ofrecer información a priori de color a la aplicación.

6.2. Trabajos futuros

Este proyecto fin de carrera, como todo proyecto de ingeniería, presenta algunas limitaciones. Una línea inmediata de progreso es tratar de paliar esas limitaciones que se dan en el sistema actual. Por ejemplo, la limitación más significativa

se da en la autodefinición de color. En este sentido se podría diseñar un autoaprendizaje de color que no excluyera colores claros y oscuros y que fuera más discriminante a la hora de fusionar o no los seguimientos de personas vestidas de un color similar.

Por otra parte, una importante línea futura de investigación para la continuidad de este proyecto es la posibilidad de ampliar la zona vigilada. En este proyecto se ha resuelto con éxito el seguimiento tridimensional de múltiples personas dentro de una habitación de gran volumen utilizando cuatro cámaras. Sin embargo, el algoritmo empleado es escalable y permite aumentar el número de fuentes de información, por lo que todo parece indicar que introduciendo más cámaras se podría cubrir con éxito cualquier espacio que se proponga, como por ejemplo un entorno de oficinas con varios despachos y habitaciones. También se podría acoplar a las cámaras un cuello mecánico que permitiera su movimiento para cubrir mejor la zona vigilada.

Otra línea de investigación podría consistir en modificar la primitiva de información utilizada para el algoritmo de seguimiento, que en este proyecto es un punto 3D. Se podría dotar de volumen a los individuos con el fin de obtener información más completa con cada uno, lo que probablemente permitiría disminuir la cantidad de individuos utilizados y aumentar la precisión del sistema.

Bibliografía

[Barrera *et al.*, 2005] Pablo Barrera, José María Cañas, y Vicente Matellán. Visual object tracking in 3d with color based particle filter. *Int. Journal of Information Technology*, 2005.

[Barrera y Cañas, 2004] Pablo Barrera y José María Cañas. Seguimiento tridimensional usando dos cámaras. 2004.

[Bishop y Welch, 2005] Gary Bishop y Greg Welch. An introduction to the Kalman filter. page 16. 2005.

[Cabello, 2006] Antonio Pineda Cabello. Aplicación de seguridad basada en visión. *Proyecto Fin de Carrera, URJC*, 2006.

[Cadenas, 2007] José Antonio Santos Cadenas. Cálculo y aplicaciones del flujo óptico en tiempo real. *Proyecto Fin de Carrera, URJC*, 2007.

[D. Margaritis, 1998] S. Thrun D. Margaritis. Learning to locate an object in 3d space from a sequence of images. 1998.

[González, 2007] Pablo Barrera González. *Aplicación de los métodos secuenciales de Monte Carlo al seguimiento visual 3D de múltiples objetos.* PhD thesis, 2007.

[Louchet *et al.*, 2002] Jean Louchet, Maud Guyon, Marie-Jeanne Lesot, y Amine Boumaza. Dynamic flies: a new pattern recognition tool applied to stereo sequence processing. *Pattern recognition letters*, 2002.

[Louchet, 2000] Jean Louchet. Stereo analysis using individual evolution strategy. 2000.

[Louchet, 2001] Jean Louchet. Using an individual evolution strategy for stereovision. *Genetic Programming and Evolvable Machines*, 2001.

[Martínez, 2007] Iván García Martínez. Reconstrucción 3d visual con algoritmos evolutivos. *Proyecto Fin de Carrera, URJC*, 2007.

[Maybeck, 1979] Peter S. Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

[Plaza, 2003] José María Cañas Plaza. *Jerarquía Dinámica de Esquemas para la generación de comportamiento autónomo*. PhD thesis, Universidad Politécnica de Madrid, 2003.

[Plaza, 2004] José María Cañas Plaza. Manual de programación de robots con jde. *URJC*, 2004.

[Ángel Cortés Maya, 2007] Ángel Cortés Maya. Localización y construcción de mapas en un robot de interiores. *Proyecto Fin de Carrera, URJC*, 2007.