



**tsi**

**Universidad de Castilla-La Mancha**

**Escuela Superior de Informática**

**Departamento de Tecnologías y Sistemas de Información**  
**Programa Oficial de Postgrado en Tecnologías Informáticas Avanzadas**

**Trabajo Fin de Máster**

---

**Título:**

**Sistema de posicionamiento en Interiores multimodal  
utilizando Técnicas y Algoritmos de Realidad  
Aumentada**

**Julio de 2012**

**Alumno: César Mora Castro**

**Director: Dr. D. Carlos González Morcillo**

**Trabajo Fin de Máster**  
(Máster en Tecnologías Informáticas Avanzadas)

**Sistema de posicionamiento en interiores  
*multimodal* utilizando técnicas y algoritmos  
de Realidad Aumentada**

Julio de 2012

**Autor**

César Mora Castro  
(Ingeniero en Informática)

**Director**

Dr. Carlos González Morcillo



Universidad de  
Castilla-La Mancha



Escuela Superior  
de Informática

© César Mora Castro. Se permite la copia y la distribución de la totalidad o parte de este documento sin ánimo de lucro. Toda copia total o parcial deber citar expresamente el nombre del autor, de la Universidad de Castilla-La Mancha y deber incluir esta misma licencia, añadiendo, si es copia literal, la mención "Copia Literal".

Se autoriza la modificación y traducción de la obra sin ánimo de lucro siempre que se haga constar en la obra resultante de la modificación el nombre de la obra originaria, el autor de la obra originaria y el nombre de la Universidad de Castilla-La Mancha. La obra resultante también deber ser libremente reproducida, distribuida, comunicada al público y transformada en términos similares a los expresados en esta licencia.

Este documento fue maquetado con  $\text{\LaTeX}$ . Imágenes compuestas con *Gimp* y *LibreOffice*.

# Resumen

El uso de sistemas de posicionamiento a nivel mundial como el *GPS* ha sido uno de los avances más importantes realizados durante las últimas décadas, ya que permite realizar ciertas tareas relativas al posicionamiento en exteriores con gran facilidad. Por ejemplo, hoy en día la mayor parte de las familias cuentan con un dispositivo *GPS* que les permite viajar a cualquier parte del mundo de una forma muy sencilla.

Sin embargo, el mayor problema de estos sistemas es su *incapacidad* para funcionar en el interior de los edificios. Esta posibilidad abriría nuevas aplicaciones en el ámbito del turismo, grandes espacios públicos, etc. Actualmente existen multitud de propuestas para solucionar esta problemática, pero ninguna es lo suficientemente robusta y eficiente como para poder ser utilizada de forma masiva.

En este documento se presenta una solución basada en técnicas de *Realidad Aumentada* y *Visión Artificial* para el posicionamiento en interiores. De esta forma, cualquier dispositivo que cuente con una cámara de vídeo podrá realizar el posicionamiento en interiores. La propuesta integra varios algoritmos heterogéneos para realizar el posicionamiento, tales como la detección de marcadores, de puntos de interés, o descriptores de imagen. A continuación, un módulo debe poder realizar la *fusión* de las percepciones para obtener una única posición lo más fiable posible. Este sistema se engloba dentro del desarrollo del proyecto *ELCANO: MULTIMODAL INDOOR NAVIGATION INFRASTRUCTURE FOR DISABLED PEOPLE* (enmarcado en la Cátedra Indra-UCLM), y se centra en la asistencia a la movilidad de personas con necesidades especiales dentro de grandes edificios públicos, proporcionando indicaciones y la ayuda necesaria para poder realizar tareas de su ámbito laboral de forma más sencilla.

La arquitectura implementada y los resultados obtenidos se describen en el presente documento.



# Abstract

One of the most important achievements during the last few decades is the development of the world-wide positioning systems like *GPS*, which allow us to perform tasks requiring outdoor positioning with much more ease. For instance, nowadays almost each family owns a *GPS* device to move easily to any place in the world.

However, the major inconvenient of these system is that they are not able to work within buildings. That possibility would offer lots of new applications in the fields of tourism, large public facilities, etc. Currently there are several approaches to this problem, but none is as robust and efficient to be used in large scale.

In this document we present an approach based on Augmented Reality and Artificial Vision techniques for indoor positioning. Thus, any device equipped with a video camera could perform the indoor positioning. This approach gathers a number of algorithms to do the positioning, like marker detection, feature points detection, or visual descriptors. Then, a module fuses the perceptions from the algorithms to get a unique perception as reliable as possible. This system is framed into the development of the project *ELCANO: MULTI-MODAL INDOOR NAVIGATION INFRASTRUCTURE FOR DISABLED PEOPLE* (by the Indra-UCLM agreement), and it is focused on assistance for disabled people within large public facilities, giving them instructions and help to perform their work tasks easily.

The suggested architecture and its results are described in this document.

# Índice general

<b>Índice de figuras</b>	<b>1</b>
<b>1. Introducción</b>	<b>2</b>
1.1. Planteamiento general y problemática . . . . .	2
1.2. Objetivo del Trabajo de Fin de Máster . . . . .	3
1.3. Marco de trabajo: el Proyecto ELCANO . . . . .	3
1.4. Estructura del documento . . . . .	5
<b>2. Estado del arte</b>	<b>6</b>
2.1. Posicionamiento en interiores . . . . .	6
2.2. Realidad Aumentada . . . . .	8
2.3. Fusión de información . . . . .	13
<b>3. Descripción de la propuesta</b>	<b>21</b>
3.1. Descripción general . . . . .	21
3.2. Métodos de posicionamiento . . . . .	24
3.3. Módulo de refinamiento . . . . .	31
3.4. Módulo de fusión . . . . .	34
<b>4. Conclusiones y trabajo futuro</b>	<b>44</b>
<b>A. Asignaturas cursadas</b>	<b>47</b>
<b>B. Currículum</b>	<b>53</b>
<b>Bibliografía</b>	<b>56</b>

# Índice de figuras

1.1. Logotipo del proyecto ELCANO . . . . .	4
2.1. Caso de uso de posicionamiento <i>Wi-Fi</i> dentro de un edificio . . . . .	7
2.2. Jerarquía de la <i>Realidad Mixta</i> . . . . .	9
2.3. Elementos básicos de un sistema de <i>Realidad Aumentada</i> (izquierda) y tablet PC utilizado en este proyecto (derecha). . . . .	11
2.4. Métodos visuales de detección de patrones (izquierda) y de puntos de interés (derecha) . . . . .	12
2.5. Ejemplo de una Red Bayesiana. . . . .	14
2.6. Marco de trabajo del sistema <i>multimodal</i> basado en Realidad Aumentada . . .	15
2.7. Plano del edificio donde se aplica el sistema de posicionamiento móvil . . . . .	16
2.8. Ejemplo de escenario de interacción entre dos personas. . . . .	17
2.9. Ejemplo de conjunto difuso representante de la variable <i>altura</i> . . . . .	18
2.10 Diagrama de bloques del sistema difuso de posicionamiento de Ojeda et al [29]	20
3.1. Arquitectura del <i>Sistema multimodal de Posicionamiento</i> . . . . .	23
3.2. Diagrama de herencia de los métodos de <i>tracking</i> . . . . .	25
3.3. Arquitectura general de <i>FuzzyWeight</i> . . . . .	35
3.4. Variables <i>fuzzificadas</i> (de izquierda a derecha, <i>Optical Flow</i> , <i>PTAM</i> y <i>Peso</i> ). . .	36
3.5. <i>Screenshot</i> del primer vídeo . . . . .	38
3.6. Resultados del escenario 1 . . . . .	39
3.7. <i>Screenshot</i> del segundo vídeo . . . . .	40
3.8. Resultados del escenario 2 . . . . .	41
3.9. <i>Screenshot</i> del tercer vídeo . . . . .	42
3.10 Resultados del escenario 3 . . . . .	43

---

# 1

## Introducción

---

### 1.1. Planteamiento general y problemática

La capacidad de posicionamiento de usuarios a través de diversos dispositivos y técnicas presentan opciones muy interesantes a la hora de ofrecer servicios o resolver problemas como indicar cuál es la ruta más corta hacia un determinado destino, o proporcionar ayuda contextual.

En este campo, el sistema de posicionamiento *GPS*<sup>1</sup> permite localizar a un usuario en cualquier parte del mundo con un margen de error de unos pocos metros [11][26]. Se comenzó su desarrollo en el año 1972, y actualmente es mantenido por el gobierno de los Estados Unidos, siendo su acceso libre a cualquiera que posea un receptor. Tiene una gran importancia tanto para fines militares como civiles. Gracias a su alta precisión (del orden de unos pocos metros de error), se han podido desarrollar aplicaciones multitud de aplicaciones como los navegadores que son capaces de indicar la ruta hacia cualquier parte del mundo desde cualquier posición.

Su funcionamiento se basa en la recepción de la señal de varios satélites geoestacionarios colocados en órbita. Estos satélites mandan continuamente señales indicando la posición exacta del satélite y el momento en que se envió. De esta forma, mediante el uso de la información de varios satélites y aplicando algoritmos triangulación, el receptor puede estimar su posición con una gran precisión.

Sin embargo, el mayor problema del sistema *GPS* es la necesidad de tener siempre una buena recepción de dichos satélites. Por este mismo motivo, *GPS no funciona dentro de los edificios*, al recibirse la señal muy débil, por lo que es considerado un sistema de posicionamiento *outdoor*.

Aunque se han realizado algoritmos más complejos que intentan funcionar con una menor intensidad de señal dentro de edificios, como el llevado a cabo por Dedes y Dempster [4],

---

<sup>1</sup>Global Positioning System

u otros estudios que utilizan *pseudo-GPS* como el de la Universidad Nacional de Seoul [12], no han conseguido alcanzar un nivel de calidad mínimo para considerarlo un sistema de uso comercial.

Por lo tanto, se ha identificado la necesidad de construir un sistema de posicionamiento *indoor* independiente de *GPS* que sea lo suficientemente robusto, versátil, preciso y eficiente como para poder ser utilizado en dispositivos actuales a gran escala. Además, este sistema puede abrir nuevos campos de aplicación a los que el sistema *GPS* no llega.

A continuación, se describe el objetivo de investigación concreto de la presente Tesis de Fin de Máster.

## 1.2. Objetivo del Trabajo de Fin de Máster

El objetivo del presente Trabajo Fin de Máster es el desarrollo de un sistema de posicionamiento *indoor multimodal* basado en el uso de técnicas de Realidad Aumentada y Visión Artificial.

Este sistema de posicionamiento se basará únicamente en la información recibida por un dispositivo de vídeo (por ejemplo, una *webcam*), que capture información de su entorno inmediato, y aplique distintos algoritmos de visión artificial y realidad aumentada para obtener la localización del usuario. De esta forma, se evita la problemática de obtener la recepción de la señal de los satélites del caso de *GPS*. Además, puesto que uno de los principales requisitos es contar con un único dispositivo de vídeo, permite que gran variedad de dispositivos móviles actuales puedan implementarlo.

Por otro lado, este sistema será *multimodal*, es decir, podrá ejecutar varias técnicas de posicionamiento independientes de forma simultánea, para luego utilizar la información proporcionada por cada una de ellas y obtener un resultado más fiable.

Este sistema se ha enmarcado dentro del desarrollo del sistema ELCANO, que se centra en la aplicación de dicha tecnología para la asistencia a la movilidad de personas con necesidades especiales dentro de edificios, y poder indicarles de forma rápida e intuitiva dónde se encuentran los servicios y lugares que deseen. En la siguiente sección se detalla las características de este proyecto.

## 1.3. Marco de trabajo: el Proyecto ELCANO

El proyecto ELCANO surge como iniciativa de los laboratorios investigación Oreto y Arco, de la Escuela Superior de Informática (Campus de Ciudad Real, Universidad de Castilla-La Mancha), y con financiación de la Catedra Indra-UCLM<sup>2</sup>. En la Imagen 1.1 se muestra el logotipo del proyecto.

<sup>2</sup><http://catedraindra.uclm.es/elcano>



Figura 1.1: Logotipo del proyecto ELCANO

El objetivo principal consiste en el desarrollo de un sistema de posicionamiento en interiores para la asistencia a personas con necesidades especiales dentro de grandes edificios. De esta forma, el sistema podrá adaptarse al tipo de necesidad que tenga un determinado usuario, y podrá guiarle para llevar a cabo su trabajo dentro del edificio de la forma más rápida y cómoda posible.

Además, el sistema podrá ofrecer asistencia técnica mediante videoconferencia en cualquier momento, indicar mediante indicaciones virtuales los puntos de interés del edificio, o incluso escuchar de forma sonora las indicaciones para alcanzar de forma satisfactoria un determinado destino.

El usuario contará con un *Dispositivo Personal de Navegación* que le presentará la información adaptada según su discapacidad, mientras que el *Sistema de multimodal de Posicionamiento* incorporado en el dispositivo calculará su posición. Esta posición es transmitida a un *Sistema de Información Centralizado* que proporcionará los distintos servicios a los usuarios que estén registrados en el sistema.

Como se ha indicado más anteriormente, el desarrollo del proyecto se realiza de forma conjunta por dos laboratorios, que llevan a cabo distintas partes del proyecto:

- **Laboratorio Arco:** se ocupa principalmente del desarrollo del *Sistema de Información Centralizado*. Este sistema gestiona el registro de los usuarios que se encuentren en el edificio, les ofrece asistencia técnica, las diferentes opciones de tareas o servicios que pueden realizar dentro del edificio, y monitoriza su posicionamiento en todo momento. Además, se ocupa del desarrollo del sistema de posicionamiento basado en redes *Wifi*, basado en la detección de los puntos de conexión del edificio.
- **Laboratorio Oreto:** su principal cometido es el desarrollo del sistema de posicionamiento *multimodal* basado en técnicas de realidad aumentada y visión artificial incorporado en el *Dispositivo Personal de Navegación*. Este sistema utiliza técnicas basadas en detección de marcadores, detección de puntos de interés o incluso *optical flow*. Una parte importante de la investigación es la forma de integrar los distintos métodos para conseguir una única percepción de localización lo más fiable posible. **Es en esta parte donde recae la labor investigadora de este Trabajo de Fin de Máster.**

Por otro lado, se encarga del desarrollo de la interfaz gráfica del *Dispositivo Personal de Navegación*, ofreciendo las distintas tareas programadas por el *Sistema de Información Centralizado*, la posición actual del usuario, la asistencia por videoconferencia, el cambio de rol según las necesidades del usuario y la reproducción de las indicaciones auditivas.

Una vez conocido el marco de trabajo en el que se desarrolla la labor de investigación del presente documento, y habiendo identificado cuál es el principal objetivo que se quiere alcanzar, se va a detallar en la siguiente sección la estructura de este documento.

## 1.4. Estructura del documento

El presente documento se estructura de la siguiente forma:

- **Capítulo 2:** presenta una visión general del estado de las disciplinas y tecnologías que ha sido necesario investigar para el desarrollo de este trabajo.  
Dentro de dichas disciplinas se cuenta el posicionamiento de dispositivos, diferentes técnicas de Realidad Aumentada y Visión Artificial como la detección de marcadores, puntos de interés y *optical flow*, o el uso de la lógica difusa para la fusión de información.
- **Capítulo 3:** describe la arquitectura propuesta para el desarrollo del sistema *multimodal* de posicionamiento *indoor*.  
Las partes principales que se describen de la solución propuesta son los métodos de *tracking* implementados, la arquitectura que permite que se ejecuten simultáneamente, y los diferentes módulos utilizados para la fusión de las percepciones, que incluyen el módulo heurístico y el basado en lógica difusa. También se muestran resultados obtenidos tras realizar pruebas de integración de los diferentes módulos
- **Capítulo 4:** muestra un apartado de conclusiones evaluando el aprovechamiento de la investigación y las líneas de trabajo futuro propuestas.
- **Anexo A:** resumen de todas las asignaturas cursadas por el alumno durante el máster, junto a una conclusión personal sobre qué han aportado cada una a la labor investigadora y a la experiencia del alumno.
- **Anexo B:** *Curriculum Vitae* en español actualizado a Julio de 2012 del alumno César Mora Castro.

---

# 2

## Estado del arte

---

El objetivo del trabajo es realizar un sistema de posicionamiento en interiores basado en técnicas de realidad aumentada para asistencia a personas con necesidades especiales para realizar tareas del ámbito de su trabajo dentro de grandes edificios públicos.

Por lo tanto, se hace necesario conocer las técnicas y resultados que ofrecen sistemas anteriores realizados en relación con el posicionamiento en interiores (Sección 2.1), haciendo especial hincapié en la tecnología base utilizada.

Al basarse la propuesta de este documento en técnicas de Realidad Aumentada, se presenta su funcionamiento y los diferentes métodos de posicionamiento existentes relacionados con esta tecnología (Sección 2.2).

Finalmente, puesto que la arquitectura de esta solución es *multimodal*, es decir, soporta distintas técnicas funcionando simultáneamente, también es necesario conocer qué métodos y modelos existen para realizar fusión de información (Sección 2.3).

### 2.1. Posicionamiento en interiores

Los sistemas de posicionamiento hoy en día tienen una gran diversidad de aplicaciones y son utilizados por un gran número de usuarios alrededor del mundo. El ejemplo más claro es el sistema *GPS*, capaz de calcular la localización de un usuario con muy pocos metros de error.

Sin embargo, la mayor problemática que presenta este sistema, es la imposibilidad de funcionar dentro de edificios, debido a que la señal recibida por los satélites no es lo suficientemente intensa. Por ello, muchos investigadores han propuesto diversas soluciones a este posicionamiento *indoor*. A continuación se presentan las más importantes:



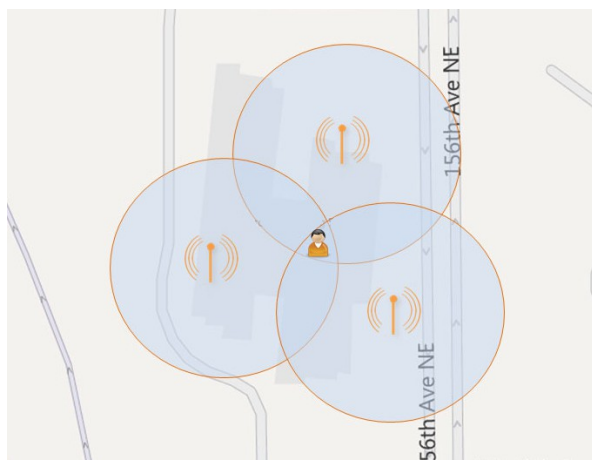


Figura 2.1: Caso de uso de posicionamiento *Wi-Fi* dentro de un edificio

### Posicionamiento mediante *Wi-Fi*

El posicionamiento *Wi-Fi* ha sido el más utilizado en los últimos años. Su funcionamiento se basa en la utilización de los puntos de conexión *Wi-Fi* existentes en la gran mayoría de los edificios actuales para proporcionar conexión a internet. Conociendo la posición y características de cada punto de conexión, y pudiendo identificarlos, es posible determinar la posición del usuario. En la Figura 2.1 se puede ver el esquema de un caso de uso típico de posicionamiento *Wi-Fi*.

El mayor inconveniente de este método es la gran variabilidad posible de la intensidad de la señal *Wi-Fi* dentro del edificio, que es afectada por multitud de factores: climatología, objetos físicos, interferencias por otras señales, número de personas en el interior, etc. Por lo tanto, el margen de error es lo suficientemente grande como para que el posicionamiento no sea posible para aplicaciones de Realidad Aumentada.

Algunos trabajos anteriores sobre este tipo de sistemas son el de Liu et al [24], o el sistema híbrido de Evvenou et al [5], que utiliza Redes Bayesianas junto a la detección de la señal *Wi-Fi* para inferir la posición del usuario. Este tipo de modelos probabilísticos de inferencia son muy útiles en entornos en los que no se puede obtener un modelo matemático preciso. En el Apartado 2.3 se detalla más en profundidad el uso de estas Redes para la obtención de la información de posicionamiento a partir de diversas fuentes, y otras aplicaciones de Realidad Aumentada.

En el proyecto ELCANO se ha incorporado un método de posicionamiento *Wi-Fi*, implementado por el laboratorio de investigación Arco. Este sistema de *tracking* no es de naturaleza visual, por lo que queda fuera del alcance de este documento.

### Posicionamiento mediante radiofrecuencia y ultrasonidos

Estos sistemas son los precursores de los sistemas de posicionamiento mediante *Wi-Fi*. Consisten en colocar transmisores de radiofrecuencia repartidos por el edificio, y realizar el mismo tipo de cálculo que en el apartado anterior para obtener la posición del usuario.

Como ventaja cuenta con que se pueden escoger las características de los transmisores de radiofrecuencia, y la intensidad esta sometida a menor variación dado que no se usa el transmisor para otro tipo de aplicaciones, al contrario que sucede con los puntos de conexión *Wi-Fi*. Además, el sistema es de un relativo bajo coste, y el número de transmisores instalados es mayor, por lo que el posicionamiento tiene mejor calidad.

Los mayores inconvenientes son precisamente que es necesaria una instalación a lo largo del edificio, mientras que en el posicionamiento *Wi-Fi* se aprovecha la infraestructura de conexión existente. Incluso aunque el coste de instalación del sistema es bajo, el de un sistema basado en *Wi-Fi* es prácticamente nulo, dado que no es necesaria infraestructura adicional. Un sistema de este tipo fue desarrollado por Randell et al [34].

### Posicionamiento mediante Bluetooth

Estos sistemas de posicionamiento *indoor* también han sido implementados utilizando tecnología *bluetooth*, que aunque no es tan usada en edificios como la *Wi-Fi*, también se puede aprovechar cualquier tipo de infraestructura existente.

Feldmann et al [6] han desarrollado un sistema utilizando este tipo de tecnología, indicando en el artículo el diseño, el funcionamiento y los resultados experimentales realizados a cabo.

### Problemática

Todos estos sistemas de posicionamiento en interiores son capaces de localizar al usuario dentro de un edificio con mayor o menor precisión, pero no son capaces de dar su orientación. Por lo tanto, estos sistemas sirven para *seguir el rastro* a los usuarios, pero no dan la precisión deseada como para desarrollar un sistema de Realidad Aumentada con él, ya que la integración que requieren los objetos virtuales necesita de un posicionamiento mucho más exacto.

El *Sistema multimodal de Posicionamiento* sugerido en este documento utilizando técnicas de Realidad Aumentada pretende alcanzar este nivel de precisión, basándose en conceptos totalmente diferentes.

## 2.2. Realidad Aumentada

La *Realidad Aumentada* es, según Ronald T. Azuma [1], una *variación de los entornos virtuales* en la cual el usuario percibe el mundo real con objetos virtuales integrados a través de algún tipo de dispositivo, siendo esta integración lo suficientemente precisa como para que la sensación de pertenencia de los objetos virtuales al mundo real sea total.



Figura 2.2: Jerarquía de la *Realidad Mixta*

Según Azuma, para que una aplicación pueda ser considerada de *Realidad Aumentada* deben cumplirse tres características:

1. Debe *combinar* el mundo real y el virtual. Es decir, el resultado debe ser una imagen con el mundo real y el virtual integrado.
2. Debe ser *interactivo* y en *tiempo real*. Los algoritmos de integración y combinación deben ser lo suficientemente eficientes y optimizados como para que la integración sea “instantánea”, sin que necesite ningún tipo de post-procesado. De esta manera, las películas de ciencia ficción no se consideran Realidad Aumentada porque no cumplen con esta condición.
3. La integración debe producirse en el espacio tridimensional. Existen aplicaciones que superponen imágenes en dos dimensiones, como es el caso de *Layar*, pero la integración no se considera completa, y por lo tanto, no es Realidad Aumentada.

El paradigma general por el cual se mezcla el mundo real y el virtual se denomina *Realidad Mixta* [20]. En la Figura 2.2 se muestra la jerarquía en la que se clasifica este tipo de paradigma. A la izquierda, se muestra el caso extremo de *integración nula*, en el que la interacción se realiza únicamente a través de los objetos reales. No existe ningún tipo de dispositivo informático con el cual realizar la interacción.

El siguiente paradigma es el de *Realidad Aumentada*. Este paradigma está más cerca del extremo del *entorno real* que del *virtual* porque se parte del *mundo real* como base, y a partir de él se añaden los objetos virtuales. Este paradigma contempla multitud de ventajas y aplicaciones. Henderson y Feiner llevaron a cabo un estudio evaluando las ventajas de la Realidad Aumentada frente a métodos convencionales en tareas específicas como la realización de procesos de mantenimiento [10]. En el Apartado 2.2 se detallan más aplicaciones de esta tecnología.

Acercándose más hacia el *entorno virtual*, se encuentra el paradigma de la *Virtualidad Aumentada* [27]. En un sistema que utilice esta tecnología, se parte de un sistema completamente virtual, que puede ser compartido por varios usuarios, y se interactúa con él a través del mundo real. Algunos proyectos que han utilizado este paradigma se enfocaban como *sistemas colaborativos*, en los que se crean un punto de encuentro virtual (una sala de reuniones, por ejemplo). Algunos de estos proyectos para colaboración son el de Regenbrecht et al. para colaboración [35] y videoconferencia remota [36].

Por último, se encuentran los sistemas de *Realidad Virtual* [39], en los que todo es generado por ordenador, y el objetivo es que el usuario se abstraiga lo máximo posible del mundo real, para conseguir una inmersión total en el mundo virtual. Estos sistemas tuvieron un gran auge hace unos años, en su modalidad con *HMD*<sup>1</sup>, aunque no llegaron a consolidarse completamente. Un proyecto interesante realizado sobre *Realidad Virtual* es *CAVE* de Cruz-Neira et al [3]. También se realizaron multitud de estudios que probaban la eficacia de este paradigma para el aprendizaje como el de Seymour et al [38].

Una vez que se tiene una visión general de los paradigmas existentes que mezclan mundo real y virtual (*Realidad Mixta*), se va a explicar en detalle las características y ventajas de la *Realidad Aumentada*, que es el utilizado en el *Sistema multimodal de Posicionamiento* de este documento.

## Elementos estructurales

Un sistema que haga uso de la Realidad Aumentada debe poseer tres elementos clave:

- **Dispositivo de captura de vídeo:** dado que la base de la Realidad Aumentada consiste en *mezclar* el mundo real y el virtual, es necesario obtener esa información del entorno que rodea al usuario con un dispositivo de vídeo (ver Figura 2.3. Estos dispositivos pueden ser desde una *webcam*, la cámara integrada de un móvil, o una cámara de alta resolución. Además, se aprovecha la información visual para realizar el *tracking* y estimar la posición del usuario en el mundo real y poder realizar la integración. De esta etapa se hablará en la Sección 2.2.
- **Dispositivo de procesamiento:** Para realizar la integración del mundo real y virtual, es necesario un dispositivo de procesamiento que, además de generar los gráficos tridimensionales y otro tipo de servicios que ofrezca la aplicación (audio, *video streaming*, etc), realice la propia integración de los dos mundos (ver Figura 2.3). Para ello es necesario calcular la posición del usuario dentro del mundo real, y poder así adaptar los objetos virtuales integrados en la imagen del mundo real. Este cálculo de posicionamiento se conoce como *tracking*. Existen diversos métodos de *tracking*, basados en diferentes técnicas. En el Apartado 2.2 se realiza un repaso sobre los diversos métodos existentes.
- **Dispositivo de visualización:** por último, esta información, una vez integrada, debe ser mostrada con algún tipo de dispositivo de visualización al usuario (ver Figura 2.3. Estos dispositivos pueden ser muy variados, desde una pantalla de un ordenador de sobremesa o un portátil, hasta un *HMD* (Un dispositivo montado en la cabeza del usuario con unas gafas que cuentan con dos *displays*), o incluso un proyector.

En [32] se puede ver una propuesta de un juego que utiliza la Realidad Aumentada con un *HMD*, y en [16] un sistema conferencias con el mismo esquema. Incluso la Realidad Aumentada sobre proyectores tiene su propia línea de investigación, como demuestran los estudios de Vogele [40].

---

<sup>1</sup>*Head Mounted Display*



Figura 2.3: Elementos básicos de un sistema de *Realidad Aumentada* (izquierda) y tablet PC utilizado en este proyecto (derecha).

En el proyecto ELCANO, y con fines de esta investigación, se ha utilizado un *tablet PC* que incorpora los tres elementos integrados: una webcam, capacidad de procesamiento suficiente para realizar el *tracking*, y dispone de una pantalla táctil para visualizar la aplicación e interactuar con ella. Además, proporciona suficiente movilidad al usuario como para desplazarse por el entorno libremente, la cual es una característica muy importante para un *Sistema multimodal de Posicionamiento* en interiores.

Como se ha visto, uno de los aspectos más importantes y críticos de una aplicación de Realidad Aumentada es la problemática del *tracking*. La eficiencia y precisión de este método determinará la calidad y las condiciones en las que la aplicación podrá funcionar.

En el siguiente apartado se muestra una revisión de los diferentes métodos de *tracking* existentes en la actualidad, analizando ventajas e inconvenientes de cada una.

### Métodos de *tracking*

A continuación, se muestra una clasificación y las características que diferencian a cada método de *tracking*. Estos métodos se pueden dividir principalmente en dos grandes grupos:

- **Métodos visuales:** basan su fuente de información en la proporcionada por dispositivos de vídeos, utilizan los *frames* de una cámara para estimar la posición del usuario. El tratamiento que se hace a estos *frames* pueden ser, a su vez, de distintos tipos:
  - *Detección de patrones:* estos métodos *analizan* la imagen en busca de patrones (ver Figura 2.4 izquierda) previamente entrenados. Estos patrones suelen tener unas características concretas que permitan al usuario localizarse dentro de la escena. Entre las bibliotecas más importantes de detección de patrones que se pueden encontrar se destaca: *ARToolKit* [17], que utiliza marcadores cuadrados en blanco y negro (o algunas de sus variantes, como *FLARToolKit*).
  - *Detección de puntos de interés:* este tipo de algoritmos no necesita buscar un determinado tipo de patrón, sino que utiliza los denominados *puntos de interés* (ver Figura 2.4 derecha). Estos puntos son zonas características de la escena, donde puede haber un gran cambio de contraste o de color, y que por lo tanto es fácil de identificar. A partir de la identificación de estos puntos en dos o más *frames*, es posible determinar la posición del usuario utilizando distintos algoritmos.

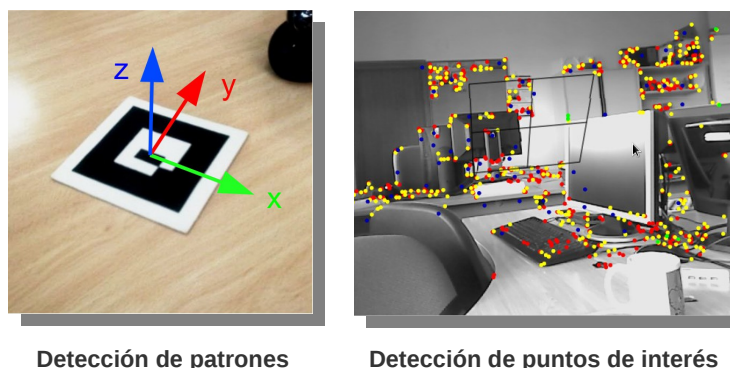


Figura 2.4: Métodos visuales de detección de patrones (*izquierda*) y de puntos de interés (*derecha*)

Algunas bibliotecas que utilizan detección de puntos de interés son: *BazAR* [2], que usa imágenes pre-entrenadas que luego intentará identificar en el vídeo, o el algoritmo *PTAM* [21], que es capaz de detectar, sin ningún tipo de entrenamiento, puntos de interés en la escena. El algoritmo, además, mejora su identificación en tiempo de ejecución. Existen distintas versiones que intentan mejorar este algoritmo, como *PTAMM-Plus* [28], que implementan soporte para entrenamiento y carga de diferentes mapas simultáneos.

- **Métodos no visuales:** existen otras múltiples formas de calcular la posición del usuario que no están basadas en información visual, sino en dispositivos o hardware externo, como acelerómetros o giróscopos. Estos sistemas no suelen ser muy precisos, ya que el error cometido al hacer el cálculo es *acumulativo*.

Algunos estudios que han utilizado este tipo de métodos para aplicaciones de Realidad Aumentada son los de Lang et al [23], o el sistema híbrido de Jiang et al [14].

En el proyecto ELCANO se utilizan actualmente cinco tipos de métodos de *tracking* visuales, que se explicarán con más nivel de detalle en el Capítulo de 3.

Una vez que se ha dado una visión más profunda sobre qué es la Realidad Aumentada, y cuáles son sus elementos principales, se va a pasar a analizar sus numerosos campos de aplicación.

## Aplicaciones

La Realidad Aumentada es un paradigma de interacción con los sistemas de información más natural e intuitivo que el clásico interfaz de manipulación directa basada en teclado y ratón. Esto hace que muchas tareas, como el aprendizaje o el entrenamiento sean más amenas y efectivas con estos sistemas. A continuación se presentan algunas de las aplicaciones más importantes de la Realidad Aumentada:

- **Ocio:** en la industria del ocio supone un paradigma muy novedoso y atractivo para el desarrollo de videojuegos. *EyePet* es un juego comercializado por Sony para su plataforma Playstation 3, cuyo objetivo principal es criar una mascota inmersa en el mundo real mediante el uso de una marca. *Invizimals* es otro ejemplo de éxito para la misma plataforma desarrollado por la empresa española Novarama Technology S.L.
- **Seguridad:** una opción interesante es aplicar la Realidad Aumentada para complementar la información relacionada con la seguridad que se obtiene de un edificio o recinto (vídeo de cámaras remotas o *logs* diversos). En el proyecto HESPERIA, se utilizaba la Realidad Aumentada como tecnología para mejorar la experiencia en labores de seguridad. En la universidad de Hokkaido [19] se realizaron igualmente investigaciones en esta materia. Estas investigaciones se centraron en aplicar la Realidad Aumentada a la monitorización en tiempo real de las instalaciones de un edificio.
- **Docencia:** la Realidad Aumentada puede ayudar durante el aprendizaje, haciéndolo de forma más amena e interactiva. También puede utilizarse para facilitar el entrenamiento de personal de mantenimiento en automóviles, fábricas, etc. En [18] se describe un sistema para la enseñanza de matemáticas ayudándose de la Realidad Aumentada, gracias a la cual se representan funciones y geometría en el espacio 3D.
- **Mantenimiento y reparación:** los mecánicos y operadores de maquinaria compleja pueden utilizar la Realidad Aumentada para llevar a cabo tareas de mantenimiento. El trabajador utilizaría un sistema en el que se superpondría información visual que indicase las instrucciones para realizar la tarea. *BMW* ha invertido en los últimos años en esta tecnología, siendo un ejemplo de empresa que emplea la Realidad Aumentada para mejorar la eficiencia de sus mecánicos.
- **Medicina:** en trabajos que requieren una precisión y un cuidado tan delicado como puede ser una operación quirúrgica, la Realidad Aumentada puede añadir información adicional como la ubicación exacta para realizar el corte o las constantes vitales del paciente. Actualmente ya se han llevado a cabo investigaciones como las de la universidad de Carolina del Norte [37].

## 2.3. Fusión de información

En el mundo real, muchos sucesos no tienen representación posible con un modelo matemático preciso, o son demasiado complejos como para ofrecerse a ello. Por esto mismo, se han recurrido a distintas técnicas para modelar, de forma probabilística o difusa, estos sucesos. Estos modelos, como las *Redes Bayesianas* o la *Lógica Difusa* dan resultados muy aproximados a sucesos reales, a pesar de que el modelo que se utiliza no es exacto.

En el *Sistema multimodal de Posicionamiento* que se propone en este trabajo, es importante inferir o fusionar la información que se tiene en un instante preciso sobre distintos métodos de posicionamiento en uno solo. A esto hay que añadir que cada método de posicionamiento tiene un error que puede variar en cada instante, por lo que hay que tener en cuenta distintos indicadores y métodos para saber cuál es el *valor de calidad* de cada método, y poder inferir una posición final lo más precisa posible.

Para el proyecto ELCANO se ha contemplado utilizar estas dos herramientas, de las cuáles se hace una revisión a continuación de cómo funcionan y en qué se han aplicado.

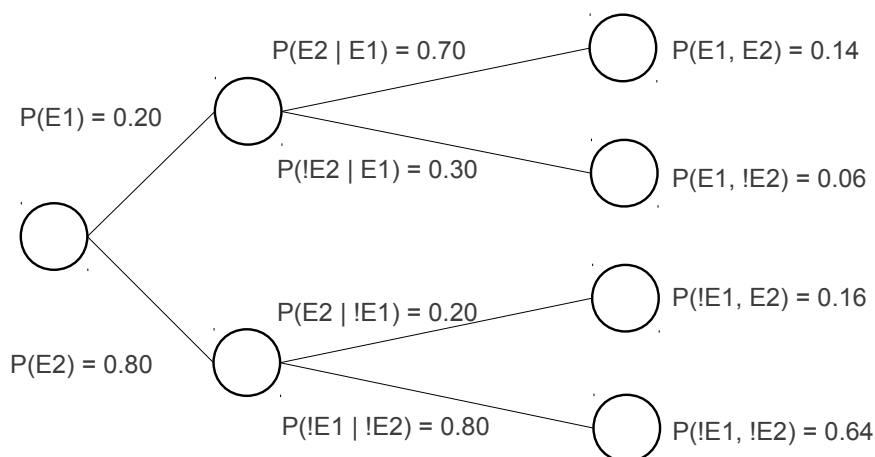


Figura 2.5: Ejemplo de una Red Bayesiana.

### Fusión basada en Redes Bayesianas

Las Redes Bayesianas [13][7], o también conocidas como *redes de creencia*, son un modelo probabilístico de representación de la información que permite la inferencia de conocimiento, dadas un conjunto de evidencias. En la Figura 2.5 se muestra un ejemplo de Red Bayesiana.

Desde el punto de vista matemático, las Redes Bayesianas son *grafos acíclicos dirigidos*, en los que cada nodo representa una variable, que suelen representar sucesos (que llueva, que suene una alarma, etc), y los arcos representan las relaciones de dependencia que existen entre dichos sucesos.

Si existe un arco entre un nodo  $N_1$  y un nodo  $N_2$ , siendo  $N_1$  el padre de  $N_2$ , quiere decir que la probabilidad de que  $N_2$  suceda depende de si  $N_1$  ha sucedido, pero al contrario no es cierto,  $N_2$  no interfiere en la probabilidad de que  $N_1$  suceda.

Existen algoritmos para inferir la probabilidad de que un nodo (o suceso) ocurra, dando como información si el resto de sucesos ha sucedido, no ha sucedido, o no se tiene constancia.

Las aplicaciones reales más importantes de estas redes son:

- Determinar ciertos comportamientos sociales.
- Prevención de enfermedades.
- Clasificación de cantidades grandes de datos.

A continuación, se presenta un estudio sobre algunos de los proyectos más notables que han utilizado las Redes Bayesianas para la fusión de información en sistemas que utilizan Realidad Aumentada como tecnología de interacción.



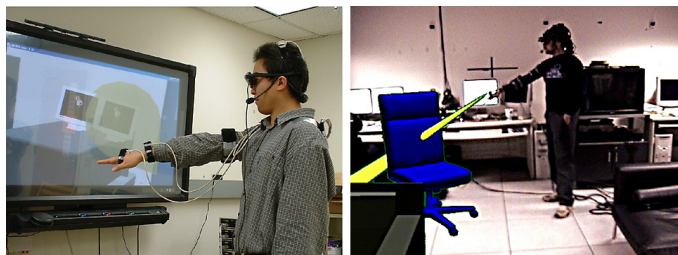


Figura 2.6: Marco de trabajo del sistema *multimodal* basado en Realidad Aumentada

### Fusión de métodos de un sistema *multimodal*

En el artículo *Mutual disambiguation of 3D multimodal interaction in augmented and virtual reality* de Kaiser et al [15], se describe un sistema de posicionamiento en un sistema inmersivo, ya sea basado en Realidad Aumentada o en Realidad Virtual, donde se hace necesario disponer de un sistema de fusión de las diferentes fuentes de información simbólicas o estadísticas, dada la naturaleza *incierta* de dichas fuentes.

Algunas de estas fuentes de información son proporcionadas por distintos agentes que controlan la posición de objetos como marcadores, o incluso la posición de los brazos de los usuarios. Esta aproximación es una solución general, que puede ser aplicada a la fusión específica que trata el tema de investigación del autor de este documentos, que concreta las fuentes de información en percepciones de distintos métodos de *tracking*.

Como se puede apreciar en la Figura 2.6, el sistema detecta la entrada de órdenes mediante sensores que registran los movimientos del usuario. La naturaleza imprecisa de estos sensores, y la existencia de multitud de ellos, necesita ser fusionado para suministrar una orden final lo más precisa posible.

El sistema que realiza esta fusión se ha implementado utilizando Redes Bayesianas. La estimación de la posición del brazo del usuario, para determinar hacia dónde está apuntando, se realiza mediante un modelo probabilístico basado en los distintos sensores instalados en torno al usuario.

El usuario también es capaz de interactuar con el sistema mediante *comandos de vocales*, utilizando un sistema de reconocimiento de voz. Sin embargo, la red no contemplaba esta entrada.

La principal conclusión del estudio sostiene que el 45% de la precisión del sistema depende de cómo se realiza esa fusión de las distintas fuentes de información, por lo que el uso de Redes Bayesianas consigue un sistema eficiente y más robusto.

### Fusión en un sistema adaptativo móvil

El artículo *Bridging the gaps: Hybrid tracking for adaptive mobile augmented reality*, de Hallaway et al [9] trabaja sobre un sistema de posicionamiento en interiores basado en distintas fuentes de posicionamiento, como *RFID*, y que necesitan un nivel de precisión muy alto pero que no siempre es posible conseguirlo. En la Figura 2.7 se muestra el mapa sobre el que se imprimen los resultados experimentales realizados con esta aplicación.

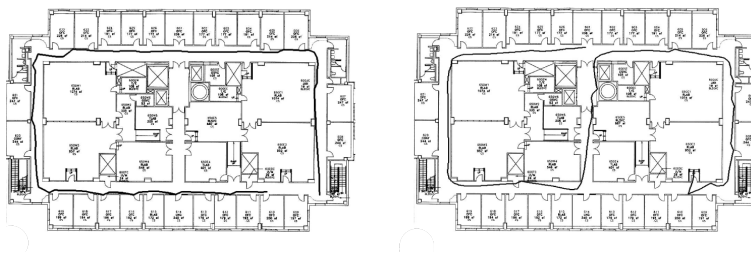


Figura 2.7: Plano del edificio donde se aplica el sistema de posicionamiento móvil

El sistema para la fusión de las distintas fuentes trabaja con algoritmos heurísticos basados en el entorno concreto en el que se despliega, es capaz de coordinarlos en tiempo real y de forma concurrente, y proporciona resultados variables dinámicamente para adaptar la interfaz virtual al movimiento del usuario.

Como en el caso de la investigación anterior, se utilizan Redes Bayesianas para realizar esta fusión. En este caso concreto, las redes se ajustan para determinar la posición del usuario teniendo en cuenta los distintos puntos de acceso localizados en el edificio, así como su posición y la intensidad de la señal.

De esta forma, dada una serie de puntos de acceso (*Wifi*, *RFID*, etc), la red trata de inferir la posición del usuario a partir de la intensidad de la señal recibida en un momento dado por cada uno de ellos.

Las redes son especialmente útiles en este caso porque la intensidad de cada punto de acceso no es constante en cada punto, sino que está sujeto a variaciones por la acción de otros elementos (cantidad de personas en el edificio, clima, etc).

La propia naturaleza de los sensores utilizados en este sistema hacen que la determinación de la localización tenga unos niveles de precisión muy bajos. Se ha demostrado que el uso de Redes Bayesianas puede paliar estos efectos, aunque no es posible alcanzar la eficacia deseada. Estos tipos de sistemas de posicionamiento son muy poco precisos hoy en día.

### Reconocimiento de interacción entre dos personas

En el artículo *A hierarchical bayesian network for event recognition of human actions and interactions*, de Park et al [31] se describe un sistema diferente a los dos anteriores, en el que el objetivo principal no es el posicionamiento, sino el reconocimiento de la naturaleza de la interacción entre los seres humanos.

En este trabajo, se propone un sistema para la detección de distintos eventos de interacción entre dos personas, a un nivel superior. Este tipo de detección tiene un alto grado de dificultad debido a la gran cantidad de movimientos y partes del cuerpo involucrados.

El sistema propuesto para la detección mediante Redes Bayesianas es jerárquico. Se utilizan dos niveles. Por un lado, se utiliza una red de bajo nivel para la inferencia de la posición de cada una de las partes del cuerpo de cada persona. Por el otro, una Red Bayesiana de alto nivel infiere, partiendo de la información de la anterior, los eventos que pueden estar ocurriendo durante la interacción.



Figura 2.8: Ejemplo de escenario de interacción entre dos personas.

En la Figura 2.8, se puede ver, en la parte izquierda, a dos personas realizando un tipo de interacción (en este caso, un abrazo). En la imagen de la derecha, se puede observar cómo la Red Bayesiana de menos nivel reconoce, a nivel de usuario, la posición de cada uno de los miembros (coloreados de forma diferente). La segunda Red Bayesiana determinaría, en base a la posición de cada miembro, que el evento que se está produciendo es un abrazo.

Las Redes Bayesianas dinámicas han conseguido dar una buena respuesta para la implementación de este tipo de eventos, de naturaleza muy incierta, ya que una misma interacción tiene el inconveniente de no poder observar todos los miembros del usuario de forma simultánea (existe oclusión). El sistema consigue, de forma eficiente, reconocer el caso concreto de interacción entre dos personas.

Como trabajo futuro de este proyecto, se propone aumentar el número de personas, o incluso realizarlo utilizando masas de gente, e incluso utilizar más de una fuente de video desde distintas perspectivas.

### Fusión basada en Lógica Difusa

La Lógica Difusa [41] es un tipo de lógica *multivaluada* o lógica *probabilística*, introducida por Lofti Zadeh en 1965. Actualmente tiene multitud de campos de aplicación, como la inteligencia artificial o la ingeniería electrónica y microelectrónica.

Se basa en razonamientos que, en vez de ser exactos, tienen una naturaleza *aproximada*, o con valores de probabilidad. Esta forma de modelado es más cercano a la forma intuitiva de pensar de los seres humanos, por lo que es capaz de representar eventos cotidianos, y dar una predicción muy acertada, sin que el modelo sea matemáticamente exacto.

Un sistema de lógica difusa puede verse como una caja negra, en el que se tienen variables de entrada, y variables de salida. A partir de las variables de entrada, que son valores numéricos exactos, se obtienen unos valores también exactos para las variables de salida.

A diferencia de los conjuntos clásicos **CRISP**, la lógica utiliza **conjuntos difusos** para modelar la pertenencia de un elemento a dicho conjunto. Mientras que el valor de pertenencia en los conjuntos CRISP era binario (1 pertenencia, 0 no pertenencia), en los conjuntos difusos se da un valor en el rango  $[0, 1]$ , indicando la *probabilidad de pertenencia a dicho conjunto*.

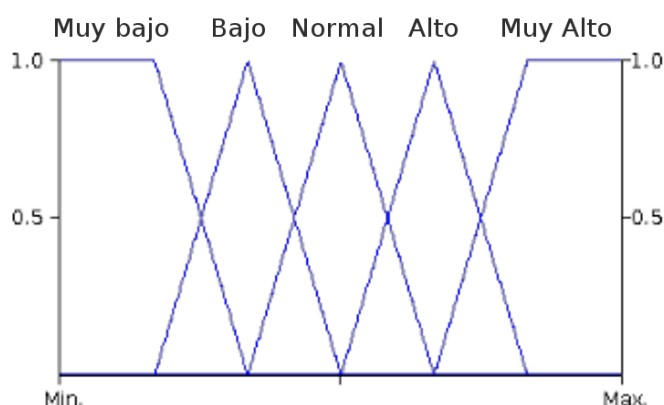


Figura 2.9: Ejemplo de conjunto difuso representante de la variable *altura*

Cada variable del sistema que se quiere representar tiene asociado un conjunto *difuso*. Estos conjuntos difusos, están subdivididos a su vez en *etiquetas lingüísticas*. Dependiendo del valor de la variable de entrada, tendrá un *valor de pertenencia* a cada una de las etiquetas lingüísticas del conjunto.

Póngase como ejemplo la variable de entrada de un sistema que mide la altura de una persona, con un valor mínimo de 1,40 metros y un valor máximo de 2,50 metros. Si queremos clasificar esos valores en *muy bajo*, *bajo*, *normal*, *alto*, *muy alto*, con los conjuntos clásicos habría que hacer una diferenciación clara de qué es bajo y qué es muy bajo. En la práctica, este problema es *difuso*.

En la Figura 2.9 se muestra el conjunto difuso asociado a esta variable. Cada etiqueta lingüística está definida por un trapecio o un triángulo que indica el *valor de pertenencia* (un valor entre 0 y 1) de un determinado valor de altura a una etiqueta. De esta forma, un valor de 1,55 metros, tendrá un valor de pertenencia de aproximadamente 0,6 a *Muy bajo*, 0,4 a *Bajo*, y de 0 al resto.

Para modelar la lógica de cálculo del sistema, se utilizan las denominadas *reglas difusas*, que no utilizan valores numéricos directamente, sino que utilizan las etiquetas lingüísticas. Estas reglas difusas son del tipo *si-entonces (if-then)*. En las condiciones de la regla se indican los valores de las variables de entrada, mientras que en el consecuente de la regla se ponen los valores de las variables de salida.

Un ejemplo de regla difusa sería:

```
IF (Altura IS Bajo AND Peso IS Alto) THEN Calidad IS Bajo;
```

Suponiendo que la variable *Altura* representa la altura de una persona, *Peso* su peso, y *Calidad* la calidad como jugador de baloncesto

De esta forma tan sencilla e intuitiva se puede modelar el comportamiento del sistema difuso, sin tener que recurrir a algoritmos exactos que, en la mayoría de los casos, son extremadamente difíciles de determinar por la complejidad del entorno que se quiere modelar.

Por último, es necesario un paso de *defuzzificación* para obtener el valor numérico de las variables de salida tras aplicar las reglas difusas. Existen diversos métodos para realizar esta transformación, que aportan distintas ventajas dependiendo de la naturaleza y características del sistema. Se puede encontrar más información en libro *Fuzzy sets and fuzzy logic* [22].

A continuación, se muestran algunas investigaciones que han utilizado la lógica difusa para fusión de información relativa a posicionamiento, o Realidad Aumentada.

### **Estimación de la posición de robots en terrenos irregulares**

El artículo *FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain*, de Ojeda et al [29], resalta las carencias inherentes de los sistemas de posicionamiento de robots actuales. Estos se basan en un posicionamiento, bien absoluto o relativo, con la consecuente acumulación de error.

Ellos proponen un sistema basado en lógica difusa (*Fuzzy Logic Expert Rule-based Navigation: FLEXnav*, que aprovechando la gran mejora en la precisión de los giroscopios actuales, son capaces de corregir el error cometido por otros sistemas basados en odometría en terrenos rugosos o irregulares.

Como se muestra en la Figura 2.10, el sistema de lógica difusa tiene como entradas la información detectada por un giroscopio (con un valor por cada uno de los tres ejes), un acelerómetro (midiendo en los ejes  $x$  e  $y$ ), y un sensor que indica las revoluciones de cada una de las cuatro ruedas de forma independiente.

El sistema es capaz de fusionar la información proveniente de las distintas fuentes para corregir el error provocado por la naturaleza irregular del terreno, error que otros sistemas no serían capaces de minimizar.

### **Posicionamiento en carreteras de alta densidad usando GPS**

El artículo *A High Accuracy Fuzzy Logic Based Map Matching Algorithm for Road Transport*, de Quddus et al [33] parte de la base de que en los sistemas de posicionamiento basados en *GPS* y mapas digitalizados para vehículos de transporte, la calidad del *matching* vehículo/carretera es bastante buena en zonas donde la densidad de vías de circulación es baja. Sin embargo, se ha podido observar que en núcleos donde la densidad y la complejidad de la red de carreteras es mayor, este *matching* es mucho peor.

En este trabajo se propone la utilización de un sistema difuso, que tome ventaja de la imprecisión característica de la lógica difusa para determinar la posición del vehículo dentro de esta compleja red de carreteras.

Las entradas del sistema son la posición absoluta de un sistema de navegación como *GPS* aumentada con información de sensores de navegación. Los experimentos llevados a cabo concluyen que el sistema difuso arrojan una mejora significativa sobre sistemas que son ampliamente aceptados para la navegación utilizando vehículos de transporte.

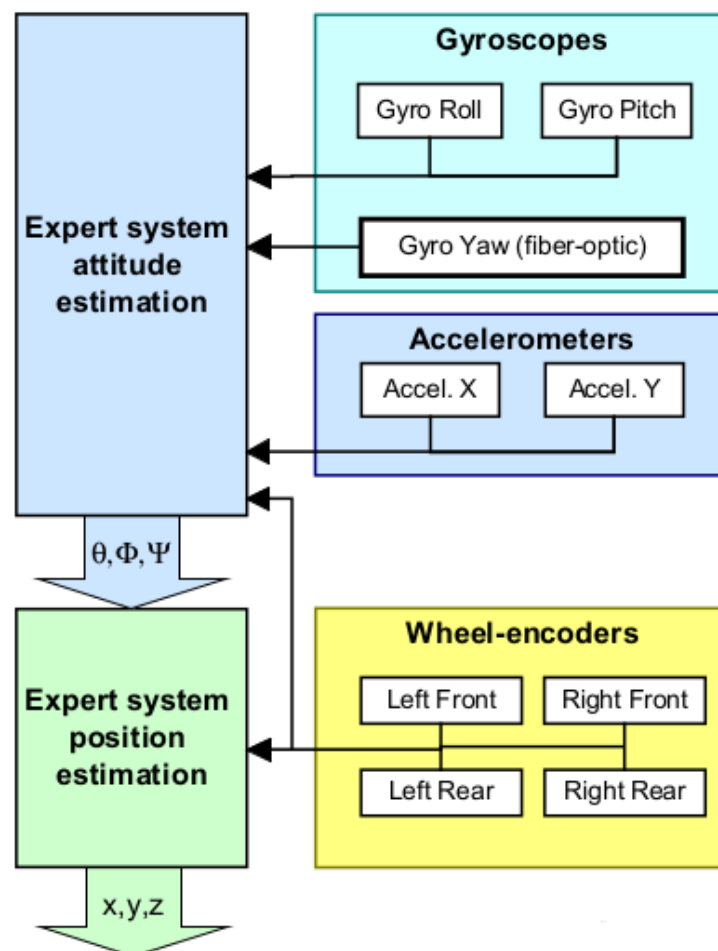


Figura 2.10: Diagrama de bloques del sistema difuso de posicionamiento de Ojeda et al [29]

---

# 3

---

## Descripción de la propuesta

---

En este Capítulo se describe la arquitectura propuesta para el desarrollo del *Sistema multimodal de Navegación en interiores* enmarcado dentro del proyecto ELCANO. Por lo tanto, se limita únicamente al alcance de la arquitectura del sistema de navegación, puesto que el proyecto ELCANO es mucho más amplio y hay aspectos de él que no entran dentro del objetivo de este documento.

De igual forma, en este documento solo se recogen los elementos de la arquitectura en los que ha trabajado directamente, desarrollándolos total o parcialmente, César Mora Castro.

El Capítulo comienza con una descripción general de la arquitectura del sistema de navegación, identificando cada uno de los módulos que la componen y el entorno en el que se implementa. Después pasa a describir más en detalle cada uno de los módulos, especificando detalles de implementación y funcionalidad, y describiendo sus resultados.

### 3.1. Descripción general

El objetivo de la arquitectura del *Sistema multimodal de Posicionamiento* es proporcionar localización en interiores a la capa de representación del proyecto ELCANO. Este posicionamiento debe ser lo suficientemente preciso como para poder representar información virtual integrada en el mundo real utilizando Realidad Aumentada. Además, la eficiencia es un aspecto muy importante, dado que los dispositivos donde se utiliza esta aplicación no cuenta con una capacidad de cómputo muy amplia.

Una de las ventajas con las que se cuenta para este sistema de navegación, es que se puede saber de antemano el entorno en el que se va a desplegar, por lo que es posible realizar algún tipo de entrenamiento previo de la zona para mejorar el posicionamiento.

En el siguiente apartado se describe el entorno y los recursos con los que se ha contado para desplegar el sistema.

## Entorno y recursos

La aplicación ha sido entrenada para funcionar en la planta baja del edificio Fermín Caballero de Ciudad Real (Escuela Superior de Informática de la Universidad de Castilla-La Mancha). En esta zona se ha podido desplegar distintos servicios, para poder facilitar el acceso a ellos por parte de usuarios con necesidades especiales. Estos servicios son, por ejemplo, el museo de la informática, conserjería, o el salón de actos. Además, supone un ejemplo típico de edificio público, por lo que las condiciones son las más cercanas a las de un caso de uso futuro real.

Los recursos hardware y software utilizados para el desarrollo se muestran en la siguiente tabla:

<b>Software</b>	
<b>Sistema operativo</b>	Debian/Ubuntu
<b>Lenguaje de programación</b>	C/C++
<b>Biblioteca de renderizado</b>	Ogre3D
<b>Biblioteca de visión por computador</b>	OpenCV
<b>Middleware de comunicaciones</b>	ZeroC Ice
<b>Hardware</b>	
<b>Dispositivo de vídeo</b>	Quickcam Sphere AF Logitech y uEye IDS USB
<b>Tablet PC</b>	HP Pavilion tx2500 y HP Compaq tc1100
<b>Estación de desarrollo</b>	Intel Pentium D 3.00 GB x2 y Zotac MAG HD-ND01

Cuadro 3.1: Recursos hardware y software utilizados en el desarrollo del proyecto.

Una vez situado el contexto del proyecto y los recursos que se han utilizado en su desarrollo, en el siguiente apartado se describen los tres módulos principales que componen la arquitectura del sistema de posicionamiento.

## Módulos de la arquitectura

Como se muestra en la Figura 3.1, el *Sistema multimodal de Posicionamiento* está compuesto por tres módulos principales:

1. **Módulo de métodos de *tracking***: este módulo implementa las distintas interfaces para implementar los métodos de *tracking* existentes. Proporciona la arquitectura para que puedan ser ejecutados de forma simultánea e independiente, y enviar el posicionamiento de cada uno de ellos al controlador de los métodos de *tracking* (*TrackingController*). En la Sección 3.2 se detalla esta arquitectura con más profundidad.
2. **Módulo de refinamiento**: una vez que se han recibido las posiciones de cada uno de los métodos, se les aplica una fase de *refinamiento*, para minimizar en la medida de lo posible el error de posicionamiento que puedan contener. Este módulo contiene a su vez dos submódulos: el de refinamiento basado en histórico de percepciones, y el basado en algoritmos heurísticos. Ambos se explican en la Sección 3.3.



3. **Módulo de fusión de percepciones:** por último, cuando se tienen las posiciones refinadas de cada uno de los métodos en un instante determinado, se deben *fusionar* para producir una única posición final que recibirá el módulo de representación, y poder realizar la integración del mundo real y el virtual. Esta *fusión* busca maximizar la precisión y minimizar el error, y puede configurarse de múltiples formas. Actualmente se utiliza la lógica difusa para implementarlo. En la Sección 3.4 se explica este submódulo de fusión, **FuzzyWeight**.

Como se ha dicho anteriormente, este sistema está integrado dentro del proyecto EL-CANO, por lo que aprovecha algunas funcionalidades adicionales que le ofrece el resto del sistema. Estas funcionalidades son muy útiles a la hora de configurar y depurar el sistema, que se explican en la siguiente sección.

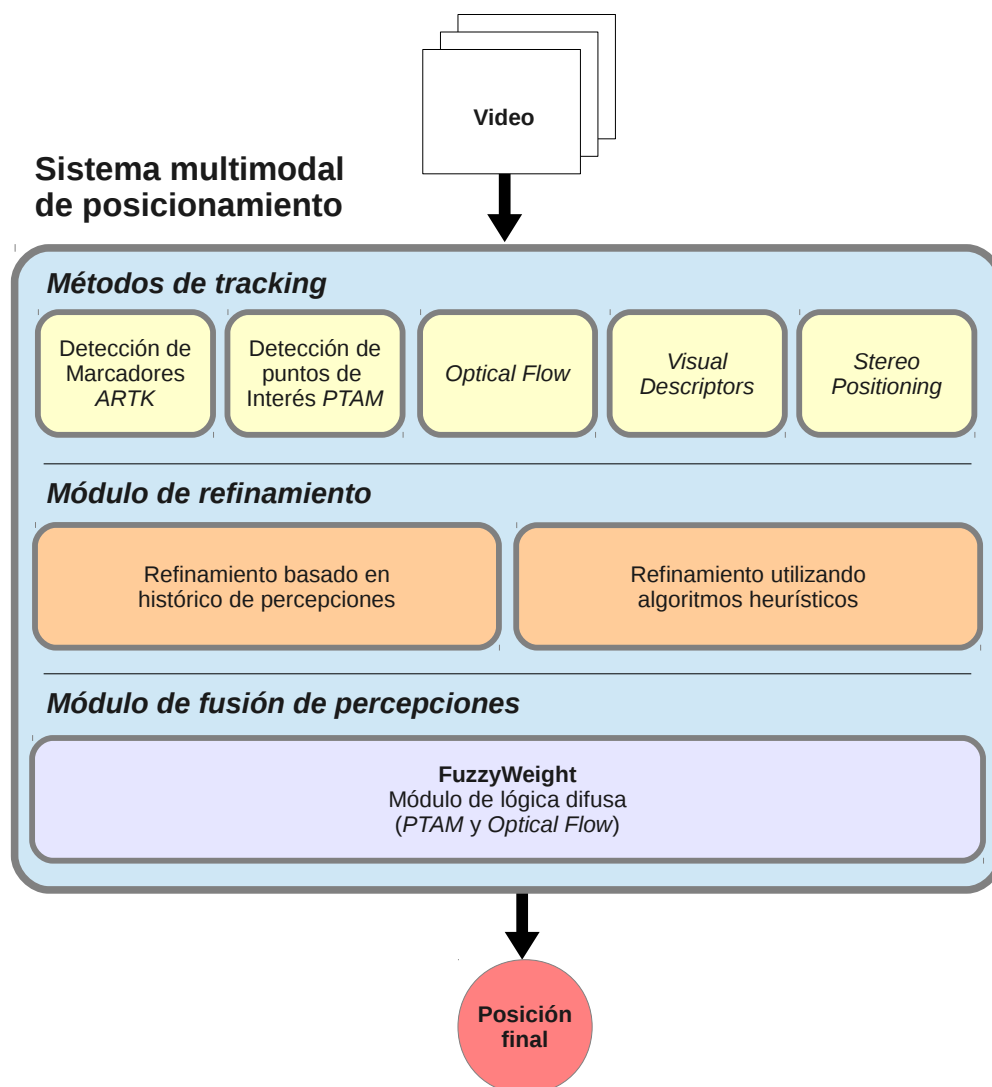


Figura 3.1: Arquitectura del Sistema multimodal de Posicionamiento.

## Funcionalidades de apoyo

La plataforma ELCANO proporciona utilidades que son útiles a la hora de desarrollar y configurar el sistema de posicionamiento, y proporcionar nuevas *features* al proyecto completo. Estas funcionalidades son:

- **Módulo de configuración:** este módulo del proyecto es capaz de leer distintos parámetros de un archivo de configuración (denominado *mars.cfg*), y leerlos en tiempo de ejecución. De esta forma, se pueden activar y desactivar los distintos submódulos o cambiar sus valores sin necesidad de compilar el proyecto entero.
- **Módulo de depuración:** este módulo proporciona un *log* implementado con un patrón *Singleton* accesible desde cualquier parte del proyecto, y proporciona salidas de depuración del tipo *nota*, *advertencia* y *error*. Todo ello es mostrado por consola, y escrito en el fichero *mars.log*.
- **Módulo de comunicaciones:** el módulo de comunicaciones permite mandar y recibir información al *Servidor Central* que gestiona las tareas que puede realizar el usuario, y debe recibir la posición de cada uno. Por ello, el sistema de posicionamiento está muy ligado a él, ya que la posición final calculada debe enviarla tanto a la capa de representación, como al *Servidor Central*. Además, el *Servidor Central* envía el posicionamiento basado en *Wi-Fi*.
- **Síntesis de voz:** el proyecto ELCANO dispone de un módulo de síntesis de voz para usuarios con discapacidad visual. Este módulo permite dictar las indicaciones que el sistema de navegación infiere que debe seguir para realizar una determinada tarea.

Hasta aquí se ha explicado todo lo necesario para entender el entorno y los recursos en los que se ha desarrollado y se despliega el sistema de posicionamiento, indicando hardware, software, entorno físico, y funcionalidades adicionales proporcionadas por el resto del sistema.

En las secciones siguientes se va a explicar el funcionamiento de cada uno de los módulos que se ha indicado que forman la arquitectura del *Sistema multimodal de Posicionamiento* para interiores.

## 3.2. Métodos de posicionamiento

El *Sistema multimodal de Posicionamiento* incorpora actualmente **cinco** métodos de *tracking* visuales, todos ellos basados en técnicas de Realidad Aumentada y Visión Artificial. Estos métodos utilizan la misma información visual para calcular la posición del usuario, por lo que todos están sincronizados. Esta información es suministrada por el módulo de vídeo del proyecto, con independencia del dispositivo utilizado.

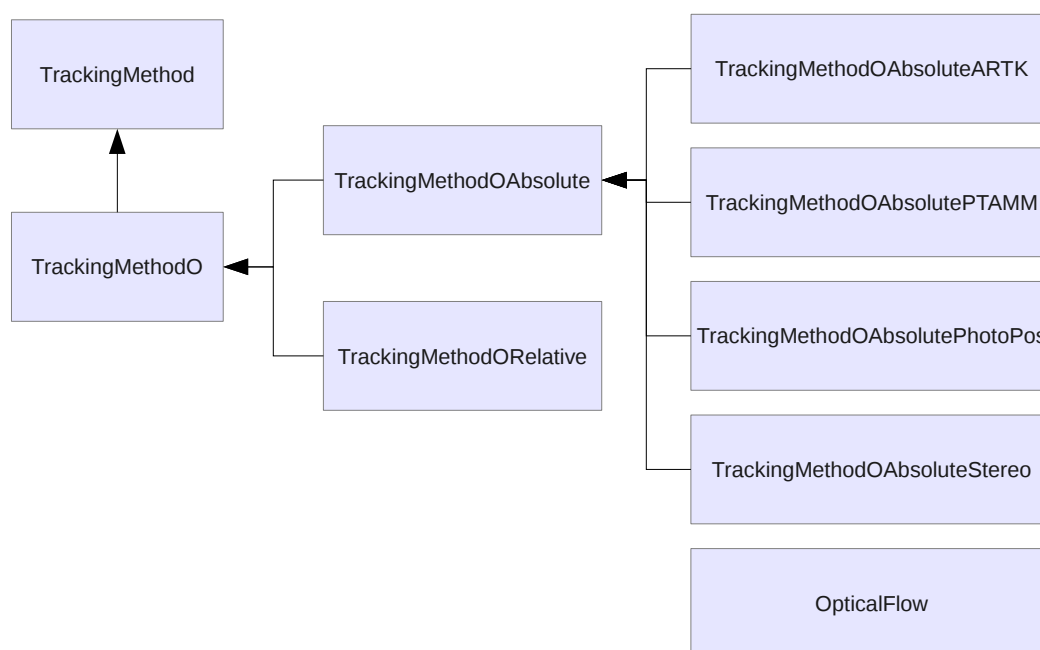


Figura 3.2: Diagrama de herencia de los métodos de *tracking*.

La arquitectura del proyecto ofrece las interfaces para implementar prácticamente cualquier tipo de método de *tracking*. En la Figura 3.2 se muestra el diagrama de herencia de estos métodos de *tracking*. Los distintos métodos se pueden clasificar en:

- Absolutos o Relativos: **TrackingMethodOAbsolute** o **TrackingMethodORelative**.
- Ópticos o no Ópticos: **TrackingMethodO** o **TrackingMethod**.

Algunas características comunes a todos los métodos de *tracking* (implementadas por la interfaz *TrackingMethod*), son el poder activarlos o desactivarlos, pausarlos, reaunadarlos, obtener la última posición o la última rotación.

Aunque *OpticalFlow* es un método de posicionamiento, no implementa ninguna de estas interfaces puesto que se integra directamente en el módulo de fusión de lógica difusa de la Sección 3.4.

Todos estos métodos de *tracking*, con independencia de su naturaleza o implementación, envían las posiciones a un módulo central denominado *TrackingController*. Este módulo se encarga de implementar el refinamiento y la fusión de las posiciones para conseguir una única percepción. Las implementaciones de estos módulos se explicará más adelante.

A continuación se explican, uno por uno, los métodos visuales de posicionamiento implementados en el *Sistema de Posicionamiento*.

## Detección de marcadores

Este método está implementado en la clase *TrackingMethodOAbsoluteARTK*, utilizando una biblioteca de detección de marcadores. Dicha biblioteca es ARToolKit [17].

Esta biblioteca necesita conocer previamente la forma y tamaño de las marcas que va a detectar. Estas marcas pueden contener cualquier forma o figura, y pueden ser entrenadas con una utilidad que proporciona el mismo *framework*. Dicha utilidad, una vez entrenada la marca, produce un fichero de descripción de la marca, que el sistema cargará para poder reconocerla.

Las dos funciones principales que tiene este método son:

- **Marcas de posicionamiento:** estas marcas tienen asociadas, aparte de un tamaño para poder ser reconocidas, una matriz de transformación que indica su posición dentro de la escena. Es decir, estas marcas están fijas en algún lugar del entorno, y proporcionan directamente la posición del usuario al ser detectadas. Esta matriz se da en forma de vector de 16 elementos en el formato *OpenGL*.

Para el proyecto ELCANO, se cuenta con un despliegue de este tipo de marcas (alrededor de 25) para proporcionar posicionamiento absoluto. Este tipo de método fue el único que se utilizó en una primera fase de la implementación del *Sistema de Posicionamiento*, aunque luego se implementó el basado en *puntos de interés*.

- **Marcas de actuación:** las marcas de actuación indican lugares donde se pueden realizar alguna acción o evento. No es necesario que contengan información de posicionamiento, puesto que pueden ser móviles. Un uso típico de este tipo de marcas es el de indicar el lugar donde se puede realizar una videoconferencia para pedir asistencia.

A parte de estos dos tipos de marcas, existe un tercer tipo que puede ser utilizado como marca de posicionamiento y de actuación.

Para indicar qué marcas están contenidas en el sistema y qué información hay asociada a ellas, el módulo de configuración del sistema lee de un archivo llamado *patterns.artk* este tipo de parámetros. El formato de este archivo es el siguiente:

```
[tipo de marca] [nombre de marca] [información de posicionamiento] [tamaño]
```

Donde el tipo de marca puede ser *a* para marcas de actuación, *p* para marcas de posicionamiento, y *b* para las marcas que cumplen ambos roles. En el caso de la marca de actuación, la información de posicionamiento debe omitirse. El nombre de la marca debe ser el nombre del archivo de descripción de la marca, y el tamaño es un número decimal expresado en metros.

El siguiente es un ejemplo de este tipo de archivo de configuración:

```
a 4x4_73.patt 0.12
p 4x4_10.patt 1 0 0 0 0 1 0 0 0 0 1 0 0.79 0.84 11.83 1 0.12
b 4x4_16.patt 0 0 1 0 0 1 0 0 -1 0 0 0 3 0.84 5.79 1 0.12
```

Por otro lado, el algoritmo debe tener un valor de *threshold* (*umbral*) que se puede configurar para realizar la binarización de la imagen (convertirla a blanco y negro). Los dispositivos de vídeo que se utilizan en el sistema son lo suficientemente inteligentes como para adaptar el obturador y la ISO de forma dinámica a las condiciones de luz del entorno, por lo que la imagen resultante puede aclararse o oscurecerse de forma automática por la cámara, y esto afecta de manera significativa a la calidad del *tracking* de este método.

Para ello se ha implementado un algoritmo que adapta el *threshold* de forma dinámica a las condiciones lumínicas en caso de que la detección de las marcas sea mala. El pseudocódigo del algoritmo es el siguiente:

```
1  #define thresDir 2
2  if(!detecting){
3      nFails++;
4      if(nFails>MIN_FAILS){
5          int t=getThreshold();
6          t+=thresDir;
7          setThreshold(t);
8
9          if(t>MAX_THRESHOLD || t<MIN_THRESHOLD){
10             thresDir*=-1;
11         }
12     }
13 }
```

El algoritmo funciona como sigue: si detecta que el número de fallos de detección consecutivos es significativo (mayor que *MIN\_FAILS*, el *threshold* comienza a variar de forma lineal en el rango definido por *MIN\_THRESHOLD* y *MAX\_THRESHOLD*, hasta que el número de fallos de detección comience a descender.

El inconveniente de este algoritmo es que un *fallo de detección* puede ser simplemente que no hay ninguna marca en el campo de visión del usuario, pero no supone una gran penalización puesto que el algoritmo no aporta una carga significativa de cómputo al sistema.

Para determinar la calidad de la detección de la marca, se ha implementado un algoritmo que, tomando como parámetros la distancia y el ángulo del usuario relativo a la marca, le asigna un valor de calidad entre 0 y 1. Este valor se utiliza para, si es capaz de detectar más de una marca en un momento dado, la posición devuelta es la media ponderada teniendo en cuenta esa estimación de calidad.

## Detección de puntos de interés

El método de detección de puntos de interés está implementado en la clase *Tracking-MethodOAbsolutePTAMM*, utilizando el algoritmo desarrollado en la Universidad de Oxford por George Klein llamado *PTAM* [21]. Es el algoritmo utilizado para reemplazar el posicionamiento con marcas usado en una primera versión del proyecto ELCANO.

Este algoritmo implementa, a grandes rasgos, dos hilos de ejecución paralelos:

- **Tracking:** el algoritmo funciona construyendo un mapa de puntos en 3D, donde cada punto representa un punto de interés 2D detectado en la escena. Estos mapas deben ser inicializados mediante una inicialización estéreo al comienzo de la ejecución. Esta inicialización toma dos *frames* separados por una pequeña distancia, realiza la detección de puntos de interés, y realiza el *matching* entre los puntos detectados entre los dos *frames*.

Además, este algoritmo es capaz de dar una estimación de la calidad del posicionamiento dado, identificado por tres valores: *MALO*, *MEDIO* y *BUENO*.

Con este *matching* inicial se construye el mapa 3D inicial, y se utiliza para la ejecución posterior. Este hilo únicamente detecta los puntos de interés del *frame* actual, los localiza en el mapa 3D, y a partir de esa información calcula la posición del usuario.

- **Mapping:** el hilo de *mapping* se ejecuta una vez inicializado el primer mapa 3D, y de forma independiente al *tracking*. Se encarga de obtener los puntos de interés detectados por el *tracking* e intentar añadir nuevos puntos 3D al mapa, o eliminar los que tengan poca calidad, para de esa forma ir mejorando el mapa con la ejecución.

Este es el funcionamiento básico del algoritmo *PTAM*. Sin embargo, en la implementación del proyecto *ELCANO* se utiliza una versión mejorada del software denominada *PTAMM*. Esta versión añade como funcionalidad la detección y carga de varios mapas simultáneamente. De esta forma, se puede entrenar un mapa, almacenarlo en un archivo *xml* y utilizarlo en tantas ejecuciones como se desee. Además en una misma ejecución se pueden tener cargados varios mapas, por lo que el área de detección aumenta considerablemente.

En el *Sistema de Posicionamiento*, este método implementa la funcionalidad de poder activar o desactivar el hilo de *Mapping*, para poder liberar de carga al sistema. El mapa se inicializa y mejora en una fase de entrenamiento de la zona inicial, para luego almacenarlo y cargarlo en la aplicación sin necesidad de seguir realizando modificaciones sobre él.

Por lo tanto, se deduce que para que este método funcione correctamente, es necesaria una fase de entrenamiento previa. El proceso y las aplicaciones utilizadas en este entrenamiento se explican en la siguiente sección.

### Entrenamiento de los mapas

El proceso de entrenamiento de un único mapa para ser utilizado con el *Sistema de Posicionamiento* implica dos aplicaciones diseñadas expresamente para este propósito. Estas aplicaciones son:

1. **AutoTunning:** esta aplicación es una variación de la aplicación *PTAMM* original.

En primer lugar, realiza la inicialización estéreo del mapa, y obtiene los primeros puntos de interés y los puntos 3D, por lo que es capaz de realizar el posicionamiento.

Sin embargo, un problema importante de estos mapas recién entrenados es que las unidades de medida no corresponden. Es necesario decir a cuántos metros equivale una unidad de medida de *PTAMM*, para poder realizar el posicionamiento correctamente. Para ello, la aplicación *AutoTunning* también ejecuta el algoritmo de detección de marcas. Esta aplicación espera encontrar una marca, que sabe de forma precisa

cuánto mide, y trata de localizar los cuatro puntos 3D que definen las cuatro esquinas de la marca dentro del mapa 3D creado por *PTAMM*.

Para llevar a cabo este *matching* de puntos, se tiene en cuenta algunos indicadores estadísticos como la *varianza* existente entre las distintas distancias entre los puntos 3D identificados como puntos de la marca.

Una vez realizado el proceso de *matching* correctamente, se escala el mapa entero para adecuar la medida a la medida real de la marca. De esta forma tenemos un mapa completamente funcional, y escalado a la medida real de la escena.

Pero aún queda un problema que resolver. El centro de coordenadas utilizado por este mapa no es posible controlarlo *a priori*, por lo que es necesario *posicionar* dicho mapa dentro del sistema de coordenadas global utilizado en la aplicación. Para ello se utiliza la siguiente aplicación.

2. **FineTunning:** esta aplicación es una variación con menos funcionalidad del proyecto ELCANO. Ejecuta el método de detección de puntos de interés, y representa de forma gráfica un *eje de coordenadas* en el centro de coordenadas local del mapa (el cual no coincide con el centro global). Al estar el mapa escalado, las medidas coinciden con las reales, por lo que únicamente hay que llevar, utilizando las flechas, el centro de coordenadas local del mapa hacia el global.

Una vez indicado esto, la aplicación almacena en un fichero denominado *PTAMMOffset.mat* la matriz de posicionamiento global del mapa. Este fichero está en un formato que el proyecto puede leer y aplicar para representar el mapa en términos del centro global de coordenadas. Este formato utilizado es el siguiente:

```
[(número de mapa)] (matriz de posicionamiento)
```

El número del mapa es un parámetro importante, puesto que la aplicación puede usar más de uno simultáneamente.

Con el proceso anteriormente explicado se consigue un mapa, escalado y posicionado con el formato apropiado para utilizar con el proyecto ELCANO. Suministrándole esta información, se pueden añadir tantos mapas como se desee. Para indicar a la aplicación el número de mapas que debe de cargar, se debe indicar en el archivo de configuración *mars.cfg* explicado anteriormente.

## Optical Flow

Este método se ha implementado en la clase *OpticalFlow*, utilizando la biblioteca libre de visión artificial *OpenCV*. El método detecta el desplazamiento relativo en un determinado momento, analizando los vectores de movimiento entre dos *frames* separados una pequeña distancia temporal. Los valores arrojados para indicar esta cantidad de movimiento y el sentido pertenecen al rango  $-15, 15$ .

Para la implementación de los algoritmos, se han utilizados dos funciones de la biblioteca *OpenCV*.

1. *goodFeaturesToTrack()*: realiza la detección de *corners*, o puntos de interés sobre el primer frame, para luego intentar encontrarlos en el segundo frame, y calcular el desplazamiento.

2. *calcOpticalFlowPyrLK()*: dados los puntos de interés devueltos por la función anterior, y los dos *frames*, detecta la posición de esos mismos puntos en el segundo frame. Es la que realiza el *Optical Flow* propiamente dicho.

Estas funciones reciben varios parámetros que pueden ser ajustados para realizar una búsqueda mas eficiente en el *Optical Flow*.

La información del *Optical Flow* se almacena en una estructura estándar para ser utilizada por el módulo de fusión basado en lógica difusa. El código de la estructura es la siguiente:

```
struct RelativeMovement
{
int horMov ;
int vertMov ;
int zoomMov;
bool movDetected ;
};
```

Cada uno de los campos, *horMov*, *vertMov* y *zoomMov* cuantifican la cantidad y el sentido del movimiento en cada uno de los tres ejes respectivamente. La variable *movDetected* almacena si el método de posicionamiento ha conseguido obtener la localización del usuario en un instante. Esta misma estructura es utilizada por el método de *tracking* basado en puntos de interés para proporcionar su propia entrada de movimiento relativo al sistema difuso.

## Visual Descriptors

El método de posicionamiento basado en descriptores visuales se ha implementado en la clase *TrackingMethodOAbsolutePhotoPos*. La implementación fue realizada para el Proyecto de Fin de Carrera del alumno Ricardo Muñoz Castellanos.

Su funcionamiento se basa en la utilización de descriptores. Los descriptores son una forma de representar, mediante una estructura de datos, el contenido de la imagen de acuerdo a un formato, de forma que utilizando esos descriptores se puede determinar cuán parecidas son dos imágenes, o hacer búsquedas entre los descriptores.

Los descriptores utilizados en esta implementación han sido tres: *Color Layour*, *Edge Histogram* y *Scalable Color*. Para ello se ha utilizado la biblioteca libre *MPEG7FexLib*.

Para comenzar a utilizar este método de *tracking* es necesario tener un repositorio de imágenes que se quieren identificar. En el caso concreto de este proyecto, se realizaron múltiples fotografías de la planta baja de la Escuela de Informática de Ciudad Real, almacenando por cada una de ellas la posición y la orientación en la que fue tomada. En total, se realizaron más de 400 fotografías.

A este repositorio de imágenes se les realizó una indexación utilizando los descriptores mencionados anteriormente, y esta información se almacena en una base de datos. Esta información es la que se utilizará durante la ejecución del algoritmo para encontrar la posición del usuario. Para agilizar la búsqueda en la base de datos, se ha utilizado como estructura de datos un *Arbol de Voronoi*. Esta estructura de datos almacena los elementos



minimizando la *distancia entre ellos*. En este caso, la distancia mide la similitud entre dos imágenes. Por lo tanto, al realizar una búsqueda, según se profundice en el árbol se asegura que se van dejando atrás las imágenes que más difieren de la que se está buscando.

Durante la propia ejecución, por cada *frame* obtenido se debe aplicar los descriptores de imagen. Los valores de estos descriptores se utilizarán para lanzar una consulta a la base de datos creada anteriormente, y encontrar la imagen cuyos coeficientes sean más similares.

Una vez identificada esa imagen de la base de datos, se extrae la posición y la orientación en que fue tomada, que se considera como la posición final del usuario en ese instante.

### Posicionamiento Estéreo

Por último, este método de posicionamiento está implementado en la clase *TrackingMethodOAbsoluteStereo*, realizado para el Proyecto de Fin de Carrera del alumno Ricardo Muñoz Castellanos.

Es un método relativo *a priori*, aunque de cara a la implementación en el *Sistema de Posicionamiento* se considera absoluto porque utiliza información de otros métodos para proporcionar la posición absoluta.

El funcionamiento consiste en mantener dos *frames* separados una cierta distancia en el tiempo, y realizar la triangulación como visión estereo. Durante la ejecución del algoritmo, lo que en un momento es el *frame derecho*, al siguiente pasa a ser el *izquierdo* y el *frame derecho* es sustituido por el *frame actual*.

En un primer momento, se realiza una búsqueda de *landmarks*, por algoritmos de *optical flow*. En las consecuentes ejecuciones, se trata de mantener esos *landmarks* e identificarlos en la imagen, para reducir el tiempo de cómputo. La extracción de *landmarks* se realiza mediante el algoritmo *Shi-Tomasi*, de la biblioteca OpenCV.

A partir de estos puntos, se extrae la matriz fundamental y la esencial. Para ello se utiliza el algoritmo *LMedS*, que es el que mejor estimación de la matriz fundamental consigue proporcionar. Con estas dos matrices es posible estimar la posición y rotación relativa de un *frame* a otro.

Este movimiento relativo es aplicado al movimiento absoluto proporcionado por otros métodos de *tracking* en el momento en que se tomó el primer *frame*. Por lo tanto, este método sirve como refuerzo al resto de métodos.

### 3.3. Módulo de refinamiento

Cuando los métodos de posicionamiento envían su estimación de localización del usuario al *TrackingController*, es posible que hayan cometido un error muy grande debido a la naturaleza imprecisa de los métodos.

Este módulo se encarga de detectar y corregir ese tipo de errores utilizando dos técnicas de refinamiento: la primera es una basada en el histórico de percepciones, y la segunda utilizando algoritmos heurísticos.

Estos módulos pueden ser activados o desactivados de forma independiente para cada

método, no funcionan simultáneamente para cada método. Esta opción de diseño permite comprobar la mejora de resultados al realizar el posicionamiento.

A continuación se explica el funcionamiento de ambos métodos.

### Refinamiento basado en histórico

Este refinamiento de las percepciones se basa en la capacidad del sistema de almacenar las últimas  $n$  percepciones. Este parámetro es configurable en tiempo de compilación.

Cuando se recibe una última percepción, se calcula una media ponderada con las cuatro últimas percepciones, de forma que la percepción refinada es la resultante de la siguiente fórmula:

$$P_{actual} = P_1 * 0,5 + P_2 * 0,25 + P_3 * 0,15 + P_4 * 0,1$$

Siendo  $P_1$  la percepción más reciente, y el resto las percepciones almacenadas en el histórico. De esta forma, se da más peso a la percepción más actual, pero se tiene en cuenta las anteriores. El resultado es un movimiento más fluido y suave, aunque da la sensación de haber un *delay* al momento de representar la posición. Este retraso es lógico, al tener en cuenta percepciones pasadas.

### Refinamiento utilizando algoritmos heurísticos

Este módulo implementa un algorítmico heurístico adaptado al método basado en puntos de interés. Su desarrollo vino motivado por la observación de que este algoritmo arroja, con una cierta frecuencia, percepciones que son totalmente erróneas. El funcionamiento del algoritmo es el siguiente:

Teniendo como entrada la última posición suministrada por el método basado en puntos de interés, y la estimación que proporciona de calidad (con los valores *MALO*, *MEDIO* y *BUENO*), se aplica el siguiente pseudocódigo:

```

1  switch(trackingQuality){
2      case MALO or MEDIO:
3          if(primerapercepcion){
4              primeraPercepcion=false;
5              ultimaPercepcionBuena = percepcionActual;
6              percepcionFinal = percepcionActual;
7          }
8          distancia = distanciaEntrePercepciones(perceptionActual, ultimaPercepcionBuena);
9
10         if(distancia > MAX_DISTANCE){
11             percepcionFinal = ultimaPercepcionBuena;
12             vector = percepcionActual - ultimaPercepcionBuena;
13             percepcionFinal = percepcionFinal + HEUR_DESP * vector;
14             ultimaPercepcionBuena = percepcionFinal;
15         }else if(distancia < MIN_DISTANCE){
16             //No hacer nada
17         }else{
18             percepcionFinal = percepcionActual;
19             ultimaPercepcionBuena = percepcionActual;
20         }
21         break;

```

```
22     case BUENO:
23         ultimaPercepcionBuena = percepcionActual;
24         percepcionFinal = ultimaPercepcionBuena;
25         break;
26     }
```

Las variables, constantes y funciones que utiliza son las siguientes:

- *percepcionActual*: percepción actual suministrada por el método de *tracking* para ser refinada. Es la percepción de entrada.
- *trackingQuality*: calidad estimada por el método de *tracking* sobre la *percepcionActual*. Este valor puede ser *MALO*, *MEDIO* o *BUENO*.
- *primeraPercepcion*: variable que indica si es la primera vez que se ejecuta el algoritmo. Sirve por motivos de inicialización.
- *ultimaPercepcionBuena*: almacena la última percepción obtenida que se consideró con calidad *BUENA*.
- *percepcionFinal*: es la percepción final que se devolverá al *TrackingController*. Se considera la percepción refinada.
- *MIN\_DISTANCE*: parámetro configurable que indica la mínima distancia que debe haber entre dos percepciones consecutivas para que se considere que el usuario se ha movido.
- *MAX\_DISTANCE*: parámetro configurable que indica la mayor distancia posible entre dos percepciones consecutivas.
- *HEUR\_DESP*: parámetro configurable que se utiliza para cuantificar qué cantidad de movimiento se considera que ha tenido el usuario en caso de que la percepción necesite ser refinada.
- *distanciaEntrePercepciones()*: función que calcula la distancia en metros entre dos percepciones dadas.

El algoritmo tiene dos maneras de actuar diferenciadas en función de la calidad de la posición suministrada:

1. Calidad *MALA* o *MEDIA*: Si es la primera vez que se ejecuta el algoritmo (*primeraPercepcion==true*), se inicializan las variables *ultimaPercepcionBuena* y *percepcionFinal* con la posición actual recibida. Sólo la primera vez con fines de inicialización.

En primer lugar se calcula la distancia entre la última posición buena conocida (se supone que el usuario estaba en esa posición) y la posición actual suministrada por el método.

Si la distancia es mayor que el máximo definido por el parámetro *MAX\_DISTANCE*, se considera que la percepción es errónea. En este caso, la percepción final será la percepción actual más un desplazamiento en dirección hacia la posición actual, pero sólo en un porcentaje indicado por la variable *HEUR\_DESP*. Además, se considera esta posición final como la última buena conocida.

Si la distancia desplazada es menor *MIN\_DISTANCE*, que es establece la distancia mínima, no hay cambio de posición. De esta forma, se suaviza el movimiento ya que se

cortan los desplazamientos muy pequeños, que pueden deberse a pequeños errores en la percepción actual.

Si se considera que la distancia recorrida ha sido un valor razonable (está comprendida entre los valores `MIN_DESP` y `MAX_DESP`), se admite la posición actual como buena, por lo que posición final y la última posición buena conocida se actualizan con dicho valor.

2. Calidad *BUENA*: si la calidad del *tracking* es directamente buena, se adopta la percepción actual directamente como percepción final y como última buena conocida.

Los valores indicados en el algoritmo se pueden configurar desde el archivo de configuración *mars.cfg*, por lo que personalizarlo para mejorar los resultados es significativamente más sencillo. Además se puede activar o desactivar completamente desde el mismo archivo de configuración.

Sin embargo, este algoritmo no deja de ser *a ciegas*, sin tener en cuenta ningún otro tipo de información. Esa fue la motivación para desarrollar el módulo de fusión basado en lógica difusa. Este módulo se explica en la siguiente sección.

### 3.4. Módulo de fusión

Este método de fusión se implementó como una mejora sobre el módulo de refinamiento heurístico, ya que pretende mejorar las percepciones proporcionadas por el método basado en puntos de interés utilizando como entrada la información suministrada por el método de *optical flow*. Para ello se escogió la lógica difusa como modelo para implementar esta fusión.

A continuación se explica *FuzzyWeight*, el módulo de lógica difusa que realiza esta fusión.

#### Introducción a *FuzzyWeight*

El objetivo principal de *FuzzyWeight* es indicar qué fiabilidad tienen las percepciones del algoritmo *PTAM* tomando como referencia el *Optical Flow* de las imágenes.

El problema de *PTAM* es que, al realizar el *matching* entre los *feature points* 2D y los puntos 3D del mapa construido, es posible que sea errónea y dar percepciones totalmente equivocadas. El resultado es que, de un instante a otro, el mundo virtual se desplaza a un lugar incorrecto.

*FuzzyWeight* es un sistema difuso que infiere el peso (un valor entre 0 y 1) que debe darse a una percepción proporcionada por el algoritmo de tracking *PTAM* teniendo en cuenta el histórico de percepciones de este, y el *Optical Flow* actual. De esta forma, si *PTAM* indica un desplazamiento muy grande, causado por un *matching* erróneo, pero el *Optical Flow* detecta que no existe tal desplazamiento, el peso que se le otorgará a esa percepción de *PTAM* será prácticamente cero.

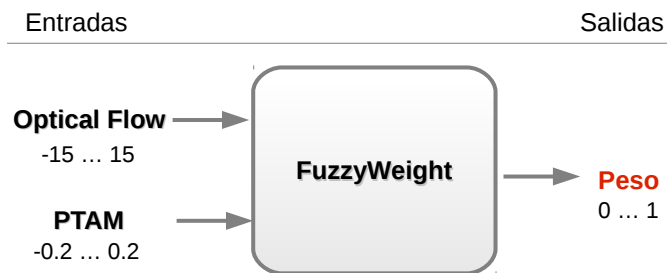


Figura 3.3: Arquitectura general de *FuzzyWeight*

En la Figura 3.3 se muestra la arquitectura básica de *FuzzyWeight*. Como se puede apreciar, el sistema tiene como entrada dos variables:

- **Optical Flow:** Este módulo indica dos tipos de desplazamiento: en el eje x (horizontal) y en el eje y (vertical). Los valores pertenecen al rango  $-15, 15$ . Estos valores indican la cantidad de movimiento, y el sentido, bien en el eje horizontal o en el vertical.
- **PTAM:** A partir del algoritmo de *tracking PTAM* se ha implementado un modulo que ofrece la cantidad de movimiento relativo en los últimos instantes, utilizando el histórico de percepciones ya implementado. Por afinidad al modulo de *Optical Flow*, los valores que proporciona también son de forma separada en el eje horizontal y en el eje vertical. El rango de valores de esta variable es  $-0,2, 0,2$

La variable de salida, denominada *Peso*, es una variable en el rango  $0, 1$  que indica cuánta fiabilidad (1 la máxima, 0 la mínima) tiene la percepción proporcionada por *PTAM* teniendo en cuenta el *Optical Flow*.

Puesto que las variables de entrada proporciona dos tipos de datos, horizontales y verticales, se utilizan dos sistemas *FuzzyWeight*. El primero calcula un peso de fiabilidad en movimiento horizontal, y el otro en movimiento vertical. El peso global es la media aritmética de ambos pesos.

## Conjuntos difusos

Para especificar las características de los conjuntos difusos, se ha utilizado la experiencia dentro del desarrollo del proyecto ELCANO. En la Figura 3.4 se muestran los conjuntos difusos y las etiquetas correspondientes cada variable. Uno de los pasos en la *fuzzificación* de las variables es crear etiquetas lingüísticas. El significado de cada etiqueta, y la definición de cada conjunto, es:

- **NG:** Negativo grande.  
*Optical Flow* (1/-15, 1/-10, 0/-5)  
*PTAM* (1/-0.2, 1/-0.15, 0/0.05)
- **NP:** Negativo pequeño.  
*Optical Flow* (0/-10, 1/-5, 0/0)  
*PTAM* (0/-0.15, 1/-0.05, 0/0)

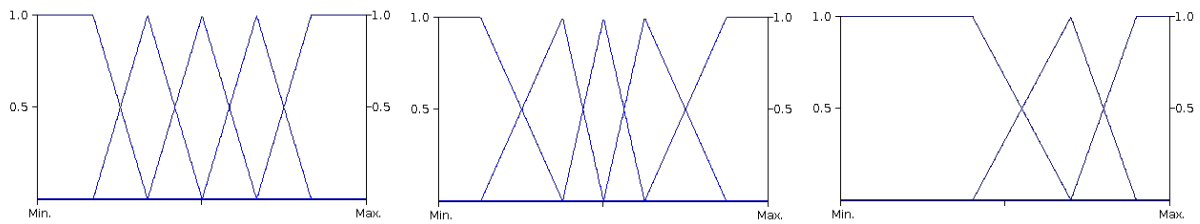


Figura 3.4: Variables *fuzzificadas* (de izquierda a derecha, *Optical Flow*, *PTAM* y *Peso*).

- **Z:** Cero.  
*Optical Flow* (0/-5, 1/0, 0/5)  
*PTAM* (0/-0.05, 1/0, 0/0.05)
- **PP:** Positivo pequeño.  
*Optical Flow* (0/0, 1/5, 0/10)  
*PTAM* (0/0, 1/0.05, 0/0.15)
- **PG:** Positivo grande.  
*Optical Flow* (0/5, 1/10, 1/15)  
*PTAM* (1/-0.2, 1/-0.15, 0/0.05)
- **LOW:** Peso bajo.  
*Peso* (1/0, 1/0.4, 0/0.7)
- **MED:** Peso medio.  
*Peso* (0/0.4, 1/0.7, 0/0.9)
- **HIGH:** Peso alto.  
*Peso* (0/0.7, 1/0.9, 1/1)

El siguiente paso es la definición de las reglas difusas en función de las etiquetas lingüísticas de cada variable. La definición de las reglas difusas de *FuzzyWeight* es:

```

R1/ if(OF == NG & PT == NG) -> P = HIGH;
R2/ if(OF == NP & PT == NP) -> P = HIGH;
R3/ if(OF == Z & PT == Z) -> P = HIGH;
R4/ if(OF == PP & PT == PP) -> P = HIGH;
R5/ if(OF == PG & PT == PG) -> P = HIGH;
R6/ if((OF == PP | OF == PG) & (PT == NP | PT == NG)) -> P = LOW;
R7/ if((OF == NP | OF == NG) & (PT == PP | PT == PG)) -> P = LOW;
R8/ if((OF == NP | OF == Z) & PT == NG) -> P = MED;
R9/ if((OF == PP | OF == Z) & PT == PG) -> P = MED;
R10/ if(OF == NG & (PT == NP | PT == Z)) -> P = MED;
R11/ if(OF == PG & (PT == PP | PT == Z)) -> P = MED;

```

Siendo *OF* la variable de entrada *Optical Flow*, *PT* la variable de entrada *PTAM*, y *P* la variable de salida *Peso*.

## Generación de *FuzzyWeight*

Para la generación del código de *FuzzyWeight* se ha utilizado la herramienta recomendada en la asignatura de *Técnicas de Softcomputing Xuzzy 3.0*<sup>1</sup>. Esta herramienta permite, mediante interfaz gráfica (aunque también se puede utilizar su lenguaje específico), especificar sistemas que hacen uso de lógica difusa. Además, la herramienta permite simular el sistema mediante la introducción de valores de entrada, ver la *defuzzificación*, e incluso generar código para los lenguajes C, C++ y Java. Para el caso concreto de *FuzzyWeight*, se ha generado el código para C, que luego se ha utilizado directamente en el proyecto ELCANO.

Ese código proporciona una función, cuyos parámetros son los valores de entrada y salida, para realizar la inferencia. El prototipo de la función de *FuzzyWeight* es:

```
void fuzzyWeightInferenceEngine ( double OF, double PT , double *P);
```

## Movimiento relativo de PTAM

El algoritmo *PTAM* proporciona percepciones, que no es más que posiciones y rotaciones del usuario en un momento dado. Pero se hace necesario calcular el movimiento relativo entre una percepción y otra. Para ello, se ha aplicado la siguiente fórmula:

$$M_{relativo} = M_{frame2} * M_{frame1}$$

Siendo  $M_{frame2}$  la percepción del frame anterior, y  $M_{frame1}$  la percepción del frame posterior, se obtiene la transformación relativa  $M_{relativo}$  entre ellas dos, que indica qué tipo de movimiento horizontal y vertical se ha realizado entre los dos instantes.

## Resultados

Para la comprobación de los resultados arrojados por este sistema, se han propuesto tres escenarios distintos. Cada uno de los escenarios está compuesto por un vídeo corto (alrededor de 5 segundos), con diferentes entornos: un entorno pequeño (de escritorio), un movimiento rotatorio en una habitación, y un movimiento con desplazamiento en un pasillo. Para cada escenario, se han tomado los valores de la posición del usuario obtenidos por *PTAM*, tanto con *FuzzyWeight* activado, como no.

En cada uno de los escenarios, se compara de forma directa, mediante gráficas, el comportamiento del sistema sin *FuzzyWeight*, y con él. Aunque el sistema sin *FuzzyWeight* es más rápido (tiene una tasa mayor de *frames per second*), la desviación entre percepciones es mucho menos con él.

Sin embargo, uno de las fuentes de error más importantes de este sistema, se da en los movimientos a lo largo del eje z. En este tipo de movimientos, *PTAM* detectará un desplazamiento, que será real. Sin embargo, el *Optical Flow* no detectará movimiento alguno, ya que sólo tiene en cuenta el eje x y el eje y, por lo que le restará peso a una percepción de *PTAM* que será buena.

A continuación, se explican de forma concreta los detalles y los resultados de cada uno de los tres escenarios propuestos.

---

<sup>1</sup><http://www2.imse-cnm.csic.es/Xfuzzy/>

### Primera prueba

En este primer vídeo, se ha querido poner en práctica un entorno de tamaño medio. Como se puede apreciar en las gráficas de la Figura 3.6, el algoritmo *PTAM* lanza posiciones que varían de forma muy rápida.

En el caso de la gráfica de la posición en el eje  $x$ , se puede apreciar un desfase en el eje temporal. Este desfase se debe a las condiciones cambiantes de la ejecución de las dos pruebas. Las percepciones de *PTAM* comienzan dando una posición errónea excesivamente baja, a continuación corrige a su posición correcta (el primer pico alto), pero vuelve a bajar a dar posiciones erróneas. Después consigue volver a la posición correcta y dar valores que se ajustan más a la posición real.

En este caso, *FuzzyWeight* ha conseguido mantener mejor la posición  $x$  con el movimiento real, aunque se puede apreciar cómo la curva no es tan pronunciada, debido a que los valores erróneos de *PTAM* hacen que el peso de las percepciones buenas sea menor y le cueste más adaptarse.

De forma parecida ocurre con las gráficas del eje  $y$ , en las que consigue adaptarse mejor al movimiento original, aunque con un *offset* de  $-0,5$  metros aproximadamente, debido a que las percepciones iniciales de *PTAM* fueron muy bajas.

En el tercer caso, podemos contemplar un error en la gráfica de la posición  $Z$ . Como se puede apreciar, las posiciones previstas por el módulo *FuzzyWeight* en este eje son alrededor de un metro más bajas. Esto se debe a que al comienzo, *PTAM* lanzó unos valores más bajos (con los que se inicializó *FuzzyWeight*), y de repente comenzó a dar unos más altos. En ese momento, *FuzzyWeight* comenzó a dar poco peso a esos valores, ya que se trataba de *valores extremos*. Sin embargo, los valores que eran erróneos eran los iniciales.

Puesto que el peso que se le da a esas percepciones es menor, es necesario que pase más tiempo hasta que *FuzzyWeight* consiga retomar buenos valores. Como se puede apreciar, al final de la gráfica la diferencia entre los dos valores va siendo cada vez menor, como prueba de que las percepciones altas de *PTAM* es correcta.

Este efecto también se ve agravado por la incapacidad de *FuzzyWeight* de detectar movimiento en el eje  $z$ . Puesto que sólo tiene en cuenta el eje  $x$  y el eje  $y$ , cuando se da un movimiento principalmente en el eje  $z$ , *FuzzyWeight* lo interpreta como que no ha habido dicho movimiento, y por lo tanto a cualquier percepción de *PTAM* que indique que sí lo hay se le dará un peso pequeño, aún siendo dicho movimiento real.

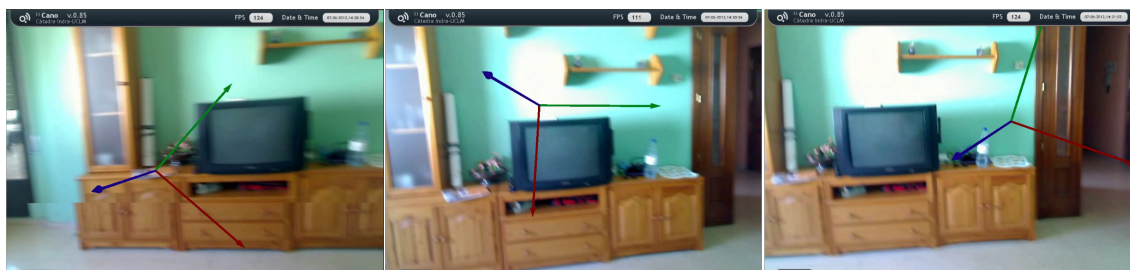


Figura 3.5: Screenshot del primer vídeo



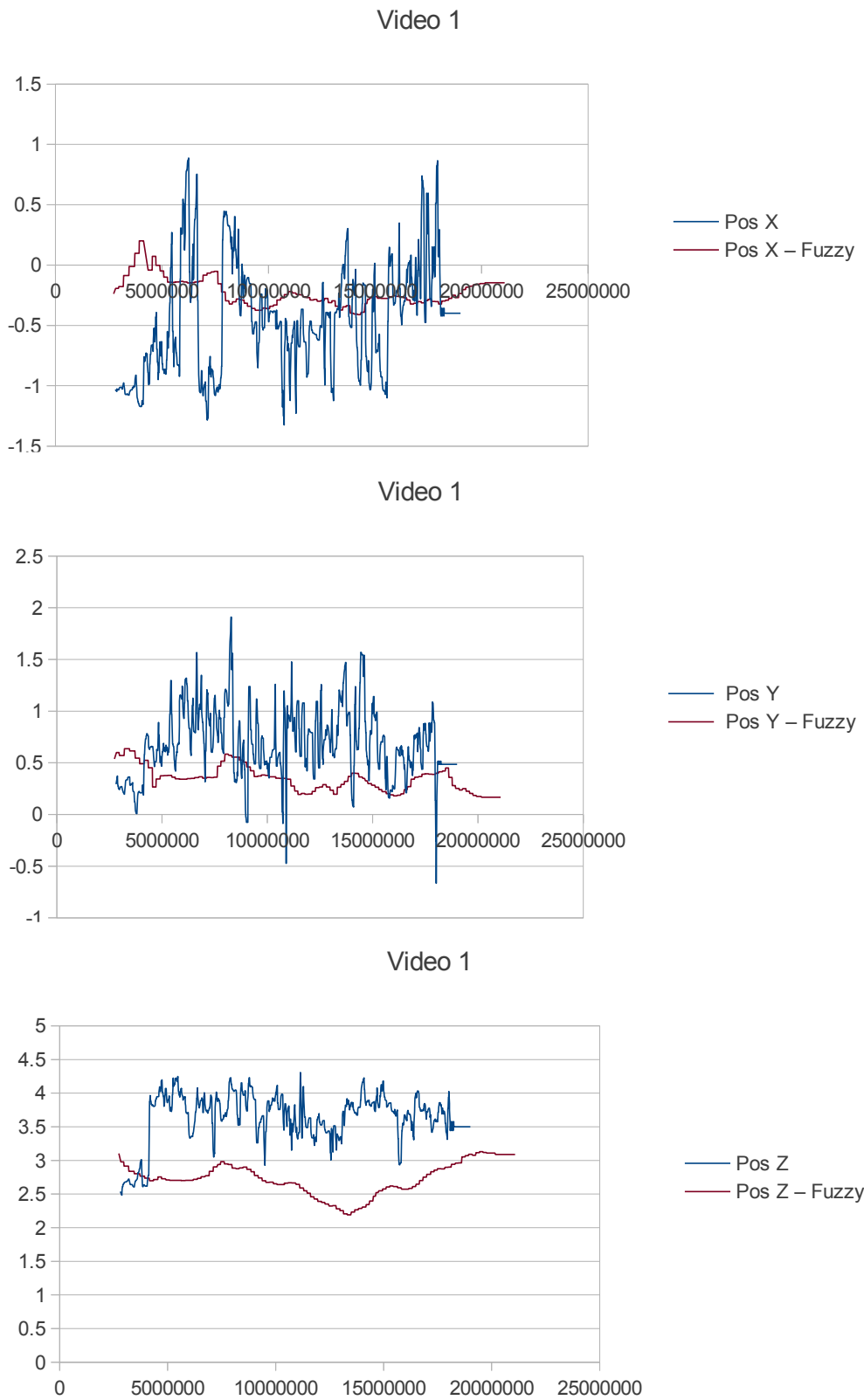


Figura 3.6: Resultados del escenario 1

### Segunda prueba

Para esta segunda prueba, se realizó la grabación del vídeo en un entorno reducido, sobre un escritorio de trabajo.

Este tracking en general ha sido muy bueno, y *FuzzyWeight* ha sido capaz de ajustarse con bastante precisión. Se puede apreciar un desfase temporal entre las dos ejecuciones (la gráfica de *FuzzyWeight* está ligeramente desplazada a la derecha).

No obstante, al inicio del vídeo, *PTAM* arrojó una percepción errónea muy alta en el caso del eje  $x$ , baja en el caso del eje  $y$ , y no demasiado alejada en el caso del eje  $z$ .

Se puede apreciar en el caso del eje  $x$  comienza dando por buenas las percepciones erróneas iniciales de *PTAM*, por lo que da valores bastante altos. Por lo tanto, cuando *PTAM* consigue recuperarse y dar los valores crecientes correctos, a *FuzzyWeight* le cuesta adaptarse a esos cambios, por lo que los desplazamientos son menos acusados.

De la misma forma se puede ver en la gráfica del eje  $y$ , que al comenzar con valores bajos, el crecimiento hacia los valores altos correctos es más lento que el de las percepciones de *PTAM*, al dar pesos medios-bajos.

En el caso de la gráfica del eje  $z$ , las percepciones durante todo el transcurso de la ejecución son muy similares, por lo que *FuzzyWeight* se limita a suavizarlas de forma sutil.

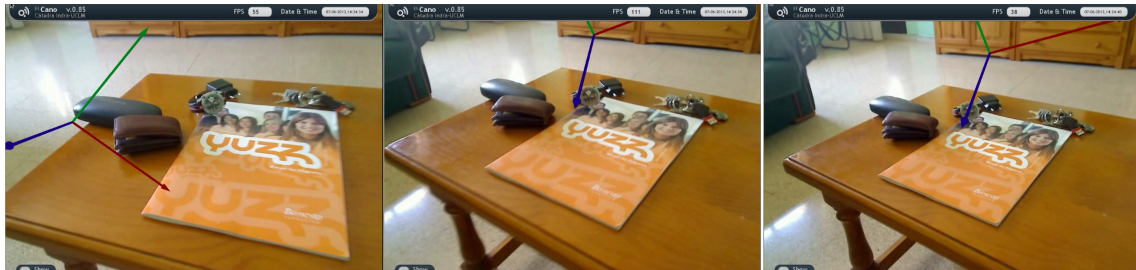


Figura 3.7: Screenshot del segundo vídeo

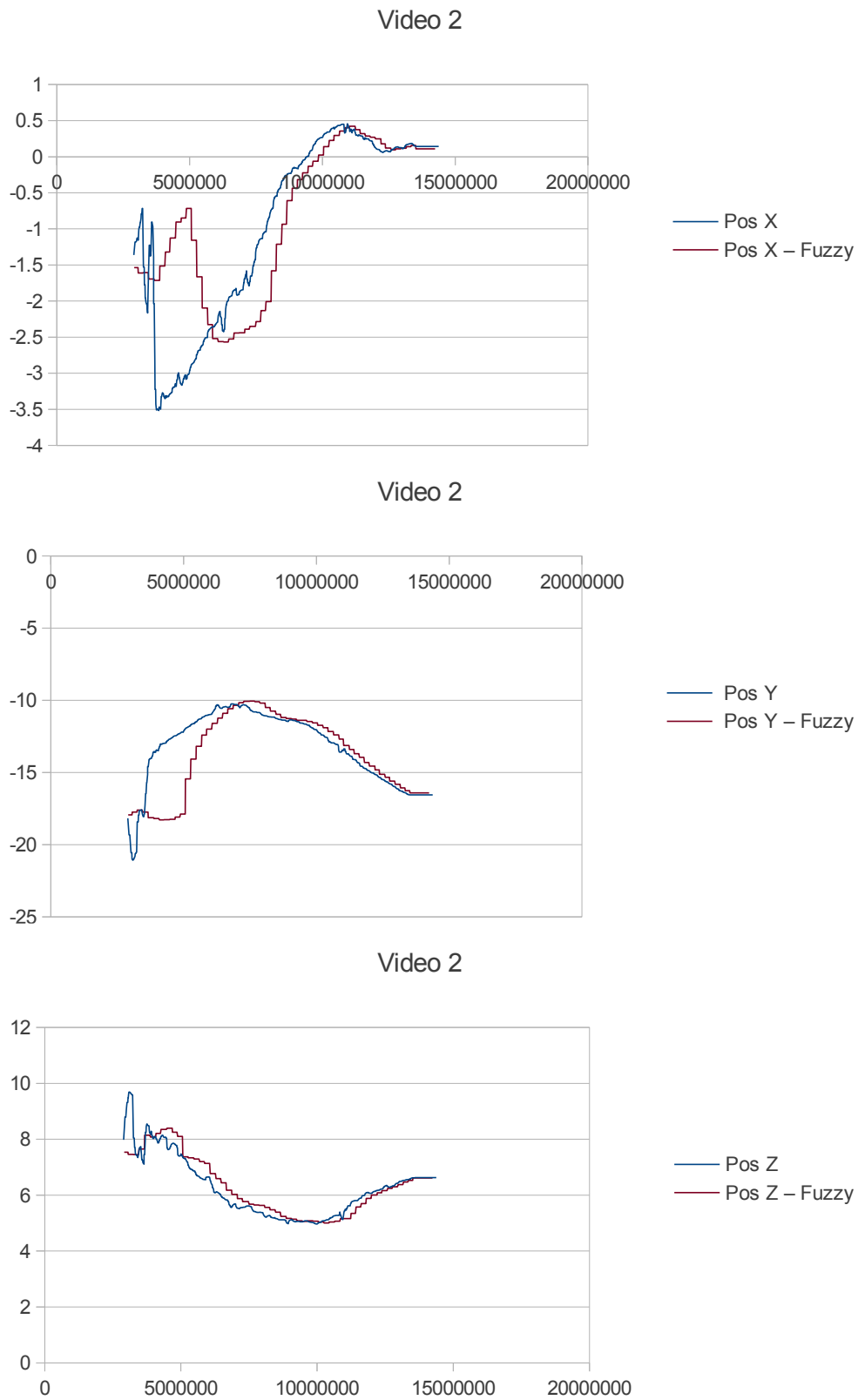


Figura 3.8: Resultados del escenario 2

### Tercera prueba

Este último ejemplo de prueba, se caracteriza por ser un entorno en el que el movimiento del usuario ha sido mayor y más brusco y la luminosidad del entorno muy pobre. Esto ha hecho que la calidad de las percepciones de *PTAM* hayan sido de igual manera de poca calidad. Aparte de esto, se puede apreciar como ventaja que no hay apenas desfase en las dos ejecuciones, por lo que se puede medir perfectamente la comparación entre las dos gráficas.

Como se ha dicho anteriormente, la mala calidad del entorno ha hecho que las respuestas de las percepciones de *PTAM* sean muy distantes. Se puede apreciar sobretodo en las gráficas del eje  $x$  y el eje  $y$  que cuenta con muchos picos muy distantes entre sí, los cuales son percepciones totalmente erróneas.

En el caso de la gráfica del eje  $x$ , vemos cómo *FuzzyWeight* consigue suavizar en gran medida estos *altibajos*, y crear un movimiento más uniforme. Sin embargo, el movimiento todavía está muy lejos de ser correcto, ya que existen desplazamientos demasiado bruscos, incluso con *FuzzyWeight* activado. En este eje, se pueden apreciar un par de pendientes descendentes bruscas incorrectas. Esto se debe a la mala calidad del *tracking* de *PTAM*, por lo que no se puede hacer mucho más.

Para la gráfica del eje  $y$  ocurre de forma parecida, aunque el movimiento es más real. Consigue paliar los efectos de las percepciones extremas erróneas y ajustarse al movimiento correcto. Sin embargo, se puede ver que en algunas partes la adaptación es más lenta debido a esos pesos bajos que se le da a las percepciones que comienzan a ser buenas.

En el eje  $z$ , los movimientos arrojados por *FuzzyWeight* son mucho más suaves que los originales de *PTAM* y se consiguen adaptar bastante bien, excepto en la parte final donde el brusco aumento de las percepciones de *PTAM* es interiorizado de forma más lenta por *FuzzyWeight*.

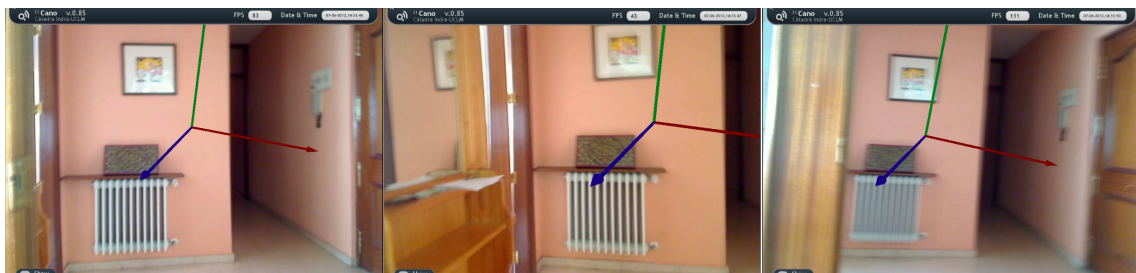


Figura 3.9: Screenshot del tercer vídeo

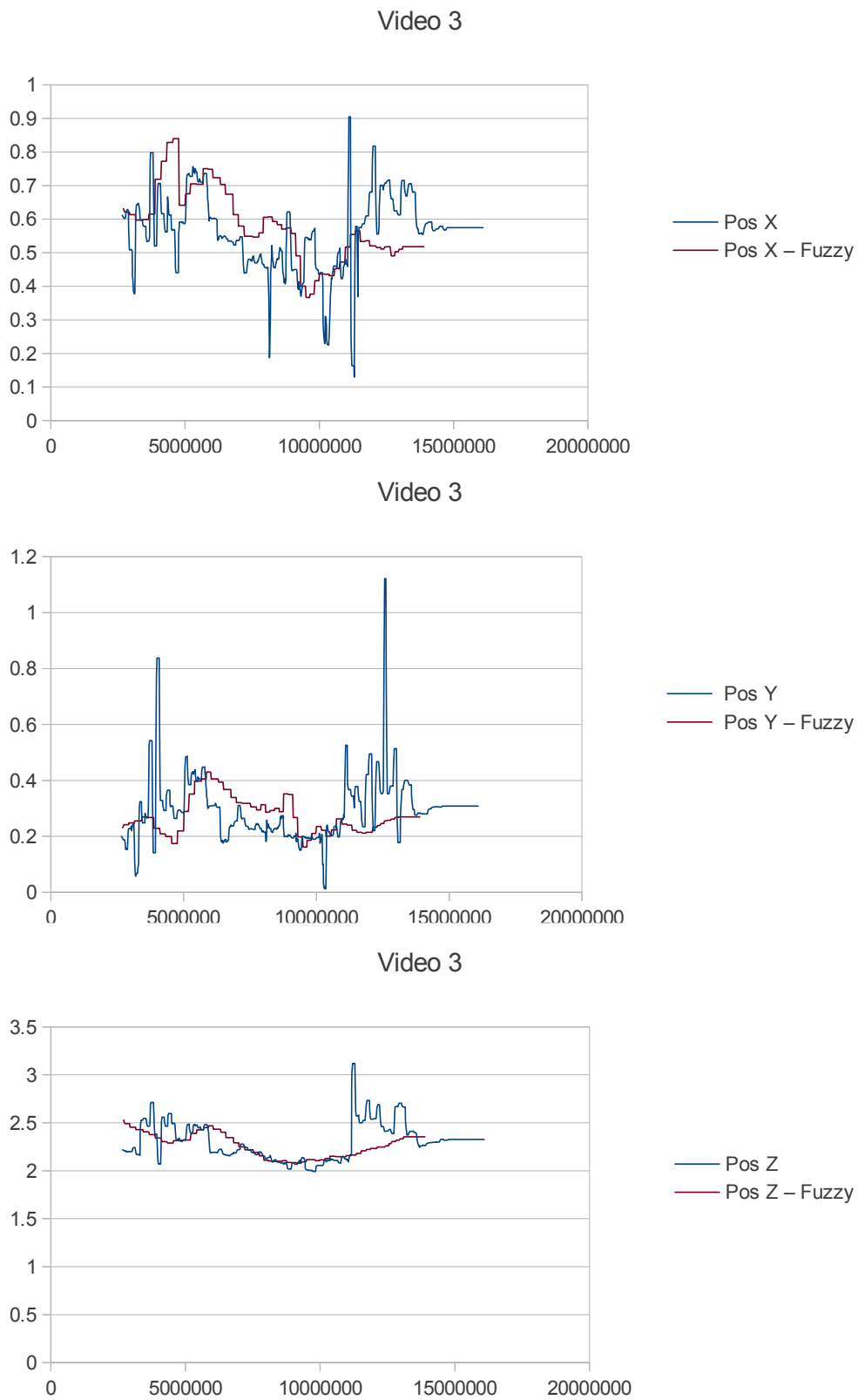


Figura 3.10: Resultados del escenario 3

---

# 4

## Conclusiones y trabajo futuro

---

En el presente documento, se ha propuesto y desarrollado un *Sistema multimodal de Navegación* para interiores enfocado a la asistencia a la movilidad de personas con necesidades especiales.

En primer lugar se realizó un análisis sobre los sistemas de posicionamiento existentes, sus virtudes y su problemática. El sistema de posicionamiento norteamericano *GPS* es ampliamente utilizado hoy en día para fines muy diversos. Este sistema consigue proporcionar la posición de un dispositivo a nivel mundial con un error menor de 5 metros. Sin embargo, el mayor inconveniente que presenta es la incapacidad de funcionar *en interiores*, debido a que necesita recibir la señal de varios satélites geoestacionarios, y dentro de los edificios esta señal es muy pobre.

Identificada la problemática, se realiza un estudio sobre las soluciones y estudios existentes sobre sistemas de posicionamiento en interiores. La mayoría de estos se basan en la detección de ondas electromagnéticas basadas en diversas tecnologías, como la *Wi-Fi* o la *RFID*. Sin embargo, estos sistemas alcanzan poca precisión debido a la gran variabilidad a la que está sometida dicha energía electromagnética, afectada por multitud de factores como la meteorología o el estado concreto de un edificio en un determinado momento. Además, estos sistemas no son capaces de proporcionar orientación del usuario, únicamente posición.

Otro tipo de sistemas son los basados en *giróscopos* y *acelerómetros*. Estos sistemas tienen un gran problema inherente a su naturaleza, ya que existe un error residual que se acumula con el tiempo, y llega a tomar unas dimensiones desproporcionadas.

Identificada esta problemática y el estado del arte existente, se propone la utilización de algoritmos y técnicas basados en Realidad Aumentada y Visión Artificial como solución al posicionamiento y orientación de usuarios dentro de edificios públicos, especializando su uso a la prestación de asistencia para personas con necesidades especiales.

El desarrollo de esta iniciativa se ha enmarcado dentro del proyecto ELCANO, financiado por la Cátedra Indra-UCLM, y llevado a cabo por los laboratorios de investigación Oreto y Arco de la Escuela Superior Informática de Ciudad Real. El proyecto ELCANO pretende, además de proporcionar un sistema de posicionamiento y navegación en interiores, ofrecer servicios y asistencia para la realización de distintas tareas dentro de edificios públicos para personas con necesidades especiales. El proyecto debe incorporar funcionalidades como síntesis de voz, listado y configuración de distintas tareas, o incluso asistencia por videoconferencia.

El sistema de navegación incorpora diferentes métodos de posicionamiento basados en Realidad Aumentada y Visión Artificial de distinta naturaleza, ejecutándose de forma simultánea, para determinar la posición del usuario. Para el desarrollo de la arquitectura, se han resuelto los siguientes problemas:

- **Estudio e incorporación de métodos de posicionamiento:** se han estudiado las diferentes alternativas de métodos de posicionamiento basados en Realidad Aumentada existentes. En total se han incorporado cinco métodos distintos a la arquitectura: uno basado en detección de marcadores, en puntos de interés, en descriptores visuales, en posicionamiento estéreo y *optical flow*.

Además, se ha debido diseñar una arquitectura modular para permitir la implementación y ejecución simultánea de cada uno de ellos.

- **Refinamiento inicial de la información:** las posiciones proporcionadas por los métodos se consideran *crudas*, o no refinadas, por lo que se ha implementado un módulo que realiza un refinamiento inicial. Para este módulo, se ha aplicado un algoritmo heurístico y otro basado en el histórico de percepciones.
- **Fusión de la información:** por último, para proporcionar una última posición final a partir de todos los métodos existentes, es necesario realizar una *integración*. Para esta fusión, es interesante maximizar la precisión y fiabilidad y minimizar el error. Se han utilizado modelos como las Redes Bayesianas y la Lógica Difusa para implementar esta fusión. Se optó por utilizar la lógica difusa y realizar un sistema que fusionase las percepciones del *optical flow* y el algoritmo basado en detección de puntos de interés.

Dado este sistema, se han realizado pruebas de integración, y gráficas contrastando el funcionamiento de los distintos módulos y métodos. El sistema consigue realizar posicionamiento, aunque tiene como requisito el realizar un entrenamiento de la zona donde se va desplegar. Este entrenamiento sólo necesita ser llevado a cabo una vez.

Actualmente, se considera que el sistema se encuentra en una fase madura, con una línea de investigación clara, habiendo sido puesto a prueba, por lo que se cuenta con resultado empíricos. De hecho, se está preparando un artículo para el *journal IEEE Pervasive Computing* (Impact Factor: 3.08) donde se explicarán las principales características y resultados de la arquitectura expuesta. Aún así, el sistema todavía seguirá madurando, como es intención de sus investigadores, siguiendo las líneas de trabajo futuro indicadas a continuación.

Como líneas de trabajo futuro, se proponen las siguientes:

- Los métodos de *tracking* tienen parámetros configurables que pueden aumentar o disminuir su rendimiento. Se pretende crear algoritmos que modifiquen estos parámetros de forma dinámica y se adapten a las condiciones del entorno. En concreto, se utilizará como base un algoritmo de adaptación dinámica del umbral propuesto por Ma et al [25].
- Mejora del módulo de fusión, para incorporar más entradas de métodos de *tracking* y aumentar la precisión de la posición final. Para ello, además de utilizar *PTAM* y *OpticalFlow* como entradas de *FuzzyWeight*, se añadirán el método de detección de patrones, el basado en descriptores visuales, y el posicionamiento estéreo.
- Implementar la arquitectura en dispositivos con menor capacidad de cómputo. Para ello, es necesario conocer las limitaciones de recursos y de interacción propias de estos dispositivos. Para afrontar esta portabilidad, se contará con el estudio de Gotow et al [8].
- Continuar estudiando las diferentes alternativas de métodos de posicionamiento visuales existentes, principalmente para conseguir más eficiencia. Un estudio interesante para dispositivos móviles es el de Park et al [30].





## Asignaturas cursadas

---

En este Apéndice se muestran todas las asignaturas cursadas por el alumno en el Máster de Tecnologías Informáticas Avanzadas impartido por la UCLM, haciendo un resumen de los conocimientos adquiridos en cada una de ellas.

### **Metodologías y Técnicas de Investigación en Informática**

- **Profesorado:** Mario Gerardo Piattini Velthuis, José Antonio Cruz Lemus y Marcela Genero Bocco.
- **Cuatrimestre:** Primer cuatrimestre.
- **Resumen:**

El objetivo de esta asignatura es inculcar al alumno los principales métodos y técnicas de investigación más adecuados para el contraste de las hipótesis de investigación en el ámbito de la informática. Estos conocimientos se encaminan también a la realización de la futura tesis doctoral del alumno.

En la asignatura se presenta el estado actual de la investigación de la informática, así como de todos los conceptos que son necesarios adquirir, el funcionamiento de los congresos, de los indicadores de calidad de los investigadores, estructuración de artículos o las revistas más importantes de estas áreas.

Toda la asignatura se desarrolló entorno a la creación de una Revisión Sistemática (SLR) sobre un tema propuesto por cada uno de los alumnos, mediante la cual se adquirirían las habilidades para buscar información en distintas fuentes específicas de

artículos, contrastar y seleccionar cada uno de los artículos (que se realiza con diferentes niveles de detalles, según la etapa en la que se encuentre la revisión), y considerar cada uno de forma crítica para la investigación personal del alumno.

En mi caso desarrollé una Revisión Sistemática sobre la literatura existente entorno a algoritmos de Realidad Aumentada para la detección de patrones en superficies no rígidas, y la interacción con el sistema mediante movimientos naturales con las extremidades o la cabeza.

## Generación de documentos científicos en informática

- **Profesorado:** José Antonio Gámez Martín, Francisco Parreño Torres y Luis de la Ossa Jiménez.
- **Cuatrimestre:** Primer cuatrimestre.
- **Resumen:**

El temario de esta asignatura, impartida en el Campus de Albacete, tenía tres bloques bien diferenciados:

1. **Bloque 1:** impartido por el profesor José Antonio Gámez, trata sobre el contexto del investigador. Explica de forma intuitiva el mundo de la investigación en informática, con conceptos básicos. Además se presentan técnicas para mejorar la escritura y las presentaciones ante una gran audiencia. También se enseñaban distintos indicadores existentes para cuantificar la calidad de los investigadores, en base a distintos factores. Al tratarse mi caso de la modalidad no presencial, tuve que realizar un trabajo para esta parte explicando un índice propuesto, el índice *H-barra*, basado en el índice *H*.
2. **Bloque 2:** impartido por el profesor Francisco Parreño Torres, enseña un *framework* para la realización de cálculos estadísticos, para contrastar de forma matemática distintos factores, como la calidad o la eficiencia, de los algoritmos propuestos por los estudiantes del máster. Como trabajo final de la asignatura, fue necesario hacer un trabajo sobre varios métodos de contraste estadísticos, y presentarlos de forma presencial en clase. Este trabajo fue realizado utilizando la herramienta de edición de documentos  $\LaTeX$ .
3. **Bloque 3:** este último bloque, impartido por el profesor Luis de la Ossa, enseñaba el uso de la herramienta  $\LaTeX$ , tanto para la creación de documentos como la de presentaciones, de forma profesional. El último trabajo de la asignatura se realizó utilizando esta herramienta.

## **Sistemas avanzados de interacción persona-computador: sistemas colaborativos y computación ubicua**

- **Profesorado:** Ana Isabel Molina Díaz, Miguel Ángel Redondo Duque y Crescencio Bravo Santos.
- **Cuatrimestre:** Primer cuatrimestre.
- **Resumen:**

El objetivo de esta asignatura fue la del aprendizaje de los distintos sistemas de interacción avanzados existentes, relacionados con la computación ubicua. Además, se presentaron distintas técnicas y *frameworks* para el desarrollo y análisis de estos sistemas.

El primer y segundo bloque, impartido por Miguel Ángel Redondo, enseñaba los fundamentos de la computación ubicua y los sistemas colaborativos, y hubo que realizar un trabajo en el que se podía escoger entre distintas temáticas. En mi caso, mi trabajo consistió en realizar un breve repaso sobre la evolución de los sistemas ubicuos a lo largo del tiempo. Para ello seleccioné diferentes artículos característicos de cada época en los últimos 20 años, describiendo cuál era la tendencia en cada uno de ellos.

El tercer bloque, que fue impartido por el profesor Crescencio Bravo, se estudiaban las bases teóricas para el diseño y desarrollo de sistemas colaborativos. Se presentaron distintos *frameworks* y herramientas como *JSDT*. Además, se estudió el caso real de una herramienta, COLLECE, a la que se le aplicaron los conocimientos adquiridos. El ejercicio asignado para este bloque consistió en análisis de dos herramientas, *Chess-Base* y *Facebook*, para identificar los elementos colaborativos que caracterizaban a cada uno, y finalmente proponer un modelo para el análisis de la colaboración basado en los apuntes vistos en clase.

El cuarto bloque, impartido por la profesora Ana Isabel Molina, explicaba los diferentes conceptos concernientes a la evaluación de los sistemas colaborativos, y la gran problemática asociada a ello. El trabajo para esta parte consistió en el análisis de dos métodos de evaluación de estos sistemas, el GHE (*Groupware Heuristic Evaluation*) y el GOT (*Groupware Observational Testing*), y una breve comparación entre ellos.

Por último, fue necesario realizar un trabajo final para la asignatura en la que se explicaba, a modo de repaso, todos los conocimientos adquiridos y trabajos realizados, y relacionarlos en la medida de lo posible con la investigación principal del alumno.

## Pruebas y seguridad de sistemas de información

- **Profesorado:** Macario Polo Usaola y Eduardo Fernández-Medina Patón

- **Cuatrimestre:** Segundo cuatrimestre.

- **Resumen:**

Esta asignatura, dividida en dos bloques principales, consiste en la introducción de diferentes conceptos relacionados con la seguridad y las pruebas de sistemas informáticos.

El primer bloque, impartido por el profesor Eduardo Fernández-Medina, consistió en la enseñanza de la parte de las metodologías existentes para llevar a cabo el diseño e implementación de sistemas de información en los que la seguridad es un aspecto vital de su desarrollo. El contenido se diferenciaba en aspectos como la seguridad en ingeniería del software, en bases de datos, y en la propia gestión de la seguridad. Para este bloque fue necesario realizar un trabajo, junto a su presentación, en el que se estudiaba y analizaba un artículo asignado. Mi trabajo consistió en explicar una metodología propuesta por Carlos Gutiérrez y David G. Rosado, para la elicitación y diseño de la seguridad de un sistema basado en *web services*.

El segundo trabajo de este bloque, consistió en analizar las técnicas de seguridad implementadas actualmente en un sistema de información actual. En mi caso, el trabajo y presentación se basó en las distintas medidas que se aplican para diseñar la seguridad de sistemas *Cloud Computing*.

El segundo bloque, impartido por Macario Polo, se centraba en los aspectos de las pruebas de los sistemas informáticos. Se aprendieron conceptos como pruebas de caja blanca frente a caja negra, valores de prueba y particiones, algoritmos de cobertura de valores de prueba, algoritmos de cobertura de código o pruebas basadas en mutantes. Se aprendieron a utilizar distintas herramientas para automatizar estas técnicas para el lenguaje Java.

Como trabajo para este último bloque se aplicaron técnicas de algoritmos de valores de pruebas y cobertura de código para un sistema de lógica difusa desarrollado en C para el proyecto ELCANO, por lo que además hubo que estudiar qué herramientas equivalentes existían para ese lenguaje.

## Técnicas de softcomputing

- **Profesorado:** Carlos González Morcillo, José Jesús Castro Sánchez y Juan Moreno García.
- **Cuatrimestre:** Segundo cuatrimestre.
- **Resumen:**

Esta asignatura, dividida en tres bloques, introduce el concepto de *Softcomputing* para la resolución de problemas demasiado complejos como para poder ser tratados de forma algorítmicamente tradicional. Estas técnicas se basan principalmente en el uso de la *lógica difusa*, por lo que se aprenden los conceptos de esta.

En el primer bloque, impartido por Carlos González, se enseñan las bases de la lógica difusa, los conjuntos difuso y los diferentes métodos de *fuzzificación* y *defuzzificación* existentes. Además se estudia una herramienta para la creación visual de estas herramientas, *XFuzzy*. Como trabajo para esta asignatura, se propone al estudiante crear un problema que pueda ser resuelto mediante la lógica difusa, y utilizar la herramienta *XFuzzy* para generar el código. Como trabajo, en mi caso se propuso un sistema difuso para estimar el peso de una percepción del algoritmo de posicionamiento *PTAM* utilizando como entrada la información de movimiento de un algoritmo de *Optical Flow*.

El segundo bloque, impartido por José Jesús Castro, explica el proceso general y las consideraciones a tener en cuenta a la hora de desarrollar un sistema difuso desde cero. Estas consideraciones se enfocaban en la definición de las reglas difusas y los conjuntos difusos que definirán el sistema.

El tercer bloque, impartido por el profesor Juan Moreno García, versaba sobre la inducción automática y el aprendizaje de los sistemas difusos, utilizando un conjunto de casos reales. De esta forma, aplicando distintos algoritmos, se pueden inferir de forma *dinámica* las reglas difusas a los que esos sistemas están sujetos.

Por último, como trabajo final para la asignatura, se aplicó el sistema difuso realizado en el primer bloque a una situación real, tomando mediciones y realizando distintas gráficas de su funcionamiento. También se realizó el planteamiento para el aprendizaje dinámico del sistema a medida que obtenía estimaciones de posicionamiento, identificando y descartando las erróneas.

## Cognición y Colaboración

- **Profesorado:** Manuel Emilio Prieto Méndez, María del Carmen Lacave Rodero y Jesús Serrano Guerrero.
- **Cuatrimestre:** Segundo cuatrimestre.
- **Resumen:**

Esta asignatura está dividida en tres bloques de temario:

- El primer bloque, impartido por el profesor Manuel Emilio Prieto, consistió en el aprendizaje de los diferentes modelos existentes para representar el conocimiento en base a la distinta naturaleza. Se mostraron las bases del aprendizaje humano, el aprendizaje automático o los modelos de árboles que se utilizan para tomar decisiones.
- El segundo bloque, impartido por José Ángel Olivas, explica las diferentes técnicas existentes para la recuperación inteligente de información. Se estudiaron técnicas basadas en *Softcomputing*, y todos los conceptos relacionados.
- El tercer y último bloque, impartido por Carmen Lacave, explicó conceptos de Redes Bayesianas para modelar el conocimiento probabilista, y las formas de inferirlo a partir de ellas. También se presentaba el programa Elvira para realizar este tipo de inferencias.

El trabajo final de la asignatura consistió en escoger un tema de los propuestos que se enmarcaba en uno de los tres bloques anteriores. Mi tema consistió en realizar un estudio sobre las aplicaciones existentes de las Redes Bayesianas a la Realidad Aumentada, concluyendo que son muy numerosas y variadas. Este trabajo fue de gran utilidad para la investigación principal del máster.



# Currículum

---

## Datos personales

- **Nombre:** César Mora Castro.
- **Titulación:** Ingeniero Superior Informático.
- **Email:** *cesarmoracastro@gmail.com*
- **Teléfono:** 656 978 247.
- **Fecha de nacimiento:** 18 de Febrero 1988.
- **Dirección:** C/ Lorenzo Godoy, nº 9 Villahermosa (Ciudad Real) C.P: 13332.

## Situación actual

- **2011-2012** Tecnólogo en el Grupo de Inteligencia Artificial Aplicada *Oreto* desarrollando el proyecto *ELCANO*: infraestructura para asistir la navegación en interiores a personas con necesidades especiales usando técnicas de Realidad Aumentada.
- **2011-2012** Cursando Máster en Tecnologías Informáticas Avanzadas en la Escuela Superior de Informática (Campus de Ciudad Real) por la Universidad de Castilla-La Mancha.
- **2012-2013** Cursar un Máster en Ingeniería Informática (*MS in Computer Science*) por la Universidad de Minnesota (Estados Unidos).

## Formación académica

- **2006-2011** Ingeniería Superior Informática por la Escuela Superior de Informática (Campus de Ciudad Real - UCLM). *Nota media: 7.9.*
- **2004-2006** Bachillerato en el Instituto de Educación Secundaria Francisco de Quevedo (Villanueva de los Infantes). *Matrícula de honor.*

## Experiencia laboral

- **2009-2012** Tecnólogo en el Grupo de Investigación Oreto. Trabajando en proyectos de Realidad Aumentada e Informática Gráfica sobre sistemas GNU/Linux.
- **2011-2012** Profesor en el Curso de Experto en Desarrollo de Videojuegos (Escuela Superior de Informática - Universidad de Castilla-La Mancha). Encargado de temas de diseño y creación de sistemas de partículas utilizando el motor de renderizado *Ogre3D*, y diseño e implementación de interfaces de usuario utilizando la biblioteca *CEGUI*.
- **2012** Ponente en las Conferencias de Accesibilidad de la Fundación Vodafone (Universidad de Alcalá - Alcalá de Henares , Madrid). Presentación del proyecto ELCANO durante las conferencias de la fundación.
- **2008-2010** Desarrollador del proyecto *Hesperia*: proyecto que desarrolló tecnología para el soporte a tareas de seguridad, vídeo vigilancia y operaciones de monitorización en infraestructuras y espacios públicos.
- **2008** Tecnólogo en el Grupo de Investigación Alarcos. Desarrollador de un portal web para facilitar la búsqueda de revistas para la publicación de artículos académicos.

## Cursos de formación y certificaciones

- **2012** TOEFL: Nota 94/120.
- **2012** GRE: Nota de *Quantitative Reasoning* 164/170.
- **2011-2012** Curso de emprendedores Desafío22 (fundación Bancaja).
- **2011-2012** Curso de emprendedores Yuzz (fundación Banesto).
- **2009-2011** Aprende Inglés en tu Campus (Universidad de Castilla-La Mancha)
- **2008** Curso de libre elección de Introducción a GNU/Linux Molinux.
- **2006-2007** Ciclo de Conferencias de la Escuela Superior de Informática (Campus de Ciudad Real).
- **2006** Curso cero física y matemáticas.



## **Becas y Premios**

- **2012** Beca por la Universidad de Castilla-La Mancha para estudiar un Máster en la Universidad de Minnesota (Estados Unidos) durante el curso 2012-2013.
- **2011** Premio al Mejor Proyecto de Fin de Carrera por la Cátedra Indra-UCLM por *Minerva: Sistema de especificación basado en Sensores, Controladores y Actuadores para aplicaciones de Realidad Aumentada*.

## **Idiomas**

- Español (nativo)
- Inglés (fluido escrito y hablado). TOEFL: 94/120.

# Bibliografia

---

- [1] R.T. et al. Azuma. A survey of augmented reality. *Presence-Teleoperators and Virtual Environments.*, 6(4):355–385, 1997.
- [2] G. Blasko and P. Fua. Real-time 3d object recognition for automatic tracker initialization. In *International Symposium on Augmented Reality.*, 2001.
- [3] C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the cave. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques.*, pages 135–142. ACM., 1993.
- [4] G. Dedes and A.G. Dempster. Indoor gps positioning. In *Proceedings of the IEEE Semi-annual Vehicular Technology Conference.*, volume 1, pages 1–5, 2005.
- [5] F. Evennou and F. Marx. Advanced integration of wifi and inertial navigation systems for indoor mobile positioning. *Eurasip journal on applied signal processing.*, 2006:164–164, 2006.
- [6] S. Feldmann, K. Kyamakya, A. Zapater, and Z. Lue. An indoor bluetooth-based positioning system: concept, implementation and experimental evaluation. In *International Conference on Wireless Networks.*, volume 1, pages 109–113, 2003.
- [7] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using bayesian networks to analyze expression data. *Journal of computational biology.*, 7(3-4):601–620, 2000.
- [8] J.B. Gotow, K. Zienkiewicz, J. White, and D.C. Schmidt. Addressing challenges with augmented reality applications on smartphones. *Mobile Wireless Middleware, Operating Systems, and Applications.*, 1:129–143, 2010.
- [9] D. Hallaway, S. Feiner, and T. Höllerer. Bridging the gaps: Hybrid tracking for adaptive mobile augmented reality. *Applied Artificial Intelligenc.e*, 18(6):477–500, 2004.
- [10] S.J. Henderson and S. Feiner. Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turre.t. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium.*, pages 135–144, 2009.
- [11] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global positioning System. Theory and Practice.* Springer., 1993.
- [12] GPS Inside and L. Edge. Centimeter-accuracy indoor navigation using gps-like pseudolites., 2001. Online article of GPS World.

- [13] F.V. Jensen. *An introduction to Bayesian networks.*, volume 36. UCL press London., 1996.
- [14] B. Jiang, U. Neumann, and S. You. A robust hybrid tracking system for outdoor augmented reality. In *Virtual Reality, 2004. Proceedings.*, volume 1, pages 3–10. IEEE., 2004.
- [15] E. Kaiser, A. Olwal, D. McGee, H. Benko, A. Corradini, X. Li, P. Cohen, and S. Feiner. Mutual disambiguation of 3d multimodal interaction in augmented and virtual reality. In *Proceedings of the 5th international conference on Multimodal interfaces.*, volume 1, pages 12–19. ACM., 2003.
- [16] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop.*, pages 85–94. IEEE., 1999.
- [17] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. 99:85–94, 1999.
- [18] H. Kaufmann and D. Schmalstieg. Mathematics and geometry education with collaborative augmented reality. *Computers & Graphics.*, 27(3):339–345, 2003.
- [19] N. Kawasaki and Y. Takai. Video monitoring system for security surveillance based on augmented reality. In *Proceedings of the 12th International Conference on Artificial Reality and Telexistence.*, volume 12, pages 4–6, 2002.
- [20] F. Kishino and P. Milgram. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems.*, 1(12):1321–1329, 1994.
- [21] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *Mixed and Augmented Reality, 2009. ISMAR 2009. 8th IEEE International Symposium.*, volume 1, pages 83–86. IEEE., 2009.
- [22] G.J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic.* Prentice Hall New Jersey., 1995.
- [23] P. Lang, A. Kusej, A. Pinz, and G. Brasseur. Inertial tracking for mobile augmented reality. In *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE.*, volume 2, pages 1583–1587. IEEE., 2002.
- [24] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions.*, 37(6):1067–1080, 2007.
- [25] X. Ma, G. Shi, and H. Tian. Adaptive threshold algorithm for multi-marker augmented reality system. In *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry*, pages 71–74. ACM, 2010.
- [26] P. Misra and P. Enge. *Global positioning system: Signals, measurements and theory.*, 2011. Winter course of the School of Engineering of Schulich.
- [27] U. Neumann and S. et al. You. Modeling and video projection for augmented virtual environments., 2009.
- [28] T. Nguyen, C. Sandor, and J. Park. Ptamm-plus: Refactoring and extending ptamm. *Camera.*, 1:84–88, 2010.

- [29] L. Ojeda and J. Borenstein. Flexnav: Fuzzy logic expert rule-based position estimation for mobile robots on rugged terrain. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference.*, volume 1, pages 317–322. IEEE., 2002.
- [30] N. Park, W. Lee, and W. Woo. Barcode-assisted planar object tracking method for mobile augmented reality. In *Ubiquitous Virtual Reality (ISUVR), 2011 International Symposium.*, pages 40–43. IEEE., 2011.
- [31] S. Park and J.K. Aggarwal. A hierarchical bayesian network for event recognition of human actions and interactions. *Multimedia Systems.*, 10(2):164–179, 2004.
- [32] W. Piekarski and B. Thomas. Arquake: the outdoor augmented reality gaming system. *Communications of the ACM.*, 45(1):36–38, 2002.
- [33] M.A. Quddus, R.B. Noland, and W.Y. Ochieng. A high accuracy fuzzy logic based map matching algorithm for road transport. *Journal of Intelligent Transportation Systems.*, 10(3):103–115, 2006.
- [34] C. Randell and H. Muller. Low cost indoor positioning system. In *Ubicomp 2001: Ubiquitous Computing.*, volume 1, pages 42–48. Springer., 2001.
- [35] H. Regenbrecht, T. Lum, P. Kohler, C. Ott, M. Wagner, W. Wilke, and E. Mueller. Using augmented virtuality for remote collaboration. *Presence: Teleoperators & Virtual Environments.*, 13(3):338–354, 2004.
- [36] H. Regenbrecht, C. Ott, M. Wagner, T. Lum, P. Kohler, W. Wilke, and E. Mueller. An augmented virtuality approach to 3d videoconferencing. In *Mixed and Augmented Reality, 2003. Proceedings. The Second IEEE and ACM International Symposium.*, pages 290–291, 2003.
- [37] M. et al. Rosenthal. Augmented reality guidance for needle biopsies: An initial randomized, controlled trial in phantoms. *Medical Image Analysis.*, 6(3):313–320, 2002.
- [38] N.E. Seymour, A.G. Gallagher, S.A. Roman, M.K. O'Brien, V.K. Bansal, D.K. Andersen, and R.M. Satava. Virtual reality training improves operating room performance: results of a randomized, double-blinded study. *Annals of surgery.*, 236(4):458–464, 2002.
- [39] J. Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of communication.*, 42(4):73–93, 1992.
- [40] M. Vögele. Augmented reality: projectors.
- [41] L.A. Zadeh. Fuzzy logi.c. *Computer.*, 21(4):83–93, 1988.