



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

**INFERENCIA DE ACTIVIDADES
EN ENTORNOS
MULTIOCUPACIÓN CON
RECONOCIMIENTO BASADO EN
VISIÓN POR COMPUTADOR**

Alumna: María del Carmen Quesada Real

Tutora: Prof.^a D.^a Silvia Satorres Martínez
Dpto: Ingeniería Electrónica y Automática

Tutora: Prof.^a D.^a Macarena Espinilla Estévez
Dpto: Informática

Junio, 2017



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Proyecto Fin de Carrera

**INFERENCIA DE ACTIVIDADES
EN ENTORNOS
MULTIOCCUPACIÓN CON
RECONOCIMIENTO BASADO EN
VISIÓN POR COMPUTADOR**

Firma:

Firmas:

Alumna: M^a Carmen Quesada Real

Tutora: Prof.^a D.^a Silvia Satorres Martínez

Tutora: Prof.^a D.^a Macarena Espinilla Estévez

Dpto: Ingeniería Electrónica y Automática

Dpto: Informática

Junio, 2017

Índice general

Índice de figuras	3
Índice de tablas	5
1. Introducción	7
1.1. Motivación	7
1.2. Propuesta	8
1.3. Objetivos	8
1.4. Estructura de la memoria	8
2. Antecedentes	11
2.1. Ambientes Inteligentes	11
2.1.1. Sensores	11
2.1.2. Actuadores	12
2.1.3. Procesamiento de información	12
2.2. Lógica Difusa	12
2.2.1. Introducción a la Lógica Difusa	13
2.2.2. Conjuntos Difusos	16
2.2.3. Variables lingüísticas	19
2.2.4. Razonamiento aproximado. Reglas difusas	21
3. Estado del arte	25
3.1. Entorno inteligente de estudio	25
3.2. Elementos que intervienen. Sensores	27
3.2.1. Sensores de visión	27
3.2.2. Sensores de proximidad	28
3.2.3. Sensores de apertura y cierre	29
3.2.4. Sensores de movimiento	29
3.2.5. Otros sensores	31
3.3. Reconocimiento mediante visión por computador	31

3.3.1. <i>ArUco marker</i>	32
3.3.2. <i>Vuforia frame marker</i>	35
4. Metodología	37
4.1. Adquisición de datos	38
4.2. Procesado de datos	39
4.2.1. Tratamiento imágenes. Reconocimiento de marcadores.	39
4.2.2. Tratamiento resto de datos	40
4.3. Términos lingüísticos empleados	40
4.4. Formulación de reglas. Inferencia de actividades	43
5. Evaluación	49
5.1. Desarrollo de escenas y presentación de resultados	49
5.1.1. Escena 1	49
5.1.2. Escena 2	53
5.1.3. Escena 3	57
5.1.4. Escena 4	60
5.1.5. Escena 5	62
5.1.6. Escena de pruebas.	65
5.2. Discusión de resultados	66
6. Conclusiones	69
6.1. Conclusiones finales	69
6.2. Trabajos futuros	72
A. Anexos	75
A.1. Estructura de trabajo	75
A.2. Programación	76
A.2.1. <i>Function block</i> para fuzzificación de datos.	76
A.2.2. Datos temporales	77
A.2.3. Datos distancia	79
A.3. Presentación de los datos obtenidos	82
A.3.1. Sensor visión etapa 1	82
A.3.2. Sensor proximidad	83
A.3.3. Sensor apertura y cierre y movimiento	84
Bibliografía	87

Índice de figuras

3.1. Plano del <i>SmartLab</i>	25
3.2. Espacio dormitorio y baño	26
3.3. Espacio salón	26
3.4. Espacio cocina	27
3.5. Sensor de visión 1	27
3.6. Sensor de visión 2	28
3.7. <i>Estimote proximity Beacons</i>	28
3.8. Sensor <i>Smartsense Open/Closed</i>	29
3.9. Sensor <i>Smartsense Motion</i>	30
3.10. Fibaro sensor de movimiento	30
3.11. <i>SmartThings Hub</i>	31
3.12. Marcador ArUco	32
3.13. Procesado de imagen	35
3.14. <i>Vuforia frame marker</i>	36
4.1. <i>Grafcet</i>	38
4.2. Entorno web de monitorización.	39
4.3. Función de pertenencia variable lingüística <i>isClose</i>	41
4.4. Función de pertenencia variable lingüística <i>Recently</i>	42
4.5. Función de pertenencia variable lingüística <i>For a while</i>	42
4.6. Función de pertenencia variable lingüística <i>For a long while</i>	43
5.1. Ejemplo de imágenes captadas durante la escena 1.	50
5.2. Escena 1: Cuarto.	51
5.3. Escena 1: Cocina.	51
5.4. Escena 1: Salón.	52
5.5. Escena 1: Baño.	53
5.6. Escena 1: Reglas.	53
5.7. Muestra de imágenes captadas durante la segunda escena.	54
5.8. Escena 2: Cuarto.	54

5.9. Escena 2: Cocina.	55
5.10. Escena 2: Baño.	56
5.11. Escena 2: Reglas.	56
5.12. Ejemplo de imágenes captadas durante la escena 3.	57
5.13. Escena 3: Reglas	57
5.14. Escena 3: Reglas Agente 1.	58
5.15. Escena 3: Reglas Agente 2.	59
5.16. Muestra de imágenes captadas durante la escena 4.	60
5.17. Escena 4: Reglas.	60
5.18. Escena 4: Reglas Agente 1.	61
5.19. Escena 4: Reglas Agente 2.	62
5.20. Muestra de imágenes captadas durante las escenas 5 y 6.	63
5.21. Escena 5: Reglas.	63
5.22. Escena 5: Reglas Agente 1.	64
5.23. Escena 5: Reglas Agente 2.	65
5.24. Imágenes captadas durante la escena de pruebas.	66
A.1. Estructura de trabajo.	75
A.2. Presentación datos sensores visión.	82
A.3. Fotograma 229.	83
A.4. Capturas de datos recibidos por los <i>eBeacons</i>	84
A.5. Array de datos de apertura y cierre y de movimiento.	85

Índice de tablas

4.1. Sensores que intervienen en el estudio.	44
4.2. Reglas establecidas del análisis del estudio.	47

Capítulo 1

Introducción

1.1. Motivación

Desde que nació el concepto de *inteligencia artificial* [3], y dentro de ésta los *sistemas expertos*[14] como sistemas computacionales con la capacidad de tomar decisiones de un humano experto, ha habido un incesante auge de las tecnologías de la información y la comunicación.

De la *inteligencia artificial* junto con otras disciplinas como la informática, electrónica, etc., se sirven los *Ambientes Inteligentes* [4].

En definitiva, este tipo de ambientes representan un nuevo paso en la evolución en la automatización de los sistemas industriales y domésticos(caso de este estudio).

Por otro lado, el avance de la medicina así como el incremento de la esperanza de vida de la población ha favorecido un aumento de enfermedades como la *demencia*. Diferentes estudios [2, 9] y ejemplares bibliográficos [13] muestran cómo este tipo de conductas siguen patrones preestablecidos, analizables con un sistema entrenado para que realice una determinada actuación.

Asimismo, otra serie de afecciones como puede ser el *autismo* [15], también muestra una serie de comportamiento definidos en momentos de crisis.

El presente proyecto tiene por objeto principal realizar el reconocimiento de actividades de varios ocupantes en un ambiente inteligente, partiendo de datos obtenidos de los sensores del entorno, para así adaptar dicho ambiente a las necesidades de los ocupantes.

La elección del tema se debe a dos circunstancias fundamentales: la primera de ellas es la posibilidad de ampliar mis conocimientos tanto en el campo de la informática como de la visión por computador; la segunda, y no menos importante, poder contribuir al desarrollo de un estudio que posibilite un avance para algunos grupos de la sociedad.

El tratamiento de datos se ha realizado gracias a la información recogida mediante los

sensores que se encuentran en el laboratorio inteligente del CEATIC¹ de la Universidad de Jaén.

Además, la temática posee cierta relación con la intensificación en Automatización Industrial y Robótica. De manera más concreta, con el tratamiento de imágenes obtenidas mediante sensores de visión y con la automatización en edificios.

Este proyecto tiene un gran fundamento teórico aunque, por otro lado la temática ha posibilitado la experimentación de diferentes escenarios para un posterior estudio de posibles actuaciones que los agentes pueden realizar en el ambiente.

1.2. Propuesta

Este proyecto final de carrera tiene como propósito el estudio de un ambiente para que, a partir de la obtención de datos por medio de los sensores integrados en el entorno, se realice el reconocimiento de las actividades que desempeñen varios ocupantes con el objetivo de adaptar dicho ambiente a las necesidades de los mismos mediante un proceso de inferencia y razonamiento.

1.3. Objetivos

Los objetivos que se definen a través de la propuesta de este proyecto son:

1. Revisión del estado actual del entorno inteligente.
2. Análisis y diseño del sistema de control.
3. Desarrollo del software.
4. Entrenamiento del sistema de control.
5. Evaluación de los resultados obtenidos.

1.4. Estructura de la memoria

A continuación, se describen los diferentes capítulos que albergará este Proyecto Fin de Carrera:

¹Centro de Estudios Avanzados en Tecnologías de la Información y la Comunicación. Es un centro de la Universidad de Jaén que da apoyo a la excelencia investigadora y docente, impulsando también la transferencia del conocimiento. A partir de una serie de medios ubicados en sus instalaciones, se permite el avance del conocimiento, desarrollo e innovación, en el campo de las tecnologías de la información y la comunicación.

- **Capítulo 1.** En este capítulo, se expone la motivación del proyecto, la propuesta específica a la que se pretende llegar y los objetivos del mismo.
- **Capítulo 2 .** En este capítulo se ha realizado una introducción teórica de las dos temáticas principales y esenciales de este trabajo:
 - Los **Ambientes Inteligentes**. Consta de una breve descripción de este tipo de ambientes y de los sensores que en ellos intervienen.
 - La **Lógica Difusa**. De manera exhaustiva y para mayor facilidad, se ha querido profundizar en este apartado para hacer entender al lector por qué se ha utilizado, así como las ventajas de la misma con respecto a la lógica tradicional.
Para ello, se ha visto necesario contextualizar y definir una serie de conceptos básicos para comprenderla.
- **Capítulo 3.** Es el capítulo en el que se describe el entorno en el que se realiza el estudio, los elementos que intervienen en el ambiente inteligente y los marcadores utilizados así como otros marcadores de gran uso en aplicaciones de reconocimiento mediante visión por computador.
- **Capítulo 4.** Es el capítulo en el que se presenta la metodología de trabajo seguida. En él se plantea la adquisición de los datos y su tratamiento, cómo se realiza y qué aplicaciones se han utilizado. Por otro lado, se definen los diferentes términos lingüísticos empleados para el procesado.
Como último punto de este capítulo, se establecen las reglas para el reconocimiento de actividades que ejecutan en el ambiente.
- **Capítulo 5.** En este capítulo, se define el desarrollo de las escenas y se evalúan los resultados.
- **Capítulo 6.** En este capítulo final, se presentan tanto las conclusiones adquiridas del estudio como las posibles líneas de investigación para trabajos futuros.
- **Capítulo Anexos.** Se han presentado los bloques de programación necesarios para la fuzzificación de los datos y la forma en la que se obtienen los datos.

Capítulo 2

Antecedentes

En este capítulo se van a definir teóricamente los conceptos de *Ambientes Inteligentes* y *Lógica Difusa*.

2.1. Ambientes Inteligentes

De entre la gran cantidad de definiciones que podemos encontrar sobre los ambientes inteligentes, encontramos la de Alan Steventon y Steve Wright [18]. Según ellos los “*ambientes inteligentes son sistemas en los que la computación es usada para introducir mejoras imperceptibles o superficiales en las actividades comunes*”.

Dicho de otro modo, los ambientes inteligentes son espacios que usan la tecnología de sistemas embebidos así como otras tecnologías de la información y la comunicación, para crear ambientes interactivos que acerquen la computación al mundo físico y a los problemas cotidianos.

Lo que se pretende conseguir con este tipo de sistemas es que sensores y computadores participen en actividades y resultados posibilitando a los usuarios interactuar con los distintos dispositivos. Éste es el punto de partida para conseguir la adaptación del ambiente a las necesidades de cada agente que intervenga en dicho entorno.

Para que haya una buena conducta del ambiente con el usuario, el entorno debe ser educado según las preferencias del agente para que los servicios implícitos a su disposición se adapten a sus actividades.

Esta retroalimentación implícita es una estrategia de gran valor en entornos que no estén expuestos a frecuentes errores en los sensores en el ambiente, pero de gran interés para sistemas que se apoyen en un buen modelo de usuario y de contexto [7].

2.1.1. Sensores

Los entornos inteligentes, definidos en la sección 2.1, confían en datos sensoriales provenientes de múltiples sensores.

Los sensores [1, 12] son dispositivos que obtienen información partiendo de un fenómeno ambiental. Dichos sensores añaden valor a los datos para dar soporte a la toma de decisiones y al procesamiento distribuido.

Estos elementos recogen información de distintos parámetros que controlan y transmiten dichos datos para su procesamiento. Dependiendo del tipo de magnitud que se quiera controlar, encontramos en el mercado una gran variedad de sensores.

Las baterías de larga duración que incorporan estos elementos evitan que tengan que conectarse a la red eléctrica, lo que ofrece una gran flexibilidad de instalación.

Como ejemplo tenemos los sensores de presencia, utilizados para que controlan la iluminación de estancias o detectar presencias no deseadas.

2.1.2. Actuadores

Un actuador es un dispositivo capaz de recibir una orden procedente de un sistema de control y realizar una acción que modifique el estado de un determinado equipo y/o instalación. Es decir, transforma una energía, del tipo que sea, en la activación de un proceso con la finalidad de generar un efecto sobre un proceso automatizado.

Un ejemplo puede ser la apertura de una electroválvula para la activación de la calefacción o aire acondicionado al llegar a un determinado valor de temperatura.

2.1.3. Procesamiento de información

Los sensores son un pilar de los ambientes inteligentes, recogen la información. Después hay que procesar la información para obtener conclusiones y poder realizar una acción sobre el ambiente.

2.2. Lógica Difusa

La lógica difusa proporciona un mecanismo de inferencia que permite simular los procedimientos de razonamiento humano en sistemas basados en el conocimiento. La teoría de la lógica difusa proporciona un marco matemático que permite modelar la incertidumbre de los procesos cognitivos humanos de forma que pueda ser tratable por un computador. En este primer capítulo se describirán los fundamentos y características de este mecanismo de representación de la incertidumbre.

La inferencia es una evaluación que realiza la mente entre proposiciones. La inferencia es la acción y efecto de inferir

2.2.1. Introducción a la Lógica Difusa

Como ya sabemos, la dialéctica humana posee grandes habilidades para comunicar hechos empleando diferentes reglas lingüísticas. Dichas reglas, por lo general, son vagas y representan instrucciones, pudiendo ser fácilmente entendibles por un humano.

Sin embargo, la lógica convencional no es adecuada para procesar este tipo de reglas ya que éstas son difícilmente representables en un lenguaje que pueda interpretar un ordenador. Para ello habría que definir fronteras que no resulten descripciones artificiales debido al uso de intervalos discretos.

Queda claro que el conocimiento experto presenta a menudo, características de vaguedad e imprecisión, debido a tres razones principalmente:

1. **Ignorancia práctica:** incluso conociendo todas las variables, puede ser difícil obtener datos concretos asociados para su estudio. Además, esta información puede estar incompleta, e incluso ser errónea.
2. **Ignorancia teórica:** en el caso de no conocer un método teórico que modele el problema por falta de factores a tener en cuenta.
3. **Pereza:** obtener una lista completa de todas las variables que intervienen en el dominio del problema puede suponer un sobreesfuerzo. Además, como el mundo real es no determinista, hay veces que no es posible establecer completamente todas las variables del entorno.

Tratamiento de la incertidumbre

Se pueden encontrar multitud de enfoques para tratamiento de la incertidumbre. Las primeras aproximaciones responden a un tratamiento de la información puramente probabilista. Los primeros sistemas expertos modelaron el conocimiento con un enfoque puramente lógico, con las limitaciones que conlleva este tipo de enfoques. La siguiente generación de sistemas expertos emplearon técnicas probabilísticas con prometedores resultados. El principal problema de la aproximación fue el cálculo de la distribución conjunta de probabilidad para un número elevado de variables ya que suponía un crecimiento exponencial de las probabilidades. De esta forma, surgieron otras aproximaciones:

Métodos no numéricos. Algunas aproximaciones no numéricas utilizan un razonamiento cualitativo, mucho más cercano al humano. Unos de los métodos más estudiados en esta categoría es el *razonamiento por defecto*, que trata las conclusiones de los sistemas de reglas como válidas hasta que se encuentre una razón mejor para creer lo

contrario. Otros ejemplo de este tipo de métodos son las redes cualitativas y los sistemas de mantenimiento de coherencia.

Métodos numéricos. Son las familias de métodos probabilistas que asocian un valor numérico, *grado de creencia*, entre un intervalo para resumir la incertidumbre entre oraciones. Las creencias dependen de las percepciones recibidas por el agente inteligente hasta el momento, que constituyen la evidencia sobre la que se hacen las afirmaciones sobre probabilidades. Las probabilidades pueden cambiar al adquirir mayor número de evidencias.

Existen varias familias de técnicas probabilísticas entre las que distinguimos los métodos exactos (redes bayesianas, diagramas de influencia..) y los aproximados (método bayesiano subjetivo, factores de certeza..).

Otros métodos numéricos no probabilistas para el tratamiento de la incertidumbre son la **teoría de Dempster-Shafer** que utiliza grados de creencia dados por intervalos de valores para representar el conocimiento; La **Lógica Difusa** es también un método de razonamiento aproximado no probabilista. Su éxito se debe principalmente a la posibilidad de resolver problemas de una gran complejidad y poco definidos que, mediante métodos tradicionales, son difíciles de solucionar.

Lógica Difusa

La **Lógica Difusa** puede definirse como una extensión de la lógica multievaluada que permite representar matemáticamente la incertidumbre y la vaguedad, proporcionando herramientas formales para su tratamiento. Como indica Zadeh¹ [21], “*Cuando aumenta la complejidad, los enunciados precisos pierden su significado y los enunciados útiles pierden precisión*”.

Básicamente, la lógica difusa permite establecer un mapeo entre un conjunto de variables de entrada dado y la obtención de un valor adecuado de variables de salida de una forma adecuada, atendiendo a criterios de significado.

Características

El Principio de Incompatibilidad dice que la descripción del comportamiento de un sistema complejo no puede realizarse de forma absolutamente precisa. Para solucionar este problema Zadeh [20] plantea la necesidad de obtener herramientas capaces de manejar de forma rigurosa y fiable información imprecisa, lo cual obliga a desarrollar dos aspectos:

¹Lotfi A. Zadeh, catedrático de la Universidad de Berkeley (California), es un matemático, ingeniero eléctrico e informático, que en 1965 introdujo los *conjuntos difusos* y la *lógica difusa*. Es considerado padre de la Teoría de la Posibilidad.

- **Inferencia sobre información imprecisa:** Se necesita una forma de combinar esta información para obtener nuevos hechos. Por ello, Zadeh establece la necesidad de un método de inferencia generalizado e introduce lo que se conoce como *Regla Composicional de Inferencia* [19].
- **Representación de la información imprecisa:** Para esto se propone el empleo de la *Teoría de conjuntos difusos* así como describir la experiencia de los sistemas complejos en sus relaciones entrada-salida mediante proposiciones condicionales del tipo *Si-Entonces* ligando de esta manera las variables de entrada y salida.

A partir de este principio, se describen las principales características esenciales de la lógica difusa y los sistemas difusos:

1. El razonamiento exacto puede verse como un caso particular del razonamiento aproximado. Cualquier sistema lógico puede ser *fussificado*. Mediante lógica difusa se puede formular el conocimiento humano de una forma sistemática, y puede ser fácilmente incluido en sistemas de ingeniería.
2. El conocimiento se interpreta como una colección de restricciones difusas sobre una colección de variables. Los sistemas difusos son especialmente interesantes para la definición de sistemas cuyo modelo exacto es difícil de obtener.
3. La inferencia puede verse como un proceso de propagación de estas restricciones difusas.
4. Se utiliza ampliamente en sistemas de ayuda a la decisión. La lógica difusa permite obtener decisiones con valores incompletos o información incierta.

Los sistemas difusos son muy recomendables en aquellos problemas complejos donde no existe asociado un modelo matemático simple así como procesos que obedecen a un comportamiento no lineal. La solución difusa requiere que el conocimiento experto sea expresado lingüísticamente, requisito que es normalmente fácil de obtener.

Aplicaciones

Uno de los factores por los que la lógica difusa se ha utilizado en los últimos años es que el uso de conocimiento experto permite la automatización de tareas. En muchas áreas de aplicación se reduce considerablemente la necesidad de operadores que basan su conocimiento en la experiencia.

Los sistemas basados en lógica difusa son fáciles de diseñar, modificar y mantener. Pese a la pérdida de precisión, la reducción de tiempo de desarrollo y mantenimiento es muy relevante para su uso industrial.

Otro factor a tener en cuenta es que el control difuso permite diseñar soluciones de alta calidad, evitando de esta manera las patentes existentes en otros sistemas de control.

2.2.2. Conjuntos Difusos

Como hemos visto en el capítulo 2.2.1, la lógica difusa permite a un ordenador razonar en términos lingüísticos y reglas de una forma similar a como lo realizan los seres humanos. A diferencia de la lógica booleana clásica, la lógica difusa es multi-evaluada definiendo grados de pertenencia (grados de verdad).

Como lógica multi-evaluada, en la definición de grados de pertenencia, la lógica difusa emplea valores continuos entre 0 (que representa hechos totalmente falsos) y 1 (totalmente ciertos). Así, la lógica binaria clásica puede verse como un caso particular de la lógica difusa.

Zadeh [22] propone en 1965 por primera vez la noción de Conjunto Difuso. Este hecho marca el principio de una nueva teoría denominada **Teoría de Conjuntos Difusos**.

Los conceptos se asocian a conjuntos difusos en un proceso llamado *fuzzificación*. Una vez que tenemos los valores fuzzificados podemos trabajar con reglas lingüísticas y obtener la salida, que podrá seguir siendo difusa o defuzzificada para obtener un valor discreto clásico.

De este modo, a diferencia de la teoría clásica de conjuntos que se basa en el principio básico de la lógica de forma que un individuo pertenece o no pertenece a un conjunto, la idea básica de un conjunto difuso es que un elemento forma parte de un conjunto con un determinado grado de pertenencia. Así, una proposición es parcialmente cierta o falsa. Este grado se expresa mediante un entero en el intervalo $[0,1]$.

De esta forma, un conjunto difuso proporciona una transición suave entre los límites de lo que sería un conjunto clásico.

El **Universo del discurso** se define como todos los posibles valores que puede tomar una variable determinada.

Definición de Conjunto Difuso

Un conjunto difuso puede definirse como una clase en la que hay una progresión gradual desde la pertenencia al conjunto hasta la no pertenencia; o visto de otra forma, en la que un objeto puede tener un grado de pertenencia definido entre la pertenencia total o no pertenencia. Desde esta perspectiva, los conjuntos convencionales pueden verse como un caso particular de conjuntos difusos: un conjunto difuso que sólo admite dos grados de pertenencia, cero y uno.

Si generalizamos una función para que los valores asignados a los elementos del conjunto caigan en un rango particular y así indicar el grado de pertenencia de los

elementos a ese conjunto, tendremos una función de pertenencia de un determinado conjunto difuso. Definido el rango como $[0,1]$:

- La función toma el valor 1 para un grado de pertenencia del elemento total; o dicho de otra forma, el elemento está totalmente en el conjunto.
- La función toma el valor 0 para un grado de pertenencia nulo del elemento; es decir, el elemento no pertenece al conjunto.
- La función toma valores que varía entre 0 y 1 para un cierto grado de pertenencia del elemento al conjunto.

Aunque $[0,1]$ es el rango de valores más utilizado para representar funciones de pertenencia, cualquier conjunto arbitrario con alguna ordenación total o parcial podría ser utilizado.

Operaciones de Conjuntos Difusos

Las tres operaciones básicas [16] que se definen sobre conjuntos convencionales pueden generalizarse de varias formas en conjuntos difusos.

1. Unión.

La forma generalizada de la unión es la T-conorma. Podemos definirla con la siguiente función:

$$\begin{aligned} \perp: [0,1] \times [0,1] &\rightarrow [0,1] \\ \mu_A \cup B(x) &= \perp[\mu_A(x), \mu_B(x)] \end{aligned}$$

Para que una función se pueda considerar como una unión difusa, debe satisfacer los siguientes axiomas $\forall a, b, c \in [0,1]$:

- a) Elemento Neutro: $\perp(a, 0) = a$
- b) Conmutatividad: $\perp(a, b) = \perp(b, a)$
- c) Monotonicidad: Si $a \leq c$ y $b \leq d$ entonces $\perp(a, b) = \perp(c, d)$
- d) Asociatividad: $\perp(\perp(a, b), c) = \perp(a, \perp(b, c))$

Algunas T-conormas ampliamente utilizadas son:

- Máximo: $\perp(a, b) = \max(a, b)$
- Producto: $\perp(a, b) = (a + b) - (a \times b)$
- Suma limitada (o de Lukasiewicz): $\perp(a, b) = \min(a+b, 1)$

2. **Intersección.** La forma generalizada de la intersección se denomina T-norma. Es una función de la forma:

$$T:[0,1] \times [0,1] \rightarrow [0,1]$$

$$\mu_A \cap B(x) = T[\mu_A(x), \mu_B(x)]$$

Una T-norma satisface los siguientes axiomas $\forall a, b, c \in [0,1]$:

- a) Elemento Neutro: $T(a, 1) = a$
- b) Conmutatividad: $T(a, b) = T(b, a)$
- c) Monotonicidad: Si $a \leq c$ y $b \leq d$ entonces $T(a, b) = T(c, d)$
- d) Asociatividad: $T(T(a, b), c) = T(a, T(b, c))$

Algunas T-normas ampliamente utilizadas son:

- Mínimo: $T(a, b) = \min(a, b)$
- Producto algebraico: $T(a, b) = ab$
- Diferencia limitada (o de Lukasiewick): $T(a, b) = \max(0, a + b - 1)$

3. **Complemento.** El complemento está definido por una función del tipo

$$c: [0,1] \rightarrow [0,1]$$

Tiene que satisfacer los siguientes axiomas:

- a) Condiciones límite o frontera: $c(0)=1$ y $c(1)=0$
- b) Monotonicidad: $\forall a, b \in [0,1]$ si $a \geq b$ entonces $c(a) \leq c(b)$
- c) c es una función continua
- d) c es involutiva $\forall a \in [0,1]$ tenemos $c(c(a)) = a$

Al igual que sucedía con los operadores de unión y de intersección también para el complemento existen gran variedad de clases.

Propiedades

Los conjuntos clásicos y los difusos tienen las mismas propiedades. A partir de ahora se podrían definir los conjuntos convencionales como un subconjunto de los conjuntos difusos.

- Conmutativa: $A \cap B = B \cap A$

- Asociativa: $A \cup (B \cup C) = (A \cup B) \cup C$
- Distributiva: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
- Idempotencia: $A \cup A = A$ Y $A \cap A = A$
- Involución: $\neg(\neg A) = A$
- Transitiva: IF $(A \subset B) \cap (B \subset C)$ THEN $A \subset C$ ¹
- Leyes de Morgan: $\neg(A \cap B) = \neg A \cup \neg B$ y $\neg(A \cup B) = \neg A \cap \neg B$

Se pueden obtener gran variedad de expresiones empleando las operaciones, propiedades y modificadores anteriores.

Representación de Conjuntos Difusos

Los conjuntos crisp son útiles pero presentan problemas en muchas situaciones. Para definir un conjunto difuso hay que definir su función de pertenencia. Un método habitual es preguntar a un experto sobre el dominio del problema y representarlo mediante diferentes funciones (típicamente triangulares y trapezoidales). También se pueden utilizar, como veremos más adelante, funciones curvas o la función Singleton.

Para representar un conjunto difuso continuo en un ordenador necesitamos expresar esa función de pertenencia y mapear los elementos del conjunto con su grado de pertenencia. Aunque puede usarse a priori cualquier tipo de función, en la práctica se emplean funciones lineales con una descripción de su vector de ajuste.

2.2.3. Variables lingüísticas

Para representar el conocimiento en razonamiento aproximado, que se explica en la sección 2.2.4, se han de utilizar variables lingüísticas. Una variable lingüística [20] es aquella cuyos valores son palabras o sentencias en un lenguaje natural o artificial. De esta forma, una variable lingüística sirve para representar cualquier elemento que sea demasiado complejo, o del cual no tengamos una definición se pueda describir en términos numéricos.

Así, una variable lingüística está caracterizada por:

$$(X, T(X), U, G, M)$$

Donde:

- X es el nombre de la variable.

- $T(X)$ es el conjunto de términos de X . Representa la colección de sus valores lingüísticos.
- U es el **universo del discurso** o dominio subyacente.
Véase el ejemplo, si hablamos de distancia “cercana” o “aproximadamente 25cm”, el dominio subyacente es un dominio numérico, los centímetros.
- G es una gramática libre de contexto mediante la que se generan los términos en $T(X)$.
- M es una regla semántica que asocia a cada valor lingüístico de X su significado $M(X)$ ².

Los símbolos terminales de las gramáticas incluyen:

- Términos primarios, se asocian a adjetivos.
- Modificadores, se asocian a adverbios.
- Conectores lógicos, *NOT*, *AND* y *OR*.

Normalmente se definen los conjuntos difusos de los términos primarios y, a partir de estos, se calculan los conjuntos difusos de los términos compuestos, con la adición de los modificadores. Una etiqueta lingüística se forma como una sucesión de los símbolos terminales de la gramática.

Un uso habitual de las variables lingüísticas es en reglas difusas. Naturalmente cada conjunto difuso representa un valor lingüístico que puede tomar la variable.

Modificadores

Los modificadores son empleados por variables lingüísticas para cambiar la forma de los conjuntos difusos.

Se distinguen tres tipos de modificadores:

- de concentración,
- de dilatación,
- y de intensificación.

² $M(X)$ denota un subconjunto difuso en el *universo del discurso*.

2.2.4. Razonamiento aproximado. Reglas difusas

La introducción del concepto de Razonamiento Aproximado y otros componentes acabarían formando el cuerpo de la lógica difusa. Con esto se propone la utilización de los conjuntos difusos para el manejo cuantitativo del razonamiento aproximado empleando lógica difusa.

Mediante el uso de los conjuntos difusos es posible dotar de significado matemático proposiciones, fácilmente entendibles para humanos, utilizando los modificadores lingüísticos para adaptar los calificativos a lo que se quiere decir. Así para la representación y utilización del conocimiento impreciso aparece el concepto de *variable lingüística*.

En multitud de ocasiones, la programación clásica no es suficiente para que un sistema realice funciones complejas. Cuando un sistema no ha sido programado explícitamente para realizar una función y se le pide que la realice, el sistema tiene que razonar. Cuando el número de hechos y reglas aumenta, el sistema tiene que poder verificar gran cantidad de hechos que surgen en las etapas de razonamiento.

Reglas Difusas

El razonamiento aproximado se utiliza para representar y razonar con conocimiento expresado en forma de primitiva atómicas, enunciadas en lenguaje natural.

A este tipo de expresión se le denomina proposición atómica difusa. La interpretación de una expresión atómica general viene dada por la pertenencia de una variable física a un conjunto difuso que denota un valor arbitrario del universo del discurso U . Esta interpretación determina el grado en que la expresión es satisfecha dado un valor específico de la variable de entrada física.

Usando este concepto de proposición difusa y conectores lingüísticos con *y*, *o* y *no* es posible componer proposiciones difusas más complejas. El significado de estas proposiciones viene dado por la interpretación de los conectores lingüísticos. Esta interpretación se hace en base a las operaciones de intersección, unión y complemento que se han visto en la sección 2.2.2. Hay que tener en cuenta que, el grado de satisfacción de una expresión constituye un conjunto difuso y, por tanto, estos conectores deben interpretarse mediante operadores de conjuntos difusos. Se puede expresar de manera simbólica una **regla difusa** como:

***IF* proposición difusa *THEN* proposición difusa**

Donde **proposición difusa** puede ser una proposición difusa atómica o compuesta. Podemos definir una proposición sencilla de este tipo mediante:

ρ : *IF* X es A *THEN* Y es B

El antecedente y consecuente de una regla puede tener múltiples partes. En los sistemas de reglas clásicos, si el antecedente es cierto, el consecuente es también cierto. En sistemas *fuzzy* donde el antecedente es difuso, todas las reglas se ejecutan parcialmente, y el consecuente es cierto en el grado de pertenencia que lo sea el antecedente.

Inferencia Difusa

Se define la inferencia difusa como el proceso de obtener un valor de salida para un valor de entrada empleando la teoría de conjuntos difusos. A continuación veremos dos tipos de inferencia: el modelo de Mamdani y el de TSK³.

■ Inferencia de Mamdani

Es posiblemente el método más empleado, propuesto por Ebrahim Mamdani [10] en 1975. Los pasos del proceso son los siguientes:

1. **Fuzzificación de las variables de entrada.** En primer lugar se definen las fronteras de los valores tomados y se determina el grado de pertenencia de las entradas a los conjuntos difusos asociados
2. **Evaluación de las reglas.** Con las entradas anteriores, se aplican a los antecedentes de las reglas difusas.

En el caso de que una regla tenga múltiples antecedentes, se utilizan los operadores *AND* u *OR* para obtener un único número que represente el resultado de la evaluación. Este número se aplica al consecuente, aplicando los métodos de recorte o escalado según el valor de verdad del antecedente:

- El *recorte* corta el consecuente con el valor de verdad del antecedente.
 - El *escalado* proporciona un valor más preciso, preservando la forma original del conjunto difuso. Se obtiene multiplicando todos los valores por el valor de verdad del antecedente.
3. **Agregación de las salidas.** Éste es el proceso en el que se combinan las funciones de pertenencia de todos los consecuentes previamente escalados, combinando para obtener un único conjunto difuso por cada variable de salida.
 4. **Defuzzificación.** Tras los pasos realizados anteriormente, es habitual expresar el resultado final mediante un valor convencional. El método más ampliamente usado es el centroide, que calcula el punto donde una línea vertical que divide el conjunto en dos áreas con igual masa.

³TSK: Takagi-Sugeno-Kang.

- Inferencia TSK

Como hemos visto , el modelo de inferencia de Mamdani requiere de algún tipo de defuzzificación. En general, este método no es muy eficiente desde el punto de vista computacional. Podemos disminuir el tiempo de inferencia empleando una función matemática en el consecuente, de forma que el formato general de regla de inferencia TSK es:

$$\rho: \text{IF } x \text{ es } A \text{ AND } y \text{ es } B \text{ THEN } z \text{ es } f(x,y)$$

Este tipo de método proporciona mayor eficiencia, pero no presentan un marco tan natural para la representación del conocimiento humano. Un tipo habitual de representación del consecuente es un **singleton**⁴ , que toma valor uno en un valor puntual del *universo del discurso* y cero en cualquier otro punto.

Empleando este tipo de aproximación, la inferencia TSK y de Mamdani son muy parecidas, tomando las reglas el siguiente formato:

$$\rho: \text{IF } x \text{ es } A \text{ AND } y \text{ es } B \text{ THEN } z \text{ es } k$$

Siendo k un valor constante para el **singleton**. La salida clásica en este caso se obtiene mediante una sencilla agregación, media de pesos, de los singletons.

⁴Singleton: instancia única o punta discreta. Es un patrón de diseño diseñado para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto.

Capítulo 3

Estado del arte

3.1. Entorno inteligente de estudio

El *SmartLab* en el que se ha desarrollado el estudio se encuentra en la dependencia 109 del edificio C6 y pertenece al CEATIC, del que se ha hablado anteriormente en el capítulo 1.1.

Dicho laboratorio está provisto con más de 130 sensores de diferentes tipos, repartidos por las distintas dependencias, los cuales se definen en la sección 3.2.

A continuación, se presenta un plano en planta, figura 3.1 del laboratorio, mostrando la disposición de sus dependencias.



Figura 3.1: Plano del *SmartLab*

Las dependencias del laboratorio son las siguientes:

- **Dormitorio y baño.**

Este espacio se muestra en la figura 3.2 y tiene integrada la cama, el lavabo y el baño.



Figura 3.2: Espacio dormitorio y baño

- **Salón.**

El espacio mostrado en la figura 3.3 corresponde al salón. Dicho espacio dispone de sofá, televisión y diferentes muebles.



Figura 3.3: Espacio salón

- **Cocina.**

La cocina, que se representa en la figura 3.4, tiene varios electrodomésticos destacables que se han utilizado para el estudio como son el microondas, el lavavajillas y la tetera, así como los muebles.



Figura 3.4: Espacio cocina

3.2. Elementos que intervienen. Sensores

A modo de introducción se van a definir los sensores utilizados para el desarrollo del proyecto, que se encuentran integrados en el ambiente inteligente definido con anterioridad en el apartado 3.1.

Para una mejor visualización, se diferencian dependiendo del tipo de sensor así como el trabajo que desempeñan en el entorno.

3.2.1. Sensores de visión

D-Link DCS-5020L

Este sensor de visión que permite una visualización remota, mediante dispositivos móviles, gracias a la fácil conexión a la red del hogar. Conectada a Internet, puede enviar notificaciones y/o imágenes vía email al detectar movimiento y/o sonido.

Una gran ventaja de este sensor es la posible monitorización en la oscuridad gracias a los LEDs infrarrojos que incorpora.



Figura 3.5: Sensor de visión 1

D-Link DCS-933L

Al igual que el citado anteriormente, la DCS-933L Wireless N es un sensor de visión que permite una visualización remota, mediante dispositivos móviles, gracias a la fácil conexión a la red del hogar.

Este dispositivo es capaz de captar imágenes y recibir datos de tipo audio.

Una de sus características más destacables es su función de extensión de *WiFi* que amplía el rango de cobertura inalámbrica, eliminando así posibles obstáculos a la hora de la instalación.



Figura 3.6: Sensor de visión 2

3.2.2. Sensores de proximidad

Estimote Proximity Beacons

Se trata de un dispositivo que trabaja con tecnología *Bluetooth* enviando información de forma continua. Los iBeacons detectados se reciben en un array ordenados por proximidad.

Disponen además de un acelerómetro y sensor de temperatura.



Figura 3.7: *Estimote proximity Beacons*

3.2.3. Sensores de apertura y cierre

Sensor *Smartsense Open/Closed*

Se trata de un sensor magnético que permite controlar el estado abierto o cerrado de puertas y ventanas. En el caso de nuestro entorno de estudio también se ha colocado en la tetera y mando a distancia para controlar si se encuentran en la posición de reposo.

Este dispositivo *ZigBee*¹ incorpora un termostato que permite controlar la temperatura de las zonas interiores y puede combinarse con un conmutador compatible con SmartThings para realizar acciones definidas previamente.

Dicho sensor dispone de una parte móvil con un imán fijada en la parte abatible y la unidad principal se encuentra situada en la parte fija del elemento. Cuando ambas partes no están enfrentadas se envía automáticamente una señal de radio *ZigBee*.



Figura 3.8: Sensor *Smartsense Open/Closed*

3.2.4. Sensores de movimiento

Sensor *Smartsense Motion*

Este sensor es un dispositivo *ZigBee* que detecta movimiento a un alcance de unos cuatro metros y medio de distancia con un campo de visión de 120 grados. Gracias a su termostato incorporado, también es capaz de detectar la temperatura.

Se instala de forma rápida y fácil en cualquier superficie gracias a unos adhesivos que incorpora o una aplicación para fijar a la pared mediante tornillería.

¹*ZigBee*: es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo.



Figura 3.9: Sensor *Smartsense Motion*

Fibaro sensor de movimiento

El detector de movimiento multisensor 4 en 1 de Fibaro es capaz de detectar movimiento, temperatura e intensidad lumínica y vibraciones. Se instala de forma rápida y fácil en cualquier superficie. Su LED multicolor indica detección de movimiento y nivel de temperatura.



Figura 3.10: Fibaro sensor de movimiento

3.2.5. Otros sensores

SmartThings Hub

En este caso, no hablamos de un sensor en sí mismo sino de un centro de actividad que conecta todos los dispositivos inteligentes que interactúan en el entorno entre sí.

Este concentrador se conecta mediante *ethernet*, lo que permite que se comuniquen con la aplicación *SmartThings* o la nube de dicha aplicación. Se conecta con dispositivos *ZigBee* y *Z-Wave*².



Figura 3.11: *SmartThings Hub*

3.3. Reconocimiento mediante visión por computador

Para el reconocimiento de los agentes que intervienen en el ambiente mediante los sensores de visión, se han utilizado unas camisetas ,disponibles en el laboratorio, que presentan marcadores como los que se definen en el apartado 3.3.1.

La razón por la cual se han utilizado este tipo de marcadores es por la implementación de la librería ArUco soportada por *OpenCV*³.

Asimismo, cabe destacar otro tipo de marcadores cada vez más utilizados en sistemas de visión por computador, como los que se definen en el apartado 3.3.2.

²*Z-Wave*: es un protocolo de comunicación inalámbrica que se controla a través de Internet, con una pasarela *Z-Wave* o un concentrador de actividad.

³*OpenCV* es una biblioteca libre de visión artificial.

3.3.1. *ArUco marker*

En la figura 3.12 se presenta un marcador ArUco, similar a los que se han utilizado para este estudio. Como se puede observar, este tipo de marcadores son un cuadrado compuesto por un borde negro ancho, que facilita su rápida detección, y una matriz binaria interna, de color blanco, que determina la identificación del mismo.

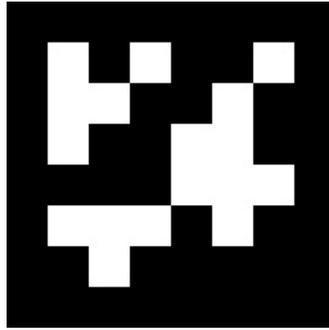


Figura 3.12: Marcador ArUco

Este tipo de marcadores son unos de los más utilizados gracias a su rápida detección y robustez. Dado que los marcadores pueden encontrarse girados en el entorno, se debe determinar una rotación para identificar inequívocamente sus esquinas [5].

Diccionario de marcadores

En primer lugar se debe establecer un diccionario de marcadores [6]. Esto es una base de datos de las codificaciones binarias del conjunto de marcadores de los que se dispongan. La propiedades que determinan dicho diccionario son:

- El tamaño del diccionario. Es el número de marcadores que componen dicho diccionario.
- El tamaño del marcador. Es el tamaño de los marcadores, el número de bits de cada marcador.

Hay que tener en cuenta la limitación que supone trabajar con marcadores de gran tamaño ya que la gestión de números muy altos de bits no es práctica. Por esta razón, es más cómodo el trabajo con el número identificador de cada marcador.

Calibración de los sensores de visión

El módulo ArUco se utiliza también para la calibración de los sensores de visión. Consiste en la obtención de los parámetros intrínsecos⁴ y de los coeficientes de distorsión

⁴Los parámetros intrínsecos definen la óptica y la geometría interna del sensor de visión. Permanecen constantes si las características y posiciones relativas entre la óptica y el sensor imagen no varían.

del sensor.

Son constantes en tanto no varíen las características y posiciones relativas entre la óptica y el sensor imagen.

La calibración se realiza utilizando la función `calibrateCamera()` de *OpenCV* que precisa de correspondencias entre puntos del ambiente y sus proyecciones para diferentes orientaciones con respecto al sensor. Estas correspondencias se obtienen gracias a las esquinas de patrones de tableros.

Detección de marcadores

En este apartado se explican los pasos seguidos para el reconocimiento automático de marcadores [6]. El proceso se dirige a la detección de rectángulos y a la extracción del código binario de los mismo. Para esta proposición, se toma como entrada una imagen en escala de grises. Los pasos empleados son:

1. **Segmentación de imágenes.** En primer lugar se detecta el contorno de la imagen en escala de grises, mediante el uso de detección de frontera Canny. Enfoque de umbral adaptativo muy robusto para diferentes condiciones de iluminación.
2. **Extracción de contornos y filtración.** Posteriormente, se realiza una extracción de contorno en la imagen umbral con el algoritmo de *Suzuki* y *Abe*. Esto produce el conjunto de contornos de imagen.

Después, se realiza una aproximación poligonal utilizando el algoritmo de Douglass-Peucker. Dado que los marcadores están encerrados en contornos rectangulares, los que no se aproximan a polígonos con cuatro vértices son descartados.

Finalmente, se simplifican los contornos cercanos dejando sólo los extremos.

3. **Extracción del código del marcador.** El siguiente paso consiste en analizar la región interna de los contornos para extraer su código interno. Primero se elimina la perspectiva calculando la matriz homográfica. La imagen resultante se limita utilizando el método de *Otsu*⁵, que proporciona el valor óptimo de límite de imagen dado que la distribución de imágenes es bimodal. Entonces la imagen binarizada, se divide en una cuadrícula regular y a cada elemento se le asigna un valor, 0 ó 1, dependiendo de los valores de la mayoría de los píxeles en él.

Una primera prueba de rechazo consiste en detectar la presencia del borde negro. Si todos los bits del borde son cero, entonces la cuadrícula interna se analiza utilizando el método siguiente.

⁵Método de *Otsu* o método del valor umbral que segmenta gráficos rasterizados

4. **Identificación de marcadores y corrección de errores.** En este punto, es necesario determinar cuáles de los marcadores obtenidos pertenecen en realidad al diccionario y cuáles sólo forman parte del entorno. Una vez extraído el código de cada marcador, se obtienen cuatro identificadores diferentes (uno por cada rotación posible). Si alguno de ellos se encuentra en D , se considera como un marcador válido. Para acelerar este proceso, los elementos del diccionario se ordenan como un árbol binario equilibrado. Para ello, los marcadores están representados por el valor entero obtenido por concatenación de todos sus bits. Se puede deducir entonces que este proceso tiene una complejidad logarítmica $O(4 \log_2(|D|))$, donde el factor 4 indica que es necesaria una búsqueda para cada rotación del marcador examinado.

Si no se encuentra ninguna coincidencia, se puede aplicar el método de corrección. Considerando que la distancia mínima entre dos marcadores en D es \hat{r} , se puede detectar y corregir un error de como máximo $\lceil (\hat{r} - 1)/2 \rceil$ bits. Por lo tanto, nuestro método de corrección de marcadores consiste en calcular la distancia del marcador examinado erróneo a todos los marcadores en D . Si la distancia es igual o menor que $\lceil (\hat{r} - 1)/2 \rceil$, se considera que el marcador más cercano es el correcto. Este proceso, sin embargo, presenta una complejidad lineal de $O(4 |D|)$, ya que cada rotación del marcador tiene que ser comparada con todo el diccionario. Sin embargo, es un proceso altamente paralelizable que se puede implementar de manera eficiente en los equipos actuales.

5. **Recuento de las esquinas y estimación de la posición.** Una vez que se ha detectado un marcador, es posible estimar su posición respecto a la cámara minimizando iterativamente el error de reproyección de las esquinas⁶.

Para la detección de esquinas se hace una regresión lineal de los píxeles del lado del marcador para calcular sus intersecciones.

En la figura 3.13 se presentan las etapas del procesado de la imagen para la detección automática de marcadores, donde:

- a) Presenta la imagen original.
- b) Muestra el resultado de la aplicación del del umbral local.
- c) Muestra la detección del contorno.
- d) Presenta la aproximación poligonal y elimina los contornos irrelevantes.
- e) Es un ejemplo uno de los marcadores tras su transformación.

⁶Para ellos se ha utilizado el algoritmo de Levenberg-Marquardt

f) Muestra la asignación de bits por célula.

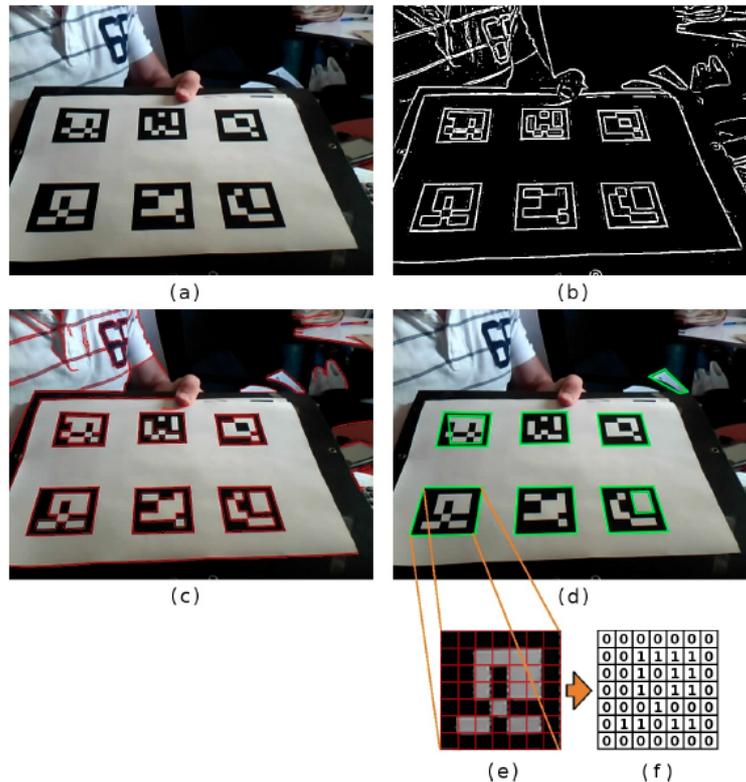


Figura 3.13: Procesado de imagen

3.3.2. Vuforia frame marker

Vuforia se trata de una multiplataforma de desarrollo de software de realidad aumentada. Ofrece multitud de funcionalidades en el reconocimiento de imágenes, objetos, formas y textos. Además, es capaz de detectar y estimar la posición, con respecto al sensor de visión, de una determinada marca.

Como se observa en la figura 3.14, esta marca se compone de un borde negro, dos veces más ancho que el patrón binario que identifica la marca. En el interior del marcador, se podría colocar alguna imagen/logo para hacer más visual la marca ya que la identificación reside en los bordes de la misma [8].

Es importante para la fase de detección, el contraste y/o brillantez de la zona interna de la marca con el borde.

Sin embargo, a pesar de la robustez de este sistema, hay que destacar algunas desventajas:

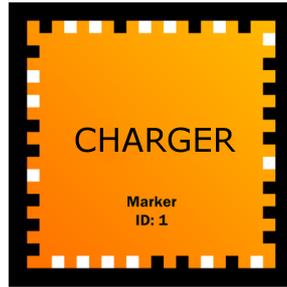


Figura 3.14: *Vuforia frame marker*

1. No hay posibilidad de acceder al código fuente, por lo que es imposible acceder al algoritmo para modificar características.
2. Baja flexibilidad ya que el número de marcadores está limitado.
3. Existe limitación en cuanto al tamaño de las marcas.

Capítulo 4

Metodología

En este capítulo se explica la metodología seguida para el desarrollo de las diferentes escenas que se han llevado a cabo para el estudio.

Como se ha explicado en la sección 2.2.4, el proceso empleado para la inferencia difusa conlleva:

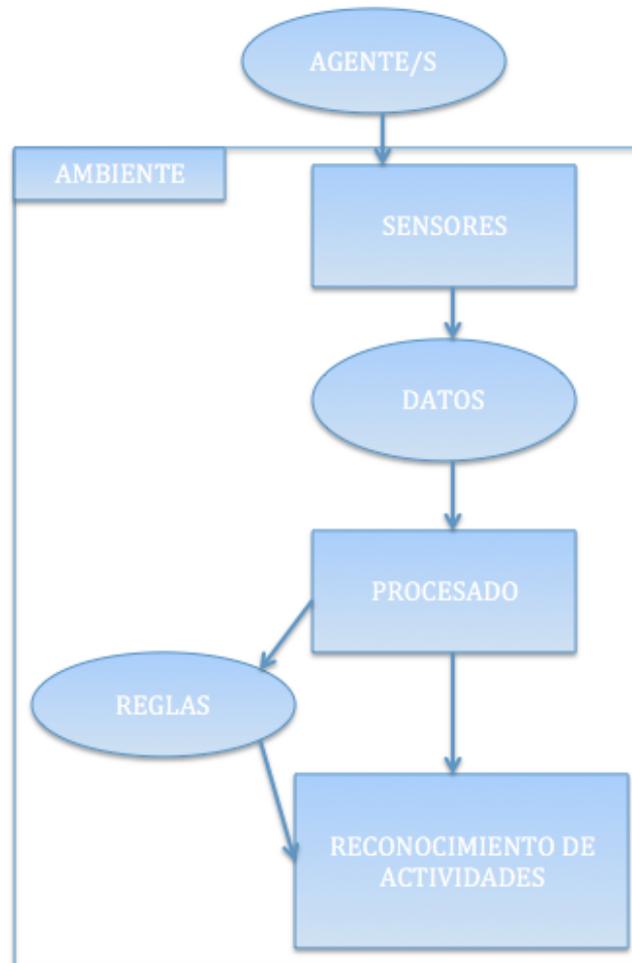
- **Fuzzificación de los valores de entrada.** En el apartado A.2.1, se ha explicado con más detenimiento el bloque de función que se ha empleado, así como la rutina, apartado A.2.2, para darle un grado de pertenencia a los valores obtenidos.
- **Formulación de reglas y evaluación de las mismas.** Para la definición de las reglas de control de un sistema difuso, previamente descritas en la sección 2.2.4, se han establecido una lista de sentencias *IF-THEN*, exclusivas para este entorno, que aparecen desarrolladas en el apartado 4.4.
- **Defuzzificación.** En nuestro caso, no se ha realizado ya que no aporta un significado relevante la obtención de unos valores concretos de salidas.

En la figura 4.1 se muestra el esquema del proceso de funcionamiento. En primer lugar tenemos un ambiente inteligente provisto de múltiples sensores que permiten la monitorización del entorno inteligente.

Cuando un determinado agente interacciona con el ambiente se reciben datos de los sensores. Dichos datos se procesan y, a partir de unas reglas definidas, se realiza el reconocimiento de las actividades que está ejecutando el agente.

En caso de actuar más de un agente en el entorno, el procedimiento sería el mismo, teniendo la precaución de identificar las actividades que realiza cada uno de ellos.

Las diferentes fases que se han dibujado en el *Graficet* se irán definiendo a lo largo de esta sección.

Figura 4.1: *Grafcet*.

4.1. Adquisición de datos

Los datos de los sensores de visión se adquieren mediante *OpenCV*. Esta API¹ incluye, entre otras, la adquisición de imágenes desde un sensor de visión web en tiempo real o mediante un fichero.

Por otro lado, la monitorización del resto de sensores se realiza a partir del sistema web [11, 23], diseñado por un alumno de la Universidad de Jaén, que se observa en la figura 4.2

Esta web ofrece un listado temporizado con el estado que presenta cada sensor en dicho instante. Más adelante, en la sección A.3, se presenta un ejemplo de muestra de datos .

¹Interfaz de programación de aplicaciones.

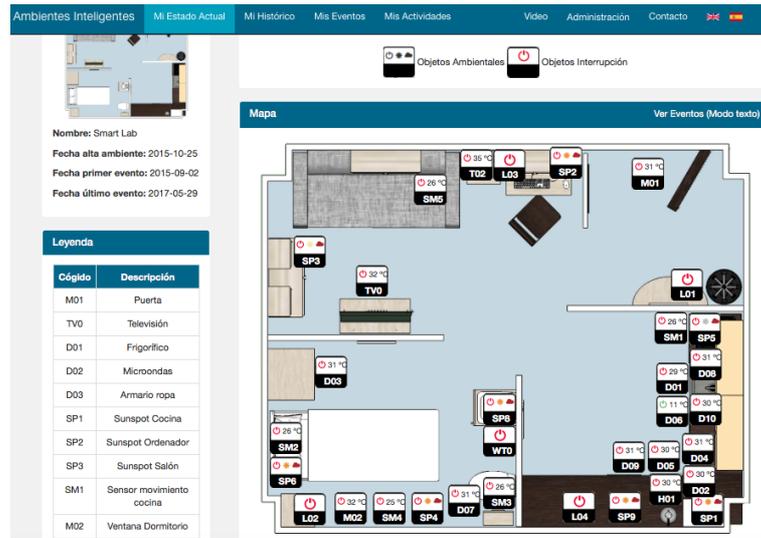


Figura 4.2: Entorno web de monitorización.

4.2. Procesado de datos

Una vez que los sensores del ambiente han proporcionado los datos que aseguran la presencia de un agente, estos han de tratarse para incluirlos en un conjunto difuso como se ha visto en el apartado 2.2.2 y asignarles cierto grado de pertenencia.

4.2.1. Tratamiento imágenes. Reconocimiento de marcadores.

Para el procesado de las marcas se ha utilizado la librería ArUco. Se trata de una librería de código **libre** para aplicaciones de realidad aumentada que soporta *OpenCV*. Esta *API* incluye la adquisición de imágenes en tiempo real o mediante un fichero de los datos recogidos por los sensores..

El proceso de reconocimiento de marcadores es el siguiente:

1. Con la *API* se realiza un reconocimiento a tiempo real de los marcadores [17].
2. Del resultado anterior, se obtiene un archivo de texto en el que presentan los marcadores reconocidos con la información del sensor, la distancia hasta la marca, la identificación de la marca, el fotograma al que corresponde el reconocimiento y una marca temporal.
3. Se pasan los valores obtenidos con el *time stamp* y la marca reconocida.
4. Una vez que se han fuzzificado los datos, se incorporan a una tabla excel con los datos correspondientes al resto de sensores.

4.2.2. Tratamiento resto de datos

Como se ha visto en la figura 4.1, una vez obtenidos los datos de los sensores, se produce su procesado. Para ello, se realiza la *fuzzificación* de estos valores, por lo que es importante tener en cuenta el tiempo de duración de cada una de las diferentes actividades para utilizar la variable lingüística precisa en cada caso.

Tras este paso, se han dispuesto los valores *fuzzificados* en una tabla de desarrollo excel para la obtención de las gráficas que reconocen las actividades que realizan los agentes, según las reglas definidas en el apartado 4.4.

4.3. Términos lingüísticos empleados

Tal y como se ha visto anteriormente en el apartado 2.2.3, las variables lingüísticas o términos lingüísticos sirven para representar cualquier elemento demasiado complejo en términos numéricos.

En este estudio, las variables lingüísticas empleadas han sido:

1. Variables de distancia

Se ha utilizado únicamente una variable para el procesado de este tipo de datos ya que las condiciones de las dependencias no dan lugar al tratamiento de longitudes mayores.

Los sensores *eBeacons*², son de los que se reciben estos datos, indicando la proximidad del agente a las balizas.

isClose

La función de pertenencia asignada a este término ha sido:

$$isClose := (0,1) (2,1) (2'5,0);$$

Que se corresponde con la función matemática 4.1.

$$f(x) = \begin{cases} 1 & \text{si } 0 < x < 2 \\ 5 - 2x & \text{si } 2 < x < 2'5 \\ 0 & \text{si } 2'5 < x \end{cases} \quad (4.1)$$

Estas funciones lineales indican que, cuando la distancia sea mayor o igual a dos metros y medio el grado de pertenencia será nulo. Es decir, a una distancia de dos metros y medio el agente no se encuentra cerca de la baliza.

Se puede representar como la figura 4.3 que se muestra a continuación.

²Estimate Proximity Beacons

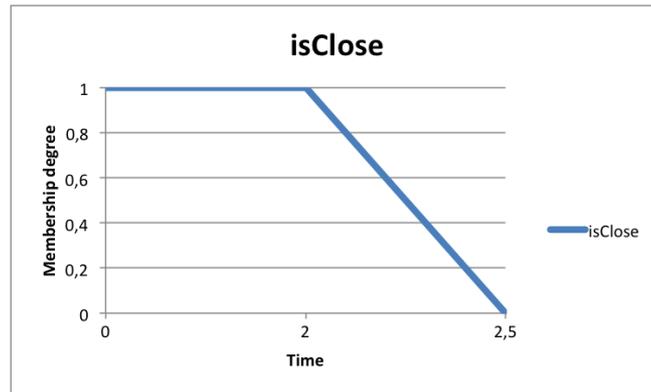


Figura 4.3: Función de pertenencia variable lingüística *isClose*.

2. Variables de tiempo

El resto de sensores ofrecen datos temporales y para facilitar el procesado de los mismos se han planteado los términos que se presentan a continuación. Es interesante la asignación de diferentes variables lingüísticas dado que no todas las actividades tienen la misma duración y para ello se asignan diferentes funciones de pertenencia en el tiempo.

Recently

Se ha descrito esta variable lingüística para acciones realizadas recientemente, duraciones cortas en el tiempo de ejecución de las actividades.

Se define de la forma:

$$\textit{Recently} := (0,1) (2,1) (4,0);$$

En la figura 4.4 se muestran las funciones lineales que define el grado de pertenencia de la variable lingüística y que se corresponde con la función 4.2.

$$f(x) = \begin{cases} 1 & \text{si } 0 < x < 2 \\ 2 - 0'5x & \text{si } 2 < x < 4 \\ 0 & \text{si } 4 < x \end{cases} \quad (4.2)$$

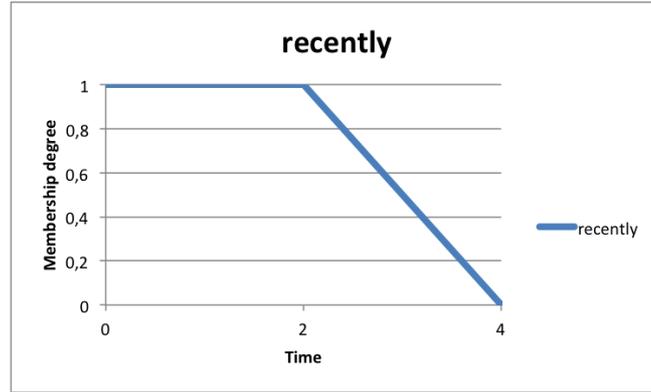


Figura 4.4: Función de pertenencia variable lingüística *Recently*.

For a while

Con este término lingüístico se procesan los datos para duraciones de tiempo algo mayores que en el caso anterior, “*por un momento*”. La función de pertenencia asignada a este término ha sido:

$$\text{For a while} := (0,1) (15,1) (25,0);$$

Cuya función matemática se ha representado en la función 4.3.

$$f(x) = \begin{cases} 1 & \text{si } 0 < x < 15 \\ 2'5 - x/10 & \text{si } 15 < x < 25 \\ 0 & \text{si } 25 < x \end{cases} \quad (4.3)$$

En la figura 4.5 se muestra la función lineal que define dicha variable lingüística.

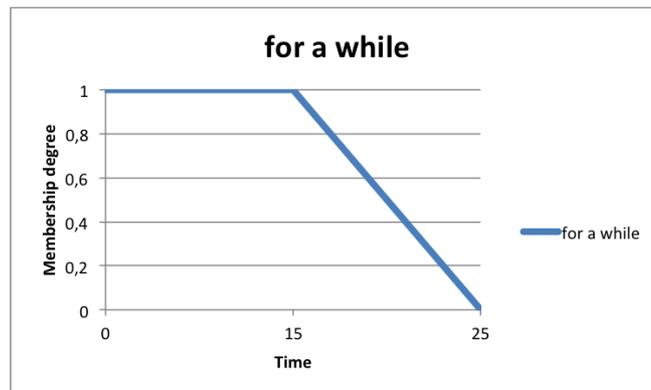


Figura 4.5: Función de pertenencia variable lingüística *For a while*.

For a long while

Para etapas temporales aún más largas, se ha definido una última variable, “*por mucho tiempo*”, cuya función se presenta en la figura 4.6 y se expresa matemáticamente como:

$$\text{For a long while} := (0,1) (30,1) (45,0);$$

Y se expresa matemáticamente como la función 4.4.

$$f(x) = \begin{cases} 1 & \text{si } 0 < x < 30 \\ 3 - x/15 & \text{si } 30 < x < 45 \\ 0 & \text{si } 45 < x \end{cases} \quad (4.4)$$

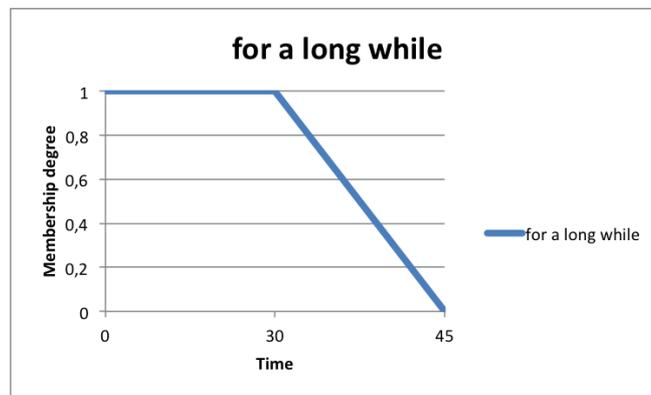


Figura 4.6: Función de pertenencia variable lingüística *For a long while*.

4.4. Formulación de reglas. Inferencia de actividades

De la realización de operaciones con diferentes antecedentes se obtiene el consecuente, como se ha visto en el apartado 2.2.4. En nuestro caso, los antecedentes son los sensores y el consecuente es la actividad ejecutada.

Por ello se han definido una serie de reglas que conllevan la inferencia de las actividades que realizan los agentes en el ambiente. La formulación de estas reglas ha sido posible gracias a la fusión entre los elementos que se encuentran instalados en el entorno inteligente.

Previamente a la definición de las reglas se presenta la tabla 4.1 que engloba la totalidad de los sensores que intervienen en el estudio. Esta codificación es importante ya que las reglas se exponen atendiendo a dicha clasificación.

Código	Descripción
D01	Frigorífico apertura/cierre
D02	Microondas apertura/cierre
D03	Armario ropa apertura/cierre
D04	Armario tazas apertura/cierre
D05	Lavavajillas apertura/cierre
D07	Tapa W.C. apertura/cierre
D11	Armario salón apertura/cierre
H01	Hervidor apertura/cierre
M01	Puerta apertura/cierre
SM1	Sensor movimiento cocina
SM2	Sensor movimiento cama
SM3	Sensor movimiento baño
SM4	Sensor movimiento dormitorio
SM5	Sensor movimiento sofá
TV0	Televisión apertura/cierre
WT0	Grifo cocina apertura/cierre
WT1	Grifo baño apertura/cierre

Tabla 4.1: Sensores que intervienen en el estudio.

La inferencia de estas actividades ha sido posible gracias a las operaciones aplicadas a los conjuntos difusos.

- Regla 1: aplicando la unión entre el sensor de movimiento y el sensor de visión situados ambos en la habitación, se ha establecido que el agente se encuentra tendido sobre la cama.
- Regla 2: tras recibir actividad del sensor de apertura de la puerta del armario que alberga las tazas, y posteriormente haberse abierto también el grifo de la cocina, se deduce que el agente ha llenando el vaso del grifo.
- Regla 3: realizando la unión de haber llenado la taza del grifo (regla 2) y detectar la apertura del microondas, se establece que el agente ha calentado agua del grifo en el microondas.
- Regla 4: tras haber observado en el plano temporal que el agente se ha preparado el desayuno, y posteriormente detectar movimiento y presencia en el salón, mediante los elementos correspondientes, sensor de movimiento y sensor de visión, respectivamente, se deduce que está desayunando en el sillón.
- Regla 5: se llega a la conclusión de que el agente se está aseando gracias a la presencia de éste en la dependencia y la apertura del grifo.

La presencia del agente se verifica mediante la intersección de los conjuntos, procesados gracias al análisis de los datos obtenidos mediante los sensores de movimiento y visión.

- Regla 6: tras afirmar que el agente haya preparado el desayuno (regla 3) y continuar con su presencia en la dependencia, se establece que se encuentre desayunando en la cocina.

La presencia del agente en la cocina se establece gracias a la intersección de varios elementos que intervienen: dos sensores de visión y un sensor de proximidad. La redundancia de datos permite una mayor precisión en la identificación de reglas.

- Regla 7: de la unión de la apertura del W.C. y a la presencia del agente en el baño, se deduce que éste está utilizando el W.C.

La presencia del agente se verifica mediante la intersección de los conjuntos, procesados gracias al análisis de los datos obtenidos mediante el sensor de movimiento y visión.

- Regla 8: de la unión de la apertura del armario de la ropa y la presencia del habitante en la habitación, se deduce que se está vistiendo.

La presencia del ocupante en la habitación se verifica mediante la intersección de los conjuntos, procesados gracias al análisis de los datos obtenidos mediante el sensor de movimiento y visión.

- Regla 9: tras la apertura del del armario del salón y confirmar la presencia del agente en la dependencia mediante otros elementos, se deduce que éste ha cogido un objeto del interior del armario.

La presencia del agente en el salón se verifica gracias la intersección de los conjuntos correspondientes al sensor de movimiento y de los dos sensores de visión de dicha dependencia.

- Regla 10: gracias a la unión de los conjuntos de obtenidos por el sensor de apertura del armario de las tazas, la presencia del hervidor en la base y el sensor de visión ,se identifica que el agente se ha servido agua del hervidor.

- Regla 11: al observarse la apertura del lavavajillas y una posterior apertura del armario donde se guardan las tazas, junto con la presencia del agente en la cocina, gracias al sensor de visión,se establece que ha las tazas sacado el lavavajillas y guardado al menos algunas de ellas.

- Regla 12: de la unión de los conjuntos difusos obtenidos de la apertura del armario de las tazas, de la apertura del grifo de la cocina y la presencia por medio del sensor de visión, se determina que el agente está bebiendo agua del grifo.
- Regla 13: al ejecutar la operación unión a los conjuntos ofrecidos por el sensor que determina que la televisión está encendida y el procesado de las imágenes, se deduce que el agente se encuentra viendo la televisión.

En este caso, el salón consta de dos sensores de visión por lo que el resultado final de reconocimiento mediante visión se traduce en la intersección de los valores obtenidos por cada uno de los sensores.

- Regla 14: de la unión de la apertura de la puerta principal y la situación del agente en el *hall*, se concluye diciendo que el primer agente ha abandonado el entorno.
- Regla 15: tras la apertura de la puerta principal del entorno y la localización del segundo agente en el recibidor, se deduce que dicho agente ha abandonado el entorno.

Puede verse que tanto esta regla como la anterior podrían definirse como una sola, ya que se trata de la misma actividad. Sin embargo, se han planteado de este modo para diferenciar e identificar en la evaluación las actividades que realiza cada agente de manera independiente.

- Regla 16: esta regla define la acción de guardar las tazas en el armario. Ello ha sido posible gracias a haberse producido la apertura del lavavajillas y la posterior apertura del armario, al mismo tiempo que el sensor de visión situaba al agente en la cocina.
- Regla 17: de la unión de la apertura del armario de las tazas y la posterior apertura del frigorífico, así como la detección del agente en la cocina, se concluye que dicho agente se está sirviendo bebida fría.

Por lo general, como se observa, el reconocimiento de las actividades realizadas en el entorno ha sido posible gracias a la operación de unión de los conjuntos difusos.

Para mayor facilidad, se han presentado el conjunto de reglas en la tabla 4.2. La denominación de las reglas en dicha tabla corresponde con las gráficas de evaluación.

Reglas ¹	Operación ²	Acción ³
Rule01	MIN(SM1 ; CameraBedroom)	Estar acostado en la cama
Rule02	MIN(D04 ; WT0 ; CameraKitchen)	Llenar vaso del grifo
Rule03	MIN(Rule02 ; D02)	Calentar agua del grifo
Rule04	MIN(SM5 ; CameraLivingroom)	Desayunar en el sillón
Rule05	MIN(WT1 ; StayBathroom ⁴)	Asearse
Rule06	MIN(Rule03 ; StayKitchen ⁵)	Desayunar en la cocina
Rule07	MIN(D07 ; StayBathroom ⁴)	Utilizar el W.C.
Rule08	MIN(D03 ; StayBedroom ⁶)	Vestirse
Rule09	MIN(D11 ; StayLivingroom ⁷)	Coger objeto
Rule10	MIN(D04 ; H01 ; KitchenAg1)	Servir agua caliente del hervidor
Rule11	MIN(D04 ; D05 ; KitchenAg1)	Sacar tazas del lavavajillas
Rule12	MIN(D04 ; WT0 ; KitchenAg2)	Beber agua del grifo
Rule13	MIN(TV0 ; CameraAg2 ⁸)	Ver la televisión
Rule14	MIN(M01 ; HallAg1)	Salir del domicilio, Agente 1
Rule15	MIN(M01 ; HallAg2)	Salir del domicilio, Agente 2
Rule16	MIN(D04 ; D05 ; KitchenAg2)	Guardar tazas
Rule17	MIN(D01 ; D04 ; KitchenAg2)	Servir bebida fría

¹ Reglas: presentan las operaciones que se tienen que realizar con los conjuntos de los sensores para realizar una determinada actividad.

² Operación: cálculo que hay que realizar con los conjuntos de los diferentes sensores para obtener el resultado de la acción.

³ Acción: actividad a realizar.

⁴ StayBathroom = MAX(SM3 ; CameraBedroom)

⁵ StayKitchen = MAX(SM1 ; Camera1Kitchen; Camera2Kitchen)

⁶ StayBedroom = MAX(SM2 ; SM4 ; CameraBedroom)

⁷ StayLivingroom = MAX(SM5 ; CameraLivingroom1Ag1 ; CameraLivingroom2Ag1)

⁸ Camera = MAX(CameraLivingroom1Ag2 ; CameraLivingroom2)

Tabla 4.2: Reglas establecidas del análisis del estudio.

Capítulo 5

Evaluación

5.1. Desarrollo de escenas y presentación de resultados

Como fase experimental del estudio, se han simulado una serie de actividades de la vida diaria con diferentes agentes en el entorno inteligente mencionado en el capítulo 3.1.

Se ha de destacar la diferencia entre las secuencias, ya que tanto los sensores como el tratamiento de los datos varía. Esto es debido a que la realización del proyecto ha tenido lugar en dos fases:

- En la primera tuvo lugar el desarrollo de las dos primeras escenas y se presenta un único agente.

En esta fase no estaban disponibles todos los elementos que actualmente componen el *Smartlab*, por lo que para algunas actuaciones como pueden ser apertura de puertas, apertura de microondas, apertura de grifo, etc., se simularon dichos valores.

- En la segunda parte, tuvieron lugar el resto de escenas, interviniendo dos agentes en el ambiente.

Con la finalidad de facilitar el seguimiento del estudio, durante la evaluación de las dos primeras escenas, apartado 5.1.1 y apartado 5.1.2, respectivamente, se han presentado gráficamente todos los elementos que definen la regla. Sin embargo, para evitar gráficos con infinidad de líneas, en el resto de escenas la representación ha sido únicamente de las reglas que tienen lugar, una global y otra dependiendo del agente.

5.1.1. Escena 1

En esta secuencia se ha querido representar una mañana en la vida de un determinado ocupante, desde que se levanta, las posibles actividades que realiza hasta que abandona el domicilio.

Inicialmente el agente se encuentra en el dormitorio, tendido sobre la cama. Transcurridos unos segundos se levanta y se dirige hacia la cocina. Abre el armario de donde coge una taza. Cierra dicho armario y se dispone a llenar el vaso del grifo.

Acto seguido, abre el microondas para introducir el vaso y calentar el agua. Espera el tiempo pertinente a que se termine de calentar el agua y saca el vaso.

Se dirige al salón a tomarse el desayuno sentado sobre el sofá. Pasados unos segundos, vuelve a la cocina a dejar el vaso en el fregadero.

Tras haber dejado el vaso, va hacia el baño, donde se asea y termina la secuencia.

En la figura 5.1 se muestran diferentes partes del desarrollo de la escena.



Figura 5.1: Ejemplo de imágenes captadas durante la escena 1.

Una vez que se ha explicado cuál ha sido la secuencia de la escena, se pasa a la evaluación de la misma.

En la figura 5.2 se observa que el agente, durante la primera parte del desarrollo de la escena, se encuentra en la cama, tal y como se ha descrito con anterioridad. Se ha llegado a esta conclusión gracias al procesado de los datos obtenidos por el sensor de visión y la baliza de proximidad situados en la habitación.

Como se puede observar, la función “SM4”, que corresponde con la baliza de proximidad del cuarto, fluctúa en regiones de tiempo en el que el análisis de los datos, en el plano temporal de esta escena, sitúa al agente en otras dependencias. No se trata de un error ya que es debido a la cercanía entre habitaciones.

Por otro lado, el agente se encuentra desde el comienzo de la escena en la cama, pero se puede observar en la figura 5.2, que hasta el segundo 10, aproximadamente, el sensor de visión toma el valor 0. Esto se debe a que dichos sensores pierden en ocasiones la localización de la marca.

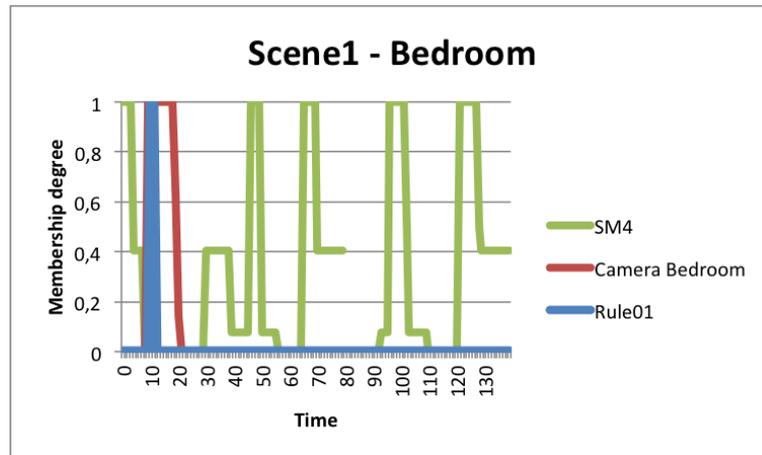


Figura 5.2: Escena 1: Cuarto.

La escena continúa en la cocina. La figura 5.3 representa el procesado de datos tomados en dicha dependencia:



Figura 5.3: Escena 1: Cocina.

En este caso, se reconocen dos actividades:

- Llenar el vaso de agua del grifo.
- Calentar el agua que contiene el vaso.

Según la lógica humana, una actividad conlleva la otra, es decir, no sería posible calentar agua si el recipiente estuviera vacío. Por tanto, se podría decir que no es una situación real ya que gráficamente se muestra que la actividad de calentar el agua en el microondas termina antes de que se termine de llenar el vaso.

Este caso se ha presentado con la intención de mostrar la importancia a la hora de definir las variables lingüísticas y las funciones de pertenencia de las misma. Si se hubieran procesado los datos con una variable temporal intermedia entre *por un momento* y *por mucho tiempo*, esta representación se habría suavizado y se reflejaría fielmente con la realidad.

Prosiguiendo con la escena, el agente se dirige al salón donde se toma el desayuno. Se muestran los datos en la figura 5.4.

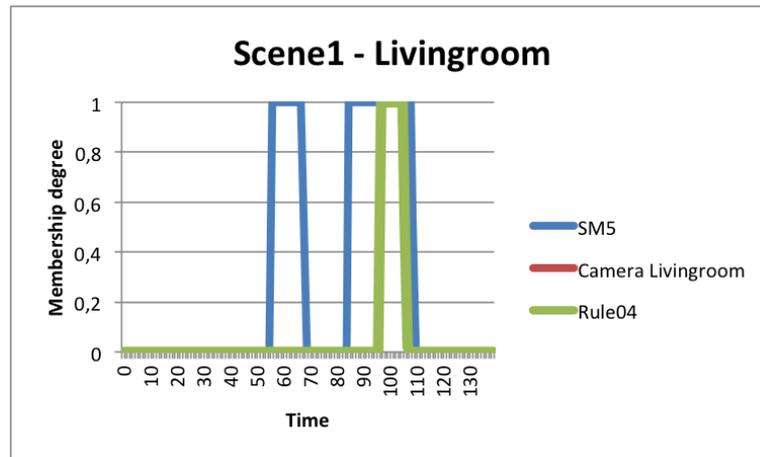


Figura 5.4: Escena 1: Salón.

De la unión del sensor de proximidad y el sensor de visión se obtiene la regla que define la actividad que está realizando el agente en esta dependencia: *sentarse en el sofá mientras desayuna*.

Este es un buen ejemplo para ver cómo la operación de unión en los conjuntos difusos lo que hace es presentar el mínimo de tantos conjuntos como confluyan. Es por ello, que dicha regla aquí coincide fielmente con el procesado de las imágenes obtenidas.

Tal y como se ha explicado al principio de este apartado, tras desayunar el agente en el salón, se dirige a la cocina a dejar el vaso. En la figura 5.3 se ve que el sensor de visión si detecta al agente. Sin embargo, falta información para asegurar la actividad que el ocupante realiza en ese momento.

Por poner un ejemplo, en el caso de que se hubiera enjuagado el vaso, se tendría información del sensor de apertura del grifo y se podría concluir en una nueva regla, atendiendo a “WT0” y el sensor de visión de la cocina.

Dicho esto, la siguiente información que tenemos es que el ocupante se dirige al baño, como se aprecia en la regla 5: *asearse*.

Al igual que ocurría en la actividad realizada en el cuarto, como se ve en la figura 5.5, hay una variación los datos aportados mediante la baliza de proximidad del baño “SM5”

en instantes de tiempo que las reglas han afirmado que el ocupante no se encontraba en dicha dependencia.



Figura 5.5: Escena 1: Baño.

Para mostrar el desarrollo de las actividades en el transcurso de la escena 1, se ha presentado la figura 5.6

Es una buena muestra gráfica de las actividades que se realizan primero y cuáles después, en qué dependencia comienza y termina la escena, etc.

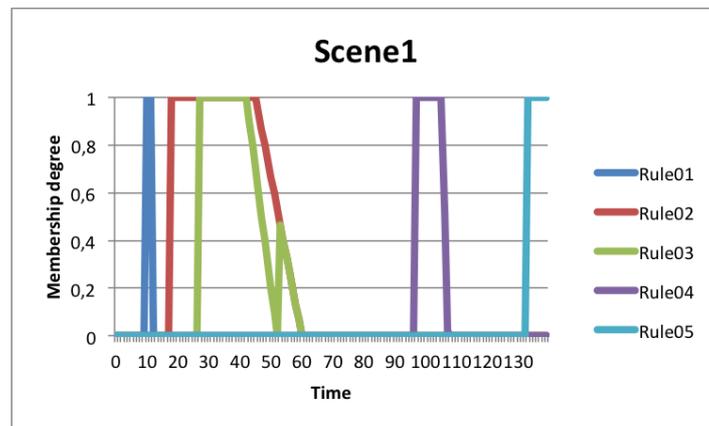


Figura 5.6: Escena 1: Reglas.

5.1.2. Escena 2

Se han representado otra variante de actividades, al igual que en la escena 5.1.1, antes de salir del domicilio de un agente.

La secuencia comienza con el agente tendido sobre la cama del dormitorio. Al cabo de un determinado tiempo, se levanta para dirigirse a la cocina. Una vez allí, abre el armario para coger un vaso y llenarlo de agua. Previamente a este acto, cierra el armario y abre el grifo.

Cuando el vaso se encuentra lleno, se dirige al microondas para calentar el contenido del vaso. Espera el tiempo pertinente a que se termine de calentar el agua y saca el vaso.

Se dirige hacia la zona de la barra situada en la cocina. Transcurridos unos segundos, deja el vaso en el fregadero y se dirige al baño para asearse.

Tras haber terminado en el baño, se dirige a la puerta, abandonando la escena.



Figura 5.7: Muestra de imágenes captadas durante la segunda escena.

Siguiendo el esquema realizado en el apartado 5.1.1, se presentan los resultados del procesado de datos dependiendo de la dependencia en orden cronológico.



Figura 5.8: Escena 2: Cuarto.

La figura 5.8 representa la habitación. Sigue el mismo patrón que la escena 1. Sin embargo, en este caso se observa que los valores obtenidos del sensor de proximidad “SM4”

no varían tanto como en el caso anterior. Esto es debido a que al observar la respuesta tras grabar la primera escena se corrigió la posición del elemento.

En este caso, se modificó la orientación del sensor que estaba captando imágenes en el salón para tener redundancia de datos adquiridos por visión.

Para simplificar las funciones representadas en la figura 5.9, se ha establecido la función *Stay* como la intersección de los dos sensores de visión que controlaron la cocina durante esta escena. En este caso se definen las reglas:

- Llenar de agua del grifo un vaso.
- Calentar el agua del vaso.
- Desayunar en la cocina.



Figura 5.9: Escena 2: Cocina.

Como en el caso anterior, hay cierta discordancia con la realidad de las operaciones a realizar para el proceso. Esto puede mejorarse incluyendo nuevas variables y asignando dependiendo de la actividad a realizar. Es decir, si aproximadamente en calentar un vaso de agua para realizar alguna infusión se invierte un minuto, pero para llenar un vaso se necesitan, dependiendo del caudal del grifo, alrededor de quince segundos, aplicar la variable propia a cada actividad.

Por lo general se ha hecho así pero hay que tener en cuenta que las escenas están simuladas y la realización real de las escenas sería mucho más larga. Es por ello que, en algunos casos, no salgan valores fiables en su totalidad.

A continuación, figura 5.11, se muestran el procesado de los datos del cuarto de baño. La regla 5 facilita el tiempo de aseo que el ocupante invierte.

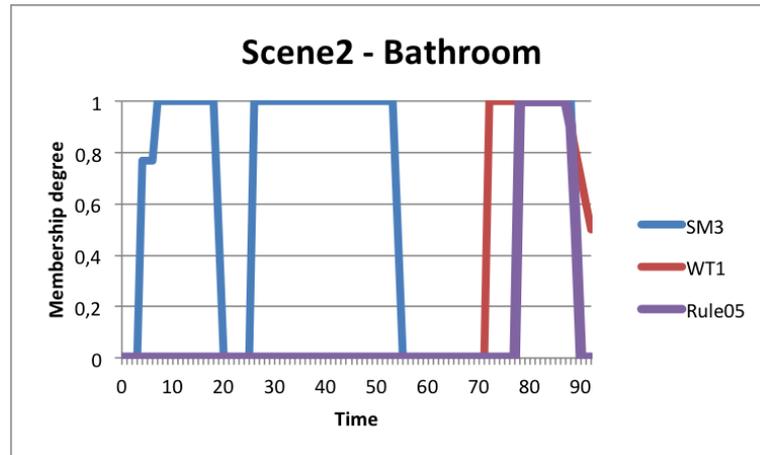


Figura 5.10: Escena 2: Baño.

Tanto en esta escena como en la anterior se puede apreciar cierta distinción con respecto a las reglas establecidas en el apartado 4.4. La diferencia reside en que la orientación del sensor de visión colocado en la habitación, para las escenas 1 y 2, no facilita el procesamiento de datos de la imagen. De ahí que la función representada en la figura 5.10 sea “SM3”, sensor de proximidad del baño, y no *StayBathroom* como indica la tabla 4.2.

Sin embargo, debido a que la función indicada es consecuencia de la operación intersección, no afectaría al conjunto de la regla.

En la figura 5.11 se muestra un resumen de cómo suceden las actividades en el plano temporal de la escena.



Figura 5.11: Escena 2: Reglas.

5.1.3. Escena 3

En esta secuencia han intervenido dos agentes, por lo que se explicarán las actividades que realizan cada uno de ellos por separado.

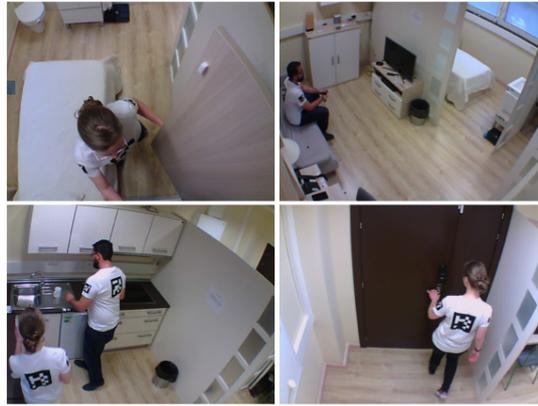


Figura 5.12: Ejemplo de imágenes captadas durante la escena 3.

La figura 5.13 muestra la coexistencia de las tareas que ejecutan los ocupantes en el transcurso de la escena.



Figura 5.13: Escena 3: Reglas .

Por facilidad y como se dijo en la introducción de esta secuencia, se estudian de manera independiente:

Agente 1

Comienza la secuencia en el dormitorio. Abre el armario para seleccionar la ropa para vestirse. Cuando termina esta acción y ya ha cerrado el armario, vuelve a abrir el armario

para comprobar la correcta colocación del resto de las prendas, las reordena y cierra el armario.

Se dirige a la cocina a desayunar. Coge un vaso y se sirve agua del hervidor. Una vez que se bebe el contenido, lo deposita en el fregadero.

Finalmente, se encamina hacia la calle. La escena finaliza dejando la puerta abierta.

Una vez conocido el desarrollo de la escena se definen las actividades que realiza este ocupante:

- Se viste
- Se sirve agua caliente del hervidor en una taza.
- Abandona el domicilio.

La figura 5.14 presenta el ciclo temporal de las actividades que realiza este agente durante la secuencia. Gráficamente se puede comprobar que la descripción de la escena coincide con los datos procesados.



Figura 5.14: Escena 3: Reglas Agente 1.

Se se analiza el gráfico, podría decirse que la acción de *servirse agua del hervidor* comienza cuando la acción de *vestirse* sigue teniendo un grado de pertenencia bastante alto. Esta función podría suavizar con alguna variable lingüística entre *reciente* y *por un momento*. No obstante, se ha querido dejar así ya que, en la mayoría de los casos, dicha acción tiene esa extensión en el tiempo o es incluso mayor, lo que da lugar a conclusiones interesantes.

Continuando con el transcurrir de la escena, no habría ningún problema de solapamiento de actividades, dejando un tiempo más que suficiente para que el ocupante se tome el té y para salir del domicilio, no situándolo en otras dependencias.

Agente 2

Por otro lado, el segundo agente se encuentra inicialmente en el salón, sentado en el sofá mientras está viendo la televisión. Pasado un rato se dirige a la cocina.

En la cocina, va a coger una taza. Se confunde y abre en primer lugar el armario de los platos y después el de las tazas. Una vez que tiene la taza, abre el grifo, la llena de agua y se lo bebe. Al terminarse el agua, deja el vaso en el fregadero y se dirige a la salida.

Abandona la escena cerrando la puerta.

De este desarrollo mostrado se obtienen las acciones ejecutadas por este ocupante:

- Ve la televisión.
- Bebe agua del grifo.
- Abandona el domicilio.



Figura 5.15: Escena 3: Reglas Agente 2.

Esta gráfica representa las acciones del segundo agente en el transcurso de la escena. Puede verse que en este caso el procesado de datos no da lugar a dudas de cuándo empieza y termina una actividad, ya que no hay ningún solapamiento.

Así se muestra la importancia de elegir las variables lingüística adecuadas atendiendo a la temporalidad de la acción.

5.1.4. Escena 4

En consonancia con la escena anterior y para estudiar la multiocupación del ambiente, esta escena también se ha desarrollado con dos agentes.

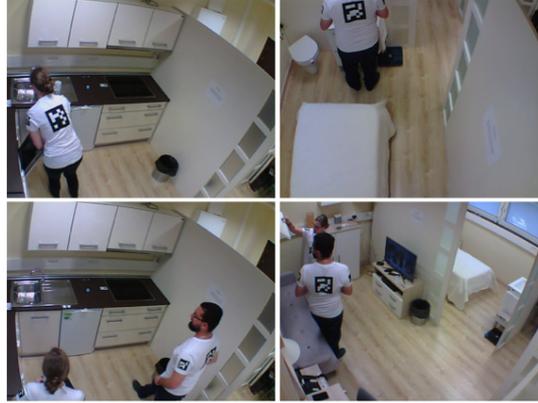


Figura 5.16: Muestra de imágenes captadas durante la escena 4.

El cómputo de actividades que se llevan a cabo durante la escena por ambos ocupantes se representan en la figura 5.17.

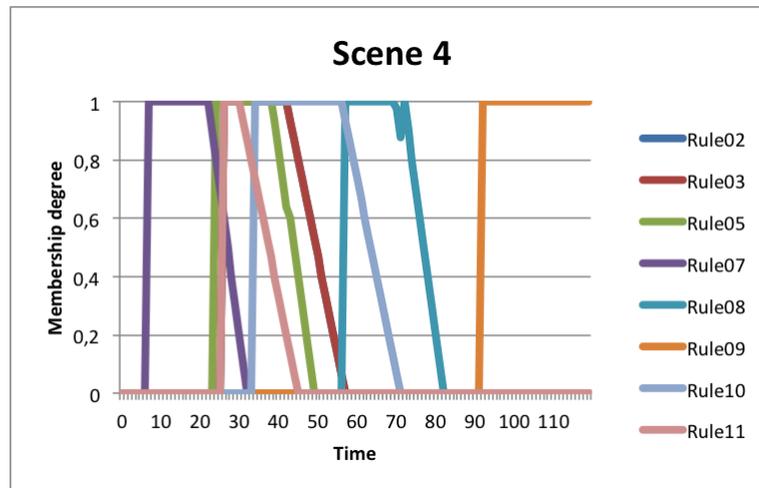


Figura 5.17: Escena 4: Reglas.

Agente 1

Inicialmente se encuentra en la cocina. Abre el lavavajillas de donde coge una taza, la cual llena e introduce en el microondas para calentar el contenido. En este caso, está preparando el desayuno del otro agente.

Abre el armario de las tazas y coge una, que se dispone a llenar con el agua del hervidor. Cuando ha terminado el microondas saca el recipiente y lo deja sobre la encimera. Se sienta a esperar al otro agente.

Una vez que tienen ambos agentes el desayuno, se dirigen hacia el sillón para tomárselo mientras están hablando y viendo la televisión.

Cuando han llegado al salón, recuerda que tiene que coger algún elemento del armario y se sientan.

De la descripción de la secuencia se tienen las siguientes actividades:

- Saca las tazas del lavavajillas.
- Llena una taza de agua del grifo.
- Calienta el contenido de la taza en el microondas.
- Se sirve agua caliente del hervidor.
- Coge un objeto del armario del salón.

Vamos a comprobar ahora gráficamente el resultado de la acción del agente. gracias a la figura 5.18.



Figura 5.18: Escena 4: Reglas Agente 1.

Agente 2

Este agente comienza la escena sentado en la cama. Se levanta y se dirige al cuarto de baño. Tras haber realizado sus necesidades, se lava las manos en el lavabo y se asea. Va hacia el armario para vestirse.

En la cocina, con el agente 1, coge la taza y se dirigen al salón a ver la televisión.

- Utiliza el W.C.
- Se asea.
- Se viste.



Figura 5.19: Escena 4: Reglas Agente 2.

Como se observa en la figura 5.19, el procesado de las acciones de este agente se encuentran muy diferenciadas y la duración de éstas es comparable a una escena real.

Por otro lado, el desarrollo de la escena indica que al final ambos agentes se encuentran en el salón viendo la televisión. Dicha implementación no se ha presentado en las gráficas por falta de información. La televisión se encendió antes del comienzo de la secuencia, por lo que los datos que aporta dicho sensor, para la franja temporal de la escena, no constaban en el *Datasheet*.

Se podría haber simulado sin problema con el uso de una variable de tiempo mucho mayor que *por mucho tiempo* ya que, como ha ocurrido a la hora de procesar los datos, cuando los agentes llegaban al salón, el grado de pertenencia de la función “que la televisión estuviera encendida” era nulo.

5.1.5. Escena 5

Tras grabar las secuencias 3 y 4, se comprobó que había dificultades a la hora del reconocimiento de los marcadores. Es por ese motivo por el que se llegaron a realizar dos escenas más, ambas en la misma dependencia.

Para esa ocasión, el sensor de visión se colocó en un lugar estratégico comprobando que la pérdida del marcador fuera mínima.

Se presentan en un mismo apartado ya que el reconocimiento de actividades realizado de las secuencias por separa ha sido el igual, modificándose algunos matices de tiempo, por lo que no aporta ninguna novedad. La figura 5.20 es una muestra representativa de ambas escenas.



Figura 5.20: Muestra de imágenes captadas durante las escenas 5 y 6.

La coexistencia de actividades realizada por los ocupantes que confluyen en la habitación se muestran en la figura 5.21. En ella se ve el tiempo que tardan los ocupantes en ejecutar las acciones. Sin embargo, para saber a ciencia cierta qué actividad realiza quién, se presentan las gráficas de los trabajos de cada ocupante por separado.

Esto se podría haber evitado definiendo las reglas por ocupante, como se ha hecho en algunas de ellas.

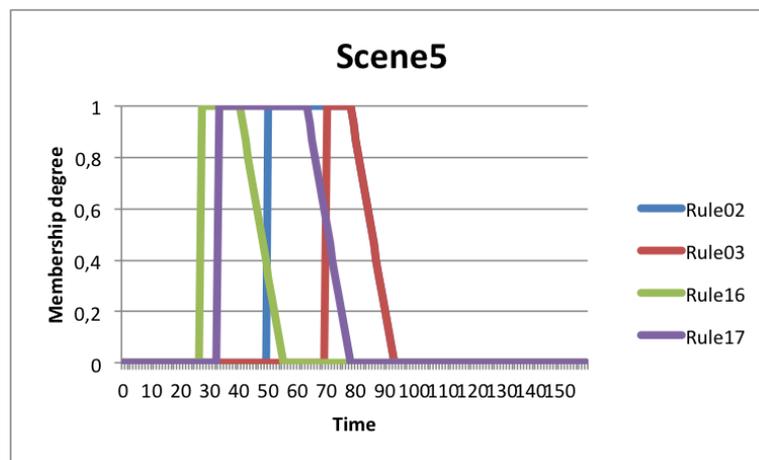


Figura 5.21: Escena 5: Reglas.

Agente 1

Entra en la cocina a los pocos segundos de comenzar la escena. Abre el armario de las tazas y coge una taza. La llena de agua del grifo.

Una vez lleno lo lleva al microondas y cuando termina de calentarse comienza a bebérselo.

Las actividades que realiza son claras:

- Llenar la taza de agua del grifo.
- Calentar el agua en el microondas.

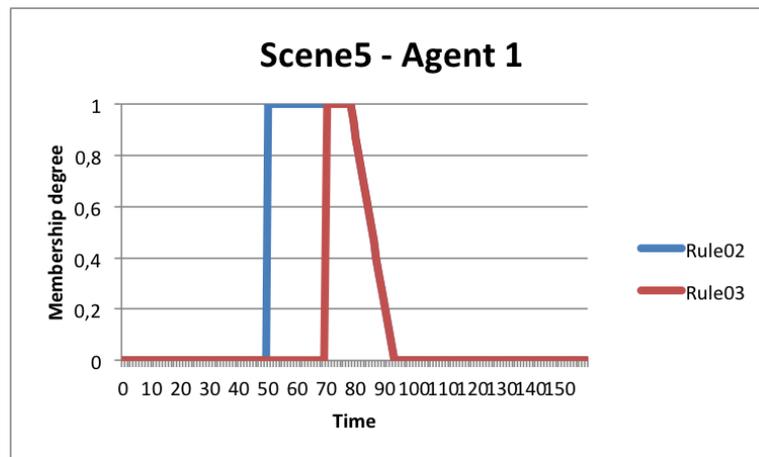


Figura 5.22: Escena 5: Reglas Agente 1.

Como ya se ha comentado en algunas escenas anteriores, la lógica sería que cuando se termina de llenar el vaso, *regla 2*, comienza el uso del microondas, *regla 3*.

Pero, ¿por qué podemos concluir con que esta suposición es correcta? Atendiendo a la regla definida, que el microondas esté funcionando implica, por un lado haber llenado con anterioridad el recipiente y por otro que dicho electrodoméstico se abra. Como es la unión de ambos conjuntos lo que hemos definido como regla, es por ello que presenta este tipo de gráfico.

Agente 2

Este ocupante comienza sacando el lavavajillas e introduciendo una de las tazas en el armario. La otra la utiliza para servirse bebida fría del frigorífico. Permanece en la escena mientras se la bebe.

En la figura 5.23 se han representado gráficamente las actividades que realiza este agente. En ella se observa que confluyen tres funciones, que son las que coinciden con las acciones que lleva a cabo el ocupante:

- Guardar las tazas del lavavajillas.
- Se sirve bebida fría.

Como se observa en la nombrada gráfica, se muestran dos funciones. En primer lugar, parece poco cercano a la realidad el hecho de que *guardar tazas del lavavajillas* sea tan corto. Esto se debe a que para la simulación se introdujeron únicamente dos tazas, una la que guarda el agente y la otra la que utiliza para grabar la escena.

Se puede apreciar que la acción de *servirse bebida fría* es más duradera en el tiempo ya que conlleva esperar a que el vaso esté lleno.



Figura 5.23: Escena 5: Reglas Agente 2.

Dado que en la cocina hay un armario para guardar las tazas y otro para guardar los platos, la regla definida es *sacar tazas del lavavajillas* y no *sacar lavavajillas*, que sería la actividad general. Esta última regla podría haberse definido de haber realizado la simulación utilizando ambos muebles.

5.1.6. Escena de pruebas.

Se grabó una última escena de bastante utilidad para sacar una serie de conclusiones a tener en cuenta, que se expondrán en la sección 6.1. En ella, ninguno de los agentes interactúan con el ambiente en cuanto a adquisición de datos de sensores de movimiento, apertura y cierre se refiere.

La finalidad era obtener valores del sensor de visión situado en la dependencia de la cocina y ver su comportamiento ante algunas situaciones de los marcadores.

En la figura 5.24 se ha realizado una muestra representativa de las imágenes adquiridas durante dicha escena.



Figura 5.24: Imágenes captadas durante la escena de pruebas.

5.2. Discusión de resultados

Del procesado de los datos de las escenas desarrolladas, se obtiene la inferencia de actividades que tienen lugar a lo largo de las diferentes secuencias por los ocupantes del entorno.

En general, excepto en las dos primeras escenas, el reconocimiento de las actividades es del 100%. El motivo de no reconocer la actividad desde el inicio de la escena puede ser debido a dos motivos:

- Que la aplicación de los *eBeacons* comenzara a adquirir datos un instante de tiempo anterior a que lo hicieran los sensores de visión.
- Que la posición del marcador con respecto al sensor no fuera el correcto o incluso que la camiseta presentara alguna arruga que dificultara su reconocimiento.

Este leve desfase en el tiempo al comienzo de las comentadas escenas no es algo significativo, ya que en el resto de las escenas no sucede.

Por otro lado, el solapamiento de actividades que se aprecia en la figura 5.22, tampoco presenta problema alguno. Esto es lógico ya que, al hacer depender una actividad de la anterior, siempre que dicha actividad haya finalizado, la siguiente también lo hará. La lógica “humana” nos diría que para que dicha gráfica fuera cierta, debería estar llenándose el vaso de agua del grifo hasta haber terminado de calentarlo. Esto depende del tipo de

definición. Es decir, siempre que la definición de la pauta sea clara se podrá considerar como cierta.

Al mismo tiempo, la representación de todas las acciones en un mismo plano temporal ayuda a reconocer qué actividades se realizan antes y cuales de ellas lo hacen después. En alguna ocasión, como por ejemplo en la figura 5.19, comienza una nueva acción sin que haya terminado otra anterior. En este caso, en el que no nos encontramos en el supuesto anterior de hacer depender una actividad de otra, estamos en un caso de posible mala elección de la variable de *fuzzificación*.

Esto ocurre ya que hay que ser muy meticuloso con la selección de las variables lingüísticas para cada actividad dependiendo de la duración. Aunque por otro lado se observa que, aunque se podría acortar la duración de la primera acción, se distingue con total claridad tanto la actividad realizada por el agente como el orden cronológico de las mismas.

Capítulo 6

Conclusiones

6.1. Conclusiones finales

La información es uno de los recursos más importantes que se pueden tener en la era tecnológica en la que nos encontramos. Por ello, un entorno inteligente repleto de sensores que nos aporta constantemente datos sobre la situación de los elementos del entorno, ofrece un valor añadido a la calidad de vida de sus ocupantes.

Cuando hablamos de calidad de vida en este sentido se habla de la tranquilidad que aporta tener una conexión íntegra con el entorno para poder comprobar el estado de tu domicilio o para realizar un seguimiento del estado de enfermos. También es posible utilizar dicha conexión para controlar las temperaturas de estancias o incluso alimentar a una mascota.

Otro factor positivo de un ambiente inteligente por el que se están moviendo los investigadores es diagnóstico precoz de enfermedades como demencia al detectar un cambios en las rutinas del agente. Asimismo, la identificación de episodios de crisis epilépticas o de enfermos con autismo.

Lo más importante es identificar lo que se quiere reconocer, las actividades cotidianas propias de los ocupantes del entorno y las reacciones que el ambiente debería tener en el caso de ser requerido.

Gracias al grabado de las escenas, se ha llegado a algunas conclusiones con respecto a los sensores:

- Los sensores de proximidad utilizados para el desarrollo de las dos primeras escenas tienen un alcance demasiado potente para el espacio de estudio.

Esto se ha demostrado cuando en multitud de ocasiones el agente estaba en una dependencia y el sensor lo ubicaba en otra. No ocasiona un problema mayor ya que la redundancia de datos posibilita un análisis adecuado.

- Los sensores de visión han supuesto el gran *hándicap* del trabajo. Esto se debe a la complejidad de estos elementos, ya que hay muchos factores a tener en cuenta para su correcta elección: luminosidad, orientación, etc. Posiblemente la mayor limitación haya sido realizar el reconocimiento con marcadores. No en sí por la aplicación, que como se ha explicado tiene bastante soporte de investigación y es muy robusta, sino por el sistema de transportarlos.

En la escena de pruebas se observó que había diferentes motivos por el que la visión no realizaba el reconocimiento:

1. Por falta de distinción completa de la marca, es decir, taparla. Cuando el agente tapaba con el pelo su marcador dejaba de reconocerse.
2. Por presencia de arrugas en la camiseta. Es evidente, como en el caso anterior, que si la marca no se visualiza bien no la detecte, pero en este caso una leve arruga debido al torso del agente le hacía perder el reconocimiento en multitud de ocasiones.

Un ejemplo de esto es la dificultad de detectar las marcas situadas en las mangas debido a la pérdida de linealidad de la misma por la curva que forma el brazo.

3. La distancia y la orientación del agente con respecto al sensor.

Es posible que el mercado disponga de sensores más precisos y adecuados para este tipo de aplicación.

Por otro lado, la detección de dos agentes para una misma marca temporal fue muy complicado. En la mayoría de las ocasiones, si un agente estaba en la escena, siendo reconocido, cuando llegaba el otro ocupante se pasaba a reconocer el marcador del segundo.

Este último hecho ha dificultado el estudio de reconocimiento mediante visión en entornos con multiocupación ya que se dejaba de reconocer uno para reconocerse el otro.

Otra dificultad encontrada es el desajuste temporal de alguno de estos sensores. Si bien se configuran en la página web del proveedor y no debe suponer mayor inconveniente, en el caso concreto de la adquisición de uno de estos elementos, el desfase temporal obligó a realizar un reajuste del mismo. Esto pudo conseguirse gracias a las imágenes captadas, observando cuándo actuaban el resto de sensores diferentes a los de la visión y aplicando el factor de corrección correspondiente.

- Los sensores de movimiento *Mother & cookies*, diseñados para soporte inteligente a personas de la tercera edad, que se pensaron para dicho estudio, habiéndose incluso utilizado en alguna de las escenas, tuvieron que descartarse.

El motivo es que presenta bastante dificultad a la hora de obtener datos, ya que la *app* tiene códigos cerrados de no fácil manejo.

En general, se puede decir que los elementos que conformen el entorno inteligente deben ir acorde con el ambiente. Existen multitud sensores en el mercado pero las necesidades cambian dependiendo de las variables que se deseen controlar. Como ejemplo, parecería de poco sentido utilizar un sensor provisto de un termostato si no se pretende regular la temperatura.

En cuanto al reconocimiento de actividades encontramos algunas limitaciones:

- Es evidente que en el caso de tener un único ocupante no habría problema ya que toda la actividad que se reciba de los sensores la realiza ese agente. Sin embargo, al tener multiocupación existe una mayor dificultad para identificar actividades que realizan agentes diferentes.

Una solución fue analizar los datos obtenidos de la visión y separar por marcadores concretos, ya que cada camiseta tiene cuatro identificadores distintos. De este modo se han solucionado algunas situaciones. Es decir, en el caso de que un agente esté en una dependencia y otro en otra, como los sensores de cada una de las dependencias ofrecen valores diferentes no tendríamos problema; pero en el caso de que ambos ocupantes se encuentre en la misma habitación, no tendríamos ninguna evidencia de qué agente actúa sobre diferentes sensores.

Estudiando la distancia del sensor a los distintos sensores se podría mejorar esta situación.

- Esta propuesta funciona de manera correcta para los espacios temporales establecidos en las escenas. A pesar de ello, no puede asegurar para escenas reales más largas, donde la monitorización conlleve unas cuantas horas de duración.

Para analizar de forma exhaustiva la acción que realiza en una determinada dependencia, habría que analizar las posiciones y el tiempo que discurre en ella, así como las imágenes y los movimientos realizados durante la acción.

Por último, una de las restricciones de mayor importancia en el paso a una implantación real es la **privacidad** del ocupante. Es evidente que, en el caso de un laboratorio inteligente en el que las pruebas que se realizan se hacen a consciencia, no se necesita la instauración de un sistema determinado de seguridad. Por el contrario, la monitorización de ambientes reales supone:

- La modificación de la forma en la que se almacena los datos obtenidos de la visión. Es importante la pixelación de los rostros de las imágenes captadas de los ocupantes para preservar toda la intimidad que un sistema controlado por visión permita.
- Una mayor seguridad en el sistema para limitar el acceso a la red y que personas ajenas puedan adquirir información de las rutinas y/o actividades que realizan los ocupantes del entorno.
- Como se puede intuir, los dos motivos indicados anteriormente suponen un sobrecoste a la instalación que, en adición al ya elevado coste de los sensores, restringe el acceso a un gran número de usuarios.

6.2. Trabajos futuros

Este proyecto podría conducir al estudio de:

- Zonas críticas o posiciones anómalas para la detección de posibles situaciones de riesgo del agente.

Teniendo en cuenta que el punto fuerte de este estudio es la contribución en la mejora de la calidad de vida de personas con demencia u otras enfermedades con unos patrones de comportamiento determinados, se puede crear una alerta en el entorno para que en ese tipo de situaciones actúe de una determinada forma.

- La identificación de agentes mediante reconocimiento facial.

Como se ha podido observar con el desarrollo del proyecto, hay momentos en los que los sensores de visión dejan de reconocer los marcadores. Este motivo junto con la imposibilidad de asegurar que el ocupante lleve la camiseta siempre que se encuentre en el entorno, son un buen motivo para realizar esa investigación.

- El reconocimiento facial de los estados de los agentes que se encuentren en el ambiente inteligente.

Hoy en día ya existe, a nivel industrial, una aplicación que implementa esta función. Se trata de *Bismart Face and Emotion Recognition* y su funcionamiento se basa en el procesado de una imagen que realiza un sensor de visión, tal y como se propone en esta mejora.

Es interesante ya que este trabajo tendría el objetivo de ofrecer un plus al agente, adaptar el ambiente al estado de ánimo que éste tenga.

- Implementación del sistema automático. Diseño del software que, atendiendo a las actividades realizadas, el ambiente actúe aconsejando cambio de actividades o avisando al agente de posibles peligros.

Apéndice A

Anexos

A.1. Estructura de trabajo

En el diagrama de bloques de la figura A.1, se presenta la estructura de trabajo seguido para este estudio, así como las relaciones que tienen entre sí las diferentes partes de adquisición y procesado.

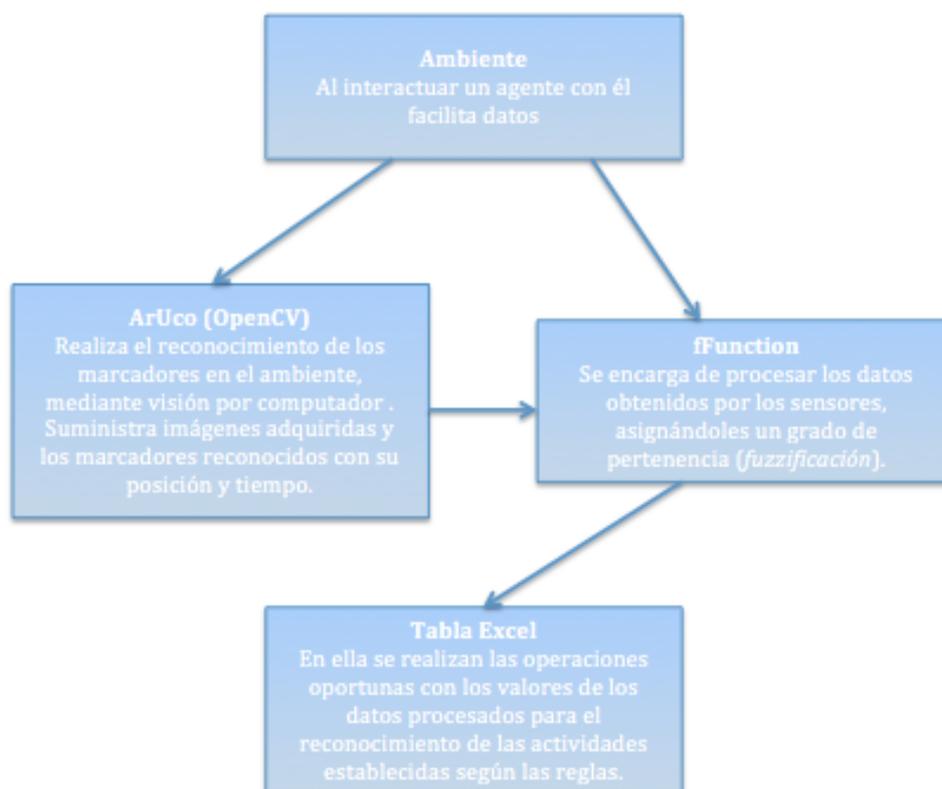


Figura A.1: Estructura de trabajo.

A.2. Programación

En esta sección, se va a presentar tanto el bloque de función utilizado para el procesado de datos como la programación implementada para el tratamiento de los mismos.

No se ha presentado el código de la adquisición mediante visión ya que la librería ArUco es **libre** y no se ha realizado ningún aporte a la misma.

A.2.1. *Function block* para fuzzificación de datos.

El lenguaje de programación utilizado para *fuzzificar* los valores obtenidos por medio de los sensores tras la actuación del agente ha sido Java™.

A continuación se presenta el código del *Function Block*:

```
// Block definition (there may be more than one block per file)
FUNCTION_BLOCK functions
// Define input variables
VAR_INPUT
    time : REAL;
    distance : REAL;
END_VAR

FUZZIFY time
    TERM recently := (0,1) (2,1) (4,0) ;
    TERM for_a_while := (0,1) (15,1) (25,0) ;
    TERM for_a_long_while := (0,1) (30,1) (45,0) ;
END_FUZZIFY

FUZZIFY distance
    TERM isClose := (0.0, 1.0) (2, 1.0) (2.5, 0.0);
END_FUZZIFY

END_FUNCTION_BLOCK
```

En primer lugar, se declaran las variables de entrada. En nuestro caso, sólo recibimos dos tipos de datos:

- Distancia: estos datos se reciben de los eBeacons.
- Tiempo: estos datos se reciben del resto de sensores.

En el caso de los sensores de movimiento, presenta el instante de tiempo en el que pasa de *FALSE* a *TRUE*, deja de estar en estado de reposo, detecta presencia, etc.,

Para los sensores de visión, se recibe el tiempo en el que se ha reconocido el marcador ArUco y el valor del identificador.

Por otro lado, el bloque *FUZZIFY* es el que se encarga de fuzzificar cada una de las variables de entrada. Dentro de este bloque se definen tantos términos lingüísticos como sean necesarios, explicados en la sección 4.3. A cada uno de estos términos lingüísticos se le asigna una función de pertenencia determinada mediante funciones lineales.

Los rangos de las funciones de pertenencia se fueron ajustando de forma experimental dependiendo de las necesidades de las reglas definidas.

A.2.2. Datos temporales

Se ha querido presentar un ejemplo de la rutina que procesa los datos obtenidos a partir de los sensores de tipo temporal.

```
package org.sensor;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.LinkedList;
import java.util.List;

import net.sourceforge.jFuzzyLogic.FIS;
import net.sourceforge.jFuzzyLogic.rule.LinguisticTerm;

public class CupBoardOpen {
    static public class T{
        public float fuzzy;
        public float stamp;
        public float value;
    public T(float fuzzy, float stamp, float value){
        this.fuzzy=fuzzy;
        this.stamp=stamp;
        this.value=value;
    }
}

    public List<T> opensensor;
```

```

public LinguisticTerm lV;
CupBoardOpen(String file,LinguisticTerm lV) throws IOException{
    this.opensensor=process(file);
    this.lV=lV;
}

public float calculateFuzzyTime(float t0, LinguisticTerm lT){
    double max=0.0;
    for(T v:this.opensensor){
        float incT=(t0-v.stamp);
        if(incT<0) break;
        float fT=(float)lT.getMembershipFunction().
            membership(incT);
        float fTV=Math.min(fT, v.fuzzy);
        max=Math.max(fTV, max);
    }
    return (float) max;
}

private List<T> process(String file) throws IOException{
    List<T> ret=new LinkedList<T>();

    BufferedReader br = null;
    String line = "";
    String cvsSplitBy = "\\t";
    br = new BufferedReader(new FileReader(file));
    while ((line = br.readLine()) != null) {

    }
    br.close();

    return ret;
}

static public void main(String [] args) throws Exception{
    FIS fis = FIS.load("fFunction.fcl", true);
        //Load from 'FCL' file

```

```

        LinguisticTerm lT=fis.getVariable("time").
            getLinguisticTerm("for_a_long_while");

        CupBoardOpen fmA=new CupBoardOpen
            ("scene1/CupBoardOpen.csv",lT);
        for(float t0=0.0f;t0<140f;t0++)
            System.out.println(t0+"\t"+fmA.
                calculateFuzzyTime(t0,lT));
    }
}

```

A.2.3. Datos distancia

A continuación se presenta un ejemplo de la rutina que procesa los datos obtenidos a partir de los sensores de proximidad.

```

package org.sensor;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.text.DecimalFormat;
import java.util.LinkedList;
import java.util.List;

import net.sourceforge.jFuzzyLogic.FIS;
import net.sourceforge.jFuzzyLogic.rule.LinguisticTerm;

public class BathBeacon {
    static public class T{
        public float fuzzy;
        public float stamp;
        public float value;
    }
    public T(float fuzzy, float stamp, float value){
        this.fuzzy=fuzzy;
        this.stamp=stamp;
        this.value=value;
    }
}

```

```
@Override
public String toString(){
    return "("+stamp+", "+value+"["+fuzzy+"]";
}
}

public List<T> beacon;
public LinguisticTerm lV;
public LinguisticTerm lT;
BathBeacon(String file,LinguisticTerm lV,
    LinguisticTerm lT) throws IOException{
    this.beacon=process(file);
    this.lV=lV;
    this.lT=lT;
    this.calculateFuzzy();
}

private void calculateFuzzy(){

    for(T v:this.beacon){
        v.fuzzy=(float)lV.getMembershipFunction().
            membership(v.value);
    }
}

public float calculateFuzzyTime(float t0){

    float max=0f;
    for(T v:this.beacon){
        float incT=(t0-v.stamp);
        if(incT<0) break;
        float fT=(float)lT.getMembershipFunction().
            membership(incT);
        float fTV=Math.min(fT, v.fuzzy);
        max=Math.max(fTV, max);
    }
    return max;
}
```

```
}

private List<T> process(String file) throws IOException{

    List<T> ret=new LinkedList<T>();

    BufferedReader br = null;
    String line = "";
    String cvsSplitBy = "\t";
    br = new BufferedReader(new FileReader(file));
    while ((line = br.readLine()) != null) {

        // use comma as separator
        String[] values = line.split(cvsSplitBy);

        ret.add(new T(0f,Float.parseFloat(values[0])
            ,Float.parseFloat(values[1])));
    }
    br.close();

    return ret;
}

static public void main(String [] args) throws Exception{
    FIS fis = FIS.load("fFunction.fcl", true);
    //Load from 'FCL' file
    LinguisticTerm lV=fis.getVariable("distance").
        getLinguisticTerm("isClose");

    LinguisticTerm lT=fis.getVariable("time").
        getLinguisticTerm("recently");

    BathBeacon fmA=new BathBeacon
        ("scene1/BathBeacon.csv",lV,lT);
//    for(T t:fmA.beacon)
    for(float t0=0.0f;t0<140f;t0++)
        System.out.println(t0+"\t"+fmA.
```

```

        calculateFuzzyTime(t0));
    }
}

```

A.3. Presentación de los datos obtenidos

La actuación del agente con los elementos del ambiente supone la generación de una serie de datos que posteriormente se procesan para la aplicación de la lógica difusa.

A.3.1. Sensor visión etapa 1

La información que se recibe de los sensores de visión se muestra en las columnas de la figura A.2 es:

- Las coordenadas en las que se encuentra el marcador.
- El fotograma al que corresponde dicha información. En este caso sería el mostrado en la figura A.3.
- El valor de lectura del marcador.
- La marca temporal en la que se toma la imagen.

Camera info:	Fotograma	Código	Marca temporal
{2.598851160611685, -2.081697431228223, -11.512888879748994}	00229	625	1445944594908
{2.927983026891322, -2.369748707648124, -11.651649365371087}	00230	625	1445944595168
{3.0894168591306643, -2.67706774125264, -11.747600230523524}	00231	625	1445944595288
{3.304712131028972, -2.9259822923935284, -11.994084128093823}	00232	625	1445944595503
{3.6559542722768095, -3.0453081843619145, -12.012344696905428}	00234	625	1445944595898
{3.0012618975271157, -1.9760198879678001, -11.814654282212146}	00249	540	1445944598862
{0.8256434179829499, -3.0904386442190437, -12.908160969436201}	00327	540	1445944615597
{0.8200022201612355, -3.0479095345316525, -12.91688310432672}	00331	540	1445944616394
{0.8039602812623411, -3.042645354722786, -12.763416114553742}	00332	540	1445944616589
{0.7978790561474481, -3.0501776679541117, -12.771925487393089}	00334	540	1445944616987
{0.806297388640966, -3.0046596872054847, -12.81263592185255}	00336	540	1445944617383
{0.7484672971876916, -2.771101064112283, -12.83183103631637}	00338	540	1445944617778
{0.8925310261542682, -2.6644719916963684, -11.748564185783072}	00497	625	1445944649623
{0.812509122352155, -2.726414948692193, -11.874758328154472}	00498	625	1445944649828
{0.8254996747084934, -2.7576019247895722, -11.985951657266673}	00499	625	1445944650016
{0.8527528976376962, -2.7312039868547413, -11.894088278560327}	00500	625	1445944650227

Figura A.2: Presentación datos sensores visión.

Se ha elegido el fotograma de la figura A.3 para mostrar que hay un segundo agente entrando en la escena¹ y llega un momento en el que en la presentación de los datos, figura A.2, cambia el código reconocido del marcador.

No se ha mostrado otra imagen en la se apreciara mejor la coexistencia de los agentes por la privacidad del mismo.

¹Se puede observar en la esquina interior izquierda de la figura A.3 como incide un segundo agente.



Figura A.3: Fotograma 229.

A.3.2. Sensor proximidad

Los *eBeacon* muestran una matriz de datos como se observa en la figura A.4.

Presenta:

- La marca temporal en la que ha sido detectada la baliza.
- La *Mac* corresponde con la identificación de los diferentes dispositivos que intervienen en el ambiente.
- La distancia muestra la longitud a la que se encuentra en cada momento el agente con respecto a la baliza.

Se ha escogido esta figura A.4 para poder observar, en la parte derecha, cómo aparece seguido el listado de dispositivos.

Time stamp	Mac	Distancia	Time stamp	Mac	Distancia
1445944538	CB:32:32:FB:13:FB	1.84561401	1445944670	CB:32:32:FB:13:FB	2.29819663
1445944539	CB:32:32:FB:13:FB	1.84561401	1445944672	CB:32:32:FB:13:FB	2.29819663
1445944540	CB:32:32:FB:13:FB	1.84561401	1445944673	CB:32:32:FB:13:FB	2.29819663
1445944542	CB:32:32:FB:13:FB	1.84561401	1445944674	CB:32:32:FB:13:FB	2.29819663
1445944543	CB:32:32:FB:13:FB	2.29819663	1445944580	CC:A6:B7:91:C7:76	0.5513369628
1445944544	CB:32:32:FB:13:FB	2.29819663	1445944581	CC:A6:B7:91:C7:76	0.5513369628
1445944545	CB:32:32:FB:13:FB	2.29819663	1445944582	CC:A6:B7:91:C7:76	0.5513369628
1445944546	CB:32:32:FB:13:FB	2.29819663	1445944583	CC:A6:B7:91:C7:76	0.5513369628
1445944547	CB:32:32:FB:13:FB	2.969439868	1445944585	CC:A6:B7:91:C7:76	0.5513369628
1445944548	CB:32:32:FB:13:FB	2.969439868	1445944586	CC:A6:B7:91:C7:76	0.5513369628
1445944549	CB:32:32:FB:13:FB	1.624434461	1445944587	CC:A6:B7:91:C7:76	0.5513369628
1445944550	CB:32:32:FB:13:FB	1.624434461	1445944588	CC:A6:B7:91:C7:76	0.5513369628
1445944551	CB:32:32:FB:13:FB	2.704118088	1445944589	CC:A6:B7:91:C7:76	0.5513369628
1445944552	CB:32:32:FB:13:FB	2.704118088	1445944590	CC:A6:B7:91:C7:76	0.5513369628
1445944554	CB:32:32:FB:13:FB	2.704118088	1445944601	CC:A6:B7:91:C7:76	1.298864213
1445944555	CB:32:32:FB:13:FB	2.704118088	1445944602	CC:A6:B7:91:C7:76	1.298864213
1445944556	CB:32:32:FB:13:FB	2.704118088	1445944603	CC:A6:B7:91:C7:76	1.298864213
1445944557	CB:32:32:FB:13:FB	2.704118088	1445944604	CC:A6:B7:91:C7:76	1.298864213
1445944558	CB:32:32:FB:13:FB	2.704118088	1445944605	CC:A6:B7:91:C7:76	1.09354947
1445944559	CB:32:32:FB:13:FB	2.704118088	1445944606	CC:A6:B7:91:C7:76	1.09354947
1445944560	CB:32:32:FB:13:FB	2.704118088	1445944607	CC:A6:B7:91:C7:76	1.09354947
1445944564	CB:32:32:FB:13:FB	5.379008963	1445944608	CC:A6:B7:91:C7:76	1.09354947
1445944565	CB:32:32:FB:13:FB	5.379008963	1445944609	CC:A6:B7:91:C7:76	1.09354947
1445944566	CB:32:32:FB:13:FB	5.379008963	1445944610	CC:A6:B7:91:C7:76	1.09354947
1445944568	CB:32:32:FB:13:FB	5.379008963	1445944611	CC:A6:B7:91:C7:76	1.09354947
1445944569	CB:32:32:FB:13:FB	5.379008963	1445944612	CC:A6:B7:91:C7:76	1.09354947
1445944570	CB:32:32:FB:13:FB	5.379008963	1445944613	CC:A6:B7:91:C7:76	1.09354947

Figura A.4: Capturas de datos recibidos por los *eBeacons*.

A.3.3. Sensor apertura y cierre y movimiento

Los sensores que recogen los datos de movimiento y de apertura y cierre, envían los datos en un *array* que consta de las siguientes partes:

1. **Sensor.** Muestra la codificación que se le ha dado a los elementos del sistema.
2. **Fecha/ Hora.** Establece la fecha y la hora a la que ha presentado un cambio de estado el sensor.
3. **Estado.** Este campo puede tomar los valores:
 - Movimiento. Sensores de movimiento detectan movimiento.
 - No movimiento. Sensores de movimiento dejan de detectar movimiento.
 - Abierta. Sensores de apertura abiertos.
 - Cerrada. Sensores de apertura cerrados.
4. **Descripción del sensor.** Indica el sensor al que se refiere.

Sensor	Fecha Hora	Estado	Descripción sensor
SM3	2017-04-28 9:42	Movimiento	Sensor movimiento baño
SM3	2017-04-28 9:42	No movimiento	Sensor movimiento baño
SM3	2017-04-28 9:42	Movimiento	Sensor movimiento baño
SM5	2017-04-28 9:42	Movimiento	Sensor movimiento sofá
SM2	2017-04-28 9:42	Movimiento	Sensor movimiento cama
D03	2017-04-28 9:42	Cerrada	Armario ropa
D03	2017-04-28 9:42	Abierta	Armario ropa
SM3	2017-04-28 9:42	No movimiento	Sensor movimiento baño
SM3	2017-04-28 9:42	Movimiento	Sensor movimiento baño
SM2	2017-04-28 9:43	No movimiento	Sensor movimiento cama
SM3	2017-04-28 9:43	No movimiento	Sensor movimiento baño
SM2	2017-04-28 9:43	Movimiento	Sensor movimiento cama
SM3	2017-04-28 9:43	Movimiento	Sensor movimiento baño
D03	2017-04-28 9:43	Cerrada	Armario ropa
SM1	2017-04-28 9:43	Movimiento	Sensor movimiento cocina
D03	2017-04-28 9:43	Abierta	Armario ropa
D10	2017-04-28 9:43	Abierta	Despensa
D06	2017-04-28 9:43	Abierta	Congelador
D06	2017-04-28 9:43	Cerrada	Congelador
D04	2017-04-28 9:43	Abierta	Armario Tazas
SM5	2017-04-28 9:43	No movimiento	Sensor movimiento sofá
D03	2017-04-28 9:43	Cerrada	Armario ropa
D04	2017-04-28 9:43	Cerrada	Armario Tazas

Figura A.5: Array de datos de apertura y cierre y de movimiento.

Bibliografía

- [1] IEEE Standards Association et al. Ieee standard for a smart transducer interface for sensors and actuators (ieee 1451.2), 1997.
- [2] A. Burns, R. Jacoby, and R. Levy. Psychiatric phenomena in alzheimer’s disease. iv: Disorders of behaviour. *The British Journal of Psychiatry*, 157(1):86–94, 1990.
- [3] F. J. Crosson. Inteligencia humana e inteligencia artificial. *Selección De Obras De Ciencia y Tecnología.*, 1975.
- [4] P. Droege. *Intelligent environments: spatial aspects of the information revolution.* Elsevier, 1997.
- [5] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280 – 2292, 2014.
- [6] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and R. Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481 – 491, 2016.
- [7] R. Hervás, J. Bravo, S. W. Nava, and G. Chavira. Interacción natural en ambientes inteligentes a través de roles en mosaicos de visualización. In *VII Congreso Internacional de Interacción Persona-Ordenador (INTERACCIÓN’2006)*, 2006.
- [8] G. C. La Delfa, S. Monteleone, V. Catania, J. F. De Paz, and J. Bajo. Performance analysis of visualmarkers for indoor navigation systems. *Frontiers of Information Technology & Electronic Engineering*, 17(8):730–740, 2016.
- [9] J. C. Llibre Guerra, M. A. Guerra Hernández, and E. Perera Miniet. Comportamiento del síndrome demencial y la enfermedad de alzheimer. *Revista Habanera de Ciencias Médicas*, 7(1):0–0, 2008.
- [10] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International journal of man-machine studies*, 7(1):1–13, 1975.

-
- [11] J. Medina, F. J. Quesada, and M. Espinilla. A lightweight distributed architecture to integrate fuzzy relevant objects in real-time environments. In *International Conference on Ubiquitous Computing and Ambient Intelligence*, pages 375–386. Springer, 2015.
- [12] H. Morillo, F. Pérez Maciá, and D. M. Jorquera. *Redes inalámbricas de sensores inteligentes. aplicación a la monitorización de variables fisiológicas*, 2013.
- [13] J. Peña-Casanova. *Activemos la mente. Barcelona. Fundación “La Caixa”*, 1999.
- [14] D. W. Rolston, A. Gama Pérez, and I. T. Ziskiend. *Principios de inteligencia artificial y sistemas expertos*. McGraw-Hill, 1990.
- [15] E. Salas, A. Cotero Gómez, O. Liévanos García, and C. Marcín. Patrón de consumo de alimentos en niños con y sin espectro autista.
- [16] L. Seymour. *Teoría de conjuntos y temas afines. Editorial Mc Graw Hill. México*, 1991.
- [17] C. Shewell, J. Medina-Quero, M. Espinilla, C. Nugent, M. Donnelly, and H. Wang. Comparison of fiducial marker detection and object interaction in activities of daily living utilising a wearable vision sensor. *International Journal of Communication Systems*, 30(5), 2017.
- [18] A. Steventon and S. Wright. *Intelligent spaces: The application of pervasive ICT*. Springer Science & Business Media, 2010.
- [19] L. Valverde. Razonamiento aproximado y lógica borrosa. *Aplicaciones de la lógica borrosa, Consejo Superior de Investigaciones Científicas*, pages 107–120, 1992.
- [20] L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information sciences*, 8(3):199–249, 1975.
- [21] L. A. Zadeh. Fuzzy logic= computing with words. *IEEE transactions on fuzzy systems*, 4(2):103–111, 1996.
- [22] L. A. Zadeh. “Fuzzy sets,” *Information and Control*, volume 8. pp.338–353, June 1965.
- [23] D. Zafra, J. Medina, L. Martínez, C. Nugent, and M. Espinilla. A web system for managing and monitoring smart environments. In *International Conference on Bioinformatics and Biomedical Engineering*, pages 677–688. Springer, 2016.