Navegación autónoma basada en visión estereoscópica y lógica difusa

Autor: María Paula Schiaffino

Director: Oscar García

Universidad Nacional de Luján
Int. Ruta 5 y 7
6700 Luján, Buenos Aires
República Argentina
Año 2006

Navegación autónoma basada en visión estereoscópica y lógica difusa

María Paula Schiaffino
Universidad Nacional de Luján
Int. Ruta 5 y 7
6700 Luján, Buenos Aires
República Argentina
pschiaffino@gmail.com

Resumen

El presente trabajo presenta la utilización de dos diferentes técnicas de procesamiento de imágenes con el objetivo de determinar la profundidad de un objeto en una escena tomada a través de visión estereoscópica. La finalidad de este procesamiento consiste en lograr la navegación de agentes autónomos hacia dicho objeto a partir del razonamiento incierto basado en reglas de lógica difusa, para luego corregir los movimientos de manera que el agente se dirija a la meta de forma más precisa.

Palabras claves:
Visión Estereoscópica, Lógica Difusa, Agentes Autónomos, Trigonometría, Sistemas de Control

Agradecimientos

A mi director de tesis, Profesor Oscar García por brindarme su tiempo y dedicación en este proyecto. A quienes colaboraron conmigo en este trabajo prestando desinteresadamente su tiempo y conocimientos. A mi familia y novio por acompañarme durante todos mis años de estudio y especialmente en esta etapa final de mi carrera.

Índice

INTRODUCCIÓN 1

1 FUNDAMENTOS DE VISIÓN ESTEREOSCÓPICA 3

1.1 Visión monocular y binocular 3
1.1.1 Disparidad binocular 4

1.2 Visión estereoscópica artificial 6
1.2.1 Adquisición de imágenes 6
1.2.2 Geometría del sistema estereoscópico y obtención de la distancia 6
1.2.3 Extracción de características y Correspondencia estereoscópica 8
1.2.3.1 Restricciones para la correspondencia 8
1.2.3.2 Técnicas de extracción de características y métodos de correspondencia 9

2 FUNDAMENTOS DE LÓGICA DIFUSA 11

2.1 Conjuntos Difusos 12
2.1.1 Función Membresía 14
2.1.1.1 Función Triangular 14
2.1.1.2 Función Trapezoidal 15
2.1.1.3 Función Gausiana 15
2.1.2 Operaciones sobre conjuntos difusos 16

2.2 Reglas si-entonces 17

2.3 Sistema de Inferencia Difuso 19
2.3.1 Proceso de inferencia difuso 20

2.4 Defusificación 23
2.4.1 Método defusificador Centroide 23
2.4.2 Otros métodos defusificadores 24

3 UBICACIÓN DEL OBJETIVO Y CÁLCULO DE LA DISTANCIA 25

3.1 Obtención de las coordenadas del objeto y cálculo de disparidad 26
3.1.1 Obtención de coordenadas del objeto 28
3.1.2 Ubicación en los cuadrantes de cada cámara 29
3.1.3 Cálculo de disparidad 30

3.2 Obtención de la distancia al objeto por Triangulación 32
3.2.1 Obtención de coordenadas del objeto 32
3.2.2 Cálculo de la distancia relativa al objeto 33
3.2.3 Cálculo del ángulo de giro 36

4 MODELO DIFUSO 38

4.1 Definición del Modelo Difuso 40
4.1.1 Operadores Lógicos 40
4.1.2 Método de Implicación 40

4.1.3 Método de Agregación 41
4.1.4 Método de Defusificación 41
4.1.5 Entradas Difusas 41
4.1.6 Salidas Difusas 42
4.1.7 Función Membresía 43
4.1.8 Reglas Difusas 44

4.2 Proceso de Inferencia Difusa (PID) 47
4.2.1 Entradas Difusas 47
4.2.2 Operadores Difusos 50
4.2.3 Inferencia Difusa 52
4.2.4 Agregación 55
4.2.5 Defusificación 56
4.2.6 Proceso de Inferencia Difusa 57

4.3 Ejemplos del PID 58

5 INTEGRACIÓN DE TÉCNICAS DE PROCESAMIENTO Y MODELO DIFUSO 61

5.1 Integración 61
5.1.1 Función de Activación de los Cuadrantes 62
5.1.2 Activación de los Efectores 63

5.2 Navegación autónoma 64
5.2.1 Secuencia de Percepciones y Acciones 64
5.2.2 Observaciones 73

5.3 Sistemas de Control para agentes autónomos 75
5.3.1 Context-Dependent Blending – CDB 76
5.3.2 Modelo de Visión Estereoscópica Difusa implementado en CDB 77

CONCLUSIÓN 79

Trabajos Futuros 80

ANEXO A: CMUCAM2 81

A.1 CMUcam2GUI 82
A.1.1 Panel Principal 82
A.1.2 Panel de Configuración 83
A.1.3 Panel de Color 84
A.1.4 Panel de Movimiento 85
A.1.5 Panel de Histogramas 86
A.1.6 Panel de Estadísticas 87
A.1.7 Panel de Servos 88

A.2 CMUcam2GUIStereo 89
A.2.1 Panel Principal 89
A.2.2 Panel de Configuración 90
A.2.3 Panel de Color 91
A.2.4 Panel de Color por Triangulación 92

A.3 Parámetros de las CMUcam2 93
A.3.1 Tolerancia 93
A.3.2 Noise Filter 93
A.3.3 Line Mode 1: Bitmap of Tracked Region 94
A.3.4 Line Mode 2: Track Color Stats 95
A.3.5 Auto gain y White Balance 96
A.3.6 Inverted Tracking, Track Window e YcrCb 96

ANEXO B: TOOLBOX DE LÓGICA DIFUSA 97
B.1 Editor FIS (Fuzzy Inference System) 98
B.2 Editor de Funciones Membresía 99
B.3 Editor de Reglas Difusas 100
B.4 Visor de Reglas Difusas 101
B.5 Visor de Superficie 102

BIBLIOGRAFÍA 103

Introducción

Este trabajo es una introducción al uso de la visión estereoscópica y la lógica difusa para ser utilizado en el desarrollo de comportamientos de agentes autónomos.

Se desarrollará un comportamiento natural para los seres humanos como es dirigirse hacia un objetivo determinado dentro del ambiente. Para un agente autónomo este comportamiento no es tan natural ya que debe distinguir el objeto en la escena y dirigir sus efectores de manera que lo lleven suavemente hacia la meta.

Este modelo está basado en dos conceptos biológicos: la visión estereoscópica que permite determinar la profundidad de un objeto a través de la disparidad y de la triangulación, y el razonamiento incierto o aproximado. A través de las reglas difusas es posible lograr un comportamiento flexible y una interpolación suave entre las acciones.

El propósito principal de este trabajo es presentar una alternativa de acción para lograr un comportamiento en agentes robóticos que puede ser utilizado para diferentes objetivos, como por ejemplo, dirigirse de un lugar a otro usando marcas en diferentes puntos intermedios, detectar un objeto que puede ser un intruso o una meta y llegar a él de manera precisa, obtener la distancia a un objeto determinado y calcular el tiempo de recorrido, entre otros.

Otro fin de este trabajo es que sirva como material de estudio para alumnos de Inteligencia Artificial donde podrán ver una forma de implementar los conocimientos adquiridos en la asignatura en un modelo concreto capaz de deliberar ante estímulos del ambiente.

El trabajo se divide en cinco capítulos. Los primeros dos se encargan de dar una base teórica para comprender cómo se logra conocer la profundidad de los objetos a través de la visión en estéreo y los fundamentos de la lógica difusa. Los últimos tres capítulos abordan el desarrollo del modelo propuesto y su integración.

En el capítulo 1 se darán los fundamentos de la visión estereoscópica: visión monocular y binocular, geometría del sistema estereoscópico, obtención de características y cálculo de disparidad.

En el capítulo 2 se plantean los fundamentos de la lógica difusa: definición de términos lingüísticos para diseñar conjuntos difusos, función membresía, reglas difusas, proceso de inferencia difusa y defusificación.

El capítulo 3 desarrolla las técnicas de procesamiento de imágenes para la detección del objetivo, ubicación del objeto en la escena, el cálculo de la disparidad y la obtención de la distancia relativa y el ángulo de desvío con respecto al centro del agente.

El capítulo 4 presentará el modelo difuso que se encargará de coordinar las acciones del agente de manera que éste se acerque al objetivo, utilizando para esto el razonamiento aproximado.

Por último, en el capítulo 5 se hará la integración de estos dos modelos explicando los pasos a seguir para lograr llegar al objetivo de manera adecuada y se dará un ejemplo de navegación en un ambiente con una meta determinada mostrando las percepciones del agente y sus acciones. Finalizando el capítulo se introduce una idea general de los sistemas de control basados en comportamientos y cómo este modelo puede implementarse en ellos.

Capítulo 1

1 Fundamentos de Visión Estereoscópica

Para percibir el mundo en forma tridimensional apreciando el volumen, profundidad y posición de los objetos, los humanos utilizamos la visión binocular o en estéreo. Esta consiste en que cada ojo recibe una imagen en dos dimensiones de una misma escena y al estar separados por una distancia que ronda entre los 4.5 a 7.5 cm, ambas imágenes difieren ligeramente entre sí, lo que se conoce como disparidad binocular. El cerebro, a través de mecanismos fisiológicos y psicológicos, procesa las diferencias entre las imágenes percibidas y las interpreta logrando fusionarlas en una única percepción donde podemos apreciar dimensiones, espesores, profundidades y distancias.

Euclides fue quien primero dejó constancia escrita de que percibimos el mundo en tres dimensiones, pero las primeras imágenes tridimensionales fueron posibles recién en 1838. Charles Wheaststone fue el pionero en la creación de estereogramas, haciendo públicos unos daguerotipos, es decir, placas metálicas de cobre bañadas con plata, en 1839. A partir de allí, se comenzaron a crear cámaras estereoscópicas y estereogramas. Ramón y Cajal desarrolló una técnica para obtener fotografías estereoscópicas con un solo objetivo de gran diámetro y Dalí realizó estereogramas tanto en fotografía como en pintura. Hoy en día la visión estereoscópica está más relegada a fines científicos, y la informática la utiliza para recrear escenas en tres dimensiones y crear lo que se conoce como "realidad virtual" [Marqués04].

1.1 Visión monocular y binocular

Cada ojo tiene un campo de visión propio, y solamente al frente ambos ojos ven en simultáneo. (Figura 1.1) Las imágenes percibidas por cada ojo corresponden a su propio campo visual y son planas o bidimensionales. Este tipo de visión se denomina monocular y también permite obtener cierta información

tridimensional a través de mecanismos geométricos, de paralaje y musculares [Marqués04].

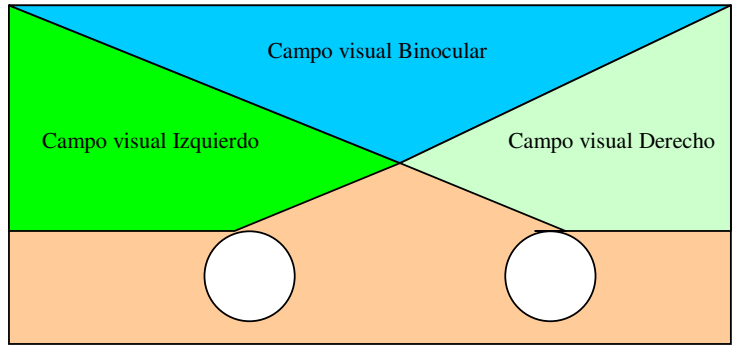


Figura 1.1 - Campo visual

La visión binocular consiste en la fusión de las dos imágenes capturadas por cada ojo en una sola imagen tridimensional, lograda a partir del método de disparidad binocular, que interpreta las diferencias entre ambas imágenes y nos permite percibir profundidad.

1.1.1 Disparidad binocular

Teniendo en cuenta que a cada punto de una retina le corresponde un punto de la otra, hecho que se conoce como puntos correspondientes o correspondencia [López], es posible comparar las imágenes recibidas por ambos ojos e interpretarlas. Esta comparación arrojará ciertas diferencias por el hecho de que cada ojo tiene su propio campo visual gracias a la separación existente entre ellos, la cuál se denomina técnicamente como línea base. Esta disparidad es la que permite obtener una sensación de tridimensionalidad y así percibir la profundidad de los objetos. El nivel de disparidad entre las imágenes depende del ángulo formado por los ejes ópticos que son las líneas rectas que unen los ojos con el objeto en el espacio (Figura 1.2). Además, este ángulo conocido como ángulo paraláctico o de convergencia o paralaje, tiene relación con la distancia relativa del objeto a los ojos. Cuanto más alejado se encuentre el objeto a los ojos, menor será el ángulo paraláctico y menor la disparidad entre las imágenes. En la figura 1.2 (a) puede verse que $\beta < \alpha$ ya que el rombo está más lejos que la cruz. Y en la figura 1.2 (b) donde se solapan los ojos se observa que la separación

relativa de los rombos es menor a la de las cruces. Habrá un punto en que la distancia del objeto a los ojos es tan grande, que no se percibirá disparidad en la visión binocular, y por lo tanto no se tendrá sensación de profundidad. Este umbral es diferente en cada persona y se conoce como poder de resolución estereoscópico o agudeza estereoscópica [Marqués04].

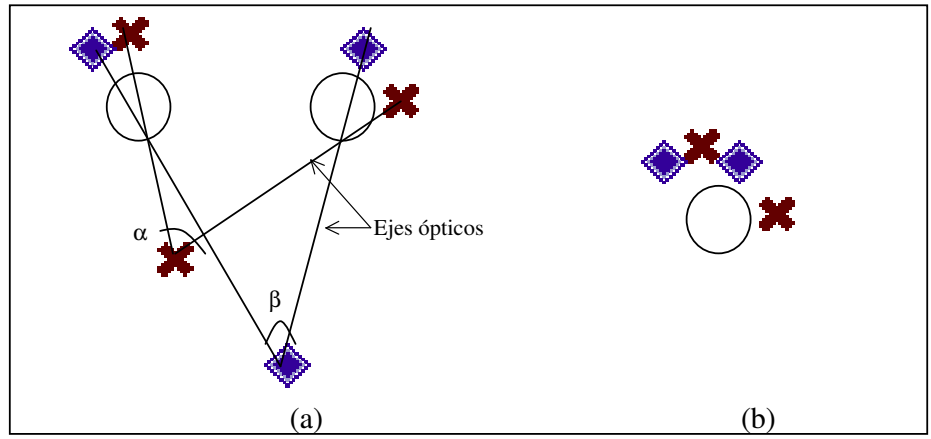


Figura 1.2 Disparidad binocular y ángulo paraláctico
(a) Visión estereoscópica donde se observa la relación entre la distancia de los objetos a los ojos y el ángulo paraláctico. (b) Solapamiento de las imágenes de ambos ojos donde se observa la disparidad binocular en relación con la distancia de los objetos a los ojos.

1.2 Visión estereoscópica artificial

El método biológico de visión estereoscópica es tomado por la inteligencia artificial como un procedimiento para la obtención de la forma de los objetos en la escena [Pajares02]. Utilizando dos cámaras separadas entre sí por cierta distancia llamada línea base, y con sus ejes ópticos paralelos se obtienen las dos imágenes monoculares, y luego, a través de procesamiento y técnicas como triangulación, se calcula la distancia de los objetos en el mundo tridimensional.

Los pasos a seguir en un proceso de visión en estéreo según Barnard y Fischler (1982), descritos con mayor detalle en [Pajares02] son los siguientes:

- Adquisición de imágenes
- Geometría del sistema
- Extracción de características
- Correspondencia de características
- Obtención de la distancia

1.2.1 Adquisición de imágenes

Las imágenes serán adquiridas a través de las cámaras, que como ya se ha dicho se encuentran separadas entre sí y tomarán una imagen cada una de la misma escena. Sobre dichas imágenes, se realizarán los siguientes pasos del proceso de visión estereoscópica, teniendo en cuenta que ambas deben ser tratadas de la misma manera.

1.2.2 Geometría del sistema estereoscópico y obtención de la distancia

El sistema estereoscópico estará compuesto por dos cámaras separadas por una distancia llamada línea base (b), simulando la distancia entre los ojos del sistema biológico, y con sus ejes ópticos paralelos entre sí y perpendiculares a b.

El centro de coordenadas de referencia es O siendo el eje Z igual a f, la longitud focal efectiva de las cámaras, y el eje X paralelo a la línea base. Para el mundo real los ejes de coordenadas serán iguales pero con centro en $b/2$. (Figura 1.3)

Los rayos de proyección PO_I y PO_D van desde el punto P(X,Y,Z) hasta el origen del sistema de coordenadas de referencia pasando por el plano de la

imagen. La intersección con este plano será la proyección de P en la imagen izquierda y derecha. El plano conformado por PO_I y PO_D es conocido como *plano epipolar*.

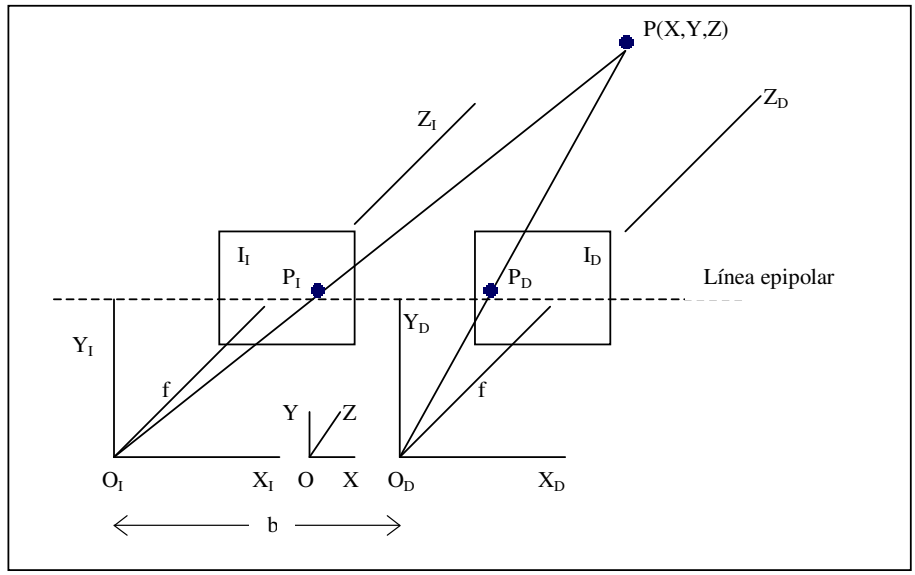


Figura 1.3 Geometría de dos cámaras estereoscópicas con ejes ópticos paralelos

Las *líneas epipolares*, es decir aquellas que unen la proyección izquierda y derecha del punto de la escena en los planos de imagen, son paralelas a la línea base. Estas mismas líneas son las originadas por la intersección entre el plano epipolar y el plano de imagen y definen el espacio de búsqueda dentro del cual se sitúan los puntos de emparejamiento, es decir los puntos que se corresponden entre las imágenes izquierda y derecha. Este espacio de búsqueda está definido por lo que luego se describirá con mayor detalle y se conoce como *restricción epipolar*. Son los puntos emparejados $P_I(x_I,y_I)$ y $P_D(x_D,y_D)$ los que permiten obtener una medida de comparación entre las imágenes monoculares que llamamos *disparidad* y se calcula $d = x_I - x_D$. Recordemos que las imágenes solo difieren horizontalmente, es por eso que el valor de y no se tiene en cuenta para obtener la disparidad.

Puede conocerse la profundidad Z del objeto de la escena 3D con la siguiente ecuación [Pajares02]:

$$\left.\begin{aligned} O_I &= \frac{\frac{b}{2}+X}{Z} = \frac{x_I}{f} \\ O_D &= -\frac{\frac{b}{2}+X}{Z} = \frac{x_D}{f} \end{aligned}\right\} \Rightarrow \left.\begin{aligned} x_I &= \frac{f}{Z}\left(X+\frac{b}{2}\right) \\ x_D &= \frac{f}{Z}\left(X-\frac{b}{2}\right) \end{aligned}\right\} \Rightarrow d = x_I - x_D = \frac{fb}{Z} \Rightarrow Z = \frac{fb}{d}$$

Esta ecuación surge de la figura 1.4 que no es más que el sistema de geometría de dos cámaras en estéreo vista desde otra perspectiva.

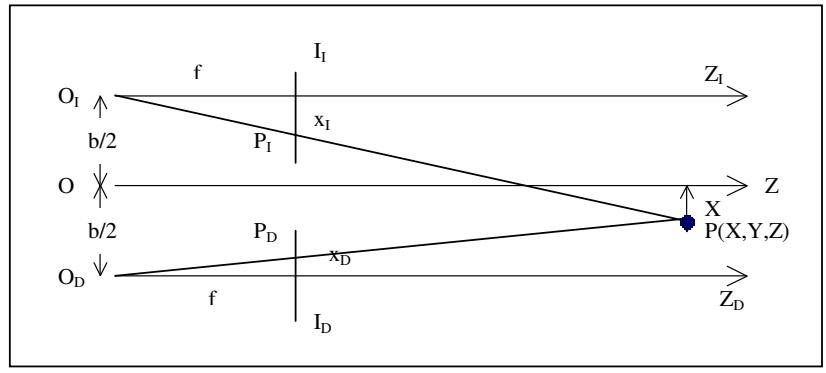


Figura 1.4 Geometría de dos cámaras estereoscópicas con ejes ópticos paralelos desde otra perspectiva

A partir de la ecuación se deduce que cuando mayor es la disparidad *d* entre las imágenes, menor es la profundidad *Z*, y a mayor distancia entre las cámaras (*b*), mayor la disparidad *d*.

1.2.3 Extracción de características y Correspondencia estereoscópica

La extracción de características consiste en procesar las dos imágenes del par estereoscópico y obtener de ellas información como intensidad de los píxeles, textura, elongación, bordes, áreas, etc. Luego, se realiza el proceso de correspondencia estereoscópica que permite determinar las proyecciones de un punto de la escena 3D en cada una de las imágenes del par estereoscópico. Para ello deben tenerse en cuenta ciertas restricciones basadas en la realidad física que se detallarán más adelante. Una vez determinadas las correspondencias de las proyecciones pueden compararse ambos puntos y obtener la disparidad entre las imágenes y conocer la profundidad del punto en el mundo real.

1.2.3.1 Restricciones para la correspondencia

- *Epipolar*: las proyecciones de un punto de la escena 3D deben encontrarse sobre la misma línea epipolar.
- *Semejanza*: las proyecciones de un punto de la escena 3D deben tener las mismas características y propiedades.

- *Unicidad*: para una proyección en una imagen debe existir una única proyección en la otra imagen, salvo oclusiones.
- *Orden posicional*: para dos proyecciones en una imagen que siguen un orden posicional (una a la derecha de la otra) debe cumplirse ese mismo orden para las proyecciones correspondientes en la otra imagen.
- *Continuidad de la disparidad*: las variaciones de disparidad en la imagen son usualmente suaves.
- *Relaciones estructurales*: los objetos, formados por aristas, vértices o superficies, tienen una estructura y disposición geométrica definida.

1.2.3.2 Técnicas de extracción de características y métodos de correspondencia

Para realizar la extracción de características de las imágenes existen dos tipos de técnicas.

- Basadas en el área
- Basadas en las características

Y según la técnica utilizada se lleva a cabo un método de correspondencia determinado. A continuación se introducen brevemente estas técnicas y métodos, los cuales pueden ser consultados por mayor detalles en [Pajares02].

1.2.3.2.1 Técnicas basadas en el área

Utilizan correlación cruzada entre la intensidad de los vecinos locales de un píxel de una imagen con la intensidad de los vecinos del píxel correspondiente en la otra imagen del par.

Al comparar cada par de píxeles correspondientes se logra un mapa de profundidad denso. Pero al utilizar como parámetro los valores de intensidad, estas técnicas son muy sensibles a la perspectiva y a la iluminación y contraste. Además se utilizan ventanas de correlación lo que puede originar falsas correspondencias debido a oclusiones.

Para determinar la correspondencia entre los puntos se tienen en cuenta las restricciones de semejanza, epipolaridad y orden posicional. Se establece una ventana de vecindad de MxN alrededor de un punto y se busca en la otra imagen

otro punto que le corresponda que tenga un nivel de intensidad similar (semajanza), que se encuentre en la línea epipolar (epipolaridad) y con sus vecinos en el mismo orden (orden posicional). Luego se puede calcular un coeficiente de correlación o la diferencia de intensidades entre los puntos de la ventana como propone Shirai (1987). Con estas medidas se puede obtener un mapa de disparidad y luego proceder al cálculo de la profundidad de los objetos de la escena.

1.2.3.2.2 Técnicas basadas en las características

Utilizan representaciones simbólicas de las imágenes de intensidad. Las características más utilizadas son puntos de borde aislados, cadenas de puntos de bordes y regiones delimitadas por bordes.

Al estar comparando características de las imágenes, estas técnicas son menos sensibles a contraste e iluminación, perspectiva, oclusiones y ruido. Por lo tanto son más robustas, exactas y rápidas. Por otro lado, deben seleccionarse primitivas confiables, y realizarse interpolación para obtener la profundidad de la escena ya que los mapas de profundidad no son muy densos.

En cuanto a los métodos de correspondencia, cuando se trabaja con técnicas basadas en características, primero se obtienen *correspondencias locales* que luego se verifican con *correspondencias globales*. Las locales consisten en la comparación de vectores de características teniendo en cuenta solo la restricción de semejanza. Luego, se tienen en cuenta también vecinos siendo esto correspondencia global. Esta técnica se conoce como *relajación* dentro de la cual pueden distinguirse procedimientos basados en procesos de *optimización*, como la red neuronal de Hopfield, y procedimientos de naturaleza *probabilista*.

Capítulo 2

2 Fundamentos de Lógica Difusa

"Todo es difuso en un grado que uno no se da cuenta hasta que trata de precisarlo." — Bertrand Russell

La lógica difusa hace referencia a un lógica multi-valuada, en la que los valores de verdad no son solo Verdadero y Falso como en la lógica tradicional, sino que pueden definirse conjuntos con varios valores posibles. Estas lógicas son utilizadas para modelar condiciones que no están claramente definidas o cuyos valores de verdad no son deterministas, dando la posibilidad de aproximarse al razonamiento humano. Es decir, que sirven para describir formalmente expresiones lingüísticas empleadas por el razonamiento natural como lo son "alto", "lento", "joven" y operar con ellas, dando resultados exactos a partir de datos imprecisos. El término "difuso" se refiere a que los valores de verdad tienen una connotación de incertidumbre, ya que es posible utilizar términos como "muy rápido", "rápido", "lento" y "muy lento", sin necesidad de precisar el valor exacto.

La lógica difusa surge formalmente de la mano de Lotfi Zadeh, quien describió la teoría matemática de los conjuntos difusos en su trabajo "Fuzzy sets" de 1965 [Zadeh65]. Anteriormente, Platón había indicado que existía una tercera región entre lo Verdadero y lo Falso. Fue Lukasiewicz quien propuso un tercer valor entre los anteriores al que llamó "posible", creando una alternativa a la lógica bi-valuada de Aristóteles. Luego, se observó que nada impedía que pudieran existir lógicas tri-valuadas, cuatro-valuadas o infinitamente valuadas.

Actualmente, la lógica difusa tiene una gran cantidad de aplicaciones, como la construcción de artefactos electrónicos de uso doméstico, procesamiento de imágenes, control de procesos industriales y control de consumo de combustible en cohetes de la NASA [Longstaff02].

2.1 Conjuntos Difusos

Un conjunto difuso es una extensión de los conjuntos nítidos, llamados en inglés "crisp sets". Para definir entonces un conjunto difuso, se definirá primero un conjunto nítido.

El *universo de discurso* es una colección de objetos de los que hablará una lógica determinada. Un *conjunto* es una colección de objetos pertenecientes al universo de discurso tal que es posible decir si un determinado objeto del universo pertenece o no a dicha colección.

Por ejemplo, dentro del universo de los números Naturales $\mathbb{N}$ se define el conjunto de los menores a 10, lo que se representa por extensión como:

$$A = \{1,2,3,4,5,6,7,8,9\} \subseteq \mathbb{N}$$

También puede representarse como un predicado, diciendo que un natural $n \in \mathbb{N}$ *menor_10*(n) es verdadero solo si n es un número dentro de los enumerados anteriormente en A.

$$A = \{ n \in \mathbb{N} \mid menor_10(n)\}$$

Otra representación posible da el valor de verdad *1* a todo elemento del universo que pertenezca a A y el valor *0* a todo aquel que no.

$$\mu_A = \mathbb{N} \rightarrow \{0,1\}$$

$$\mu_A(n) = \begin{cases} 1 & menor_10(n) \\ 0 & en \quad otro \quad caso \end{cases}$$

Esta representación constituye una *función de membresía* determinista, es decir, el elemento n pertenece o no pertenece al conjunto A. [Nauck98] (Figura 2.1)

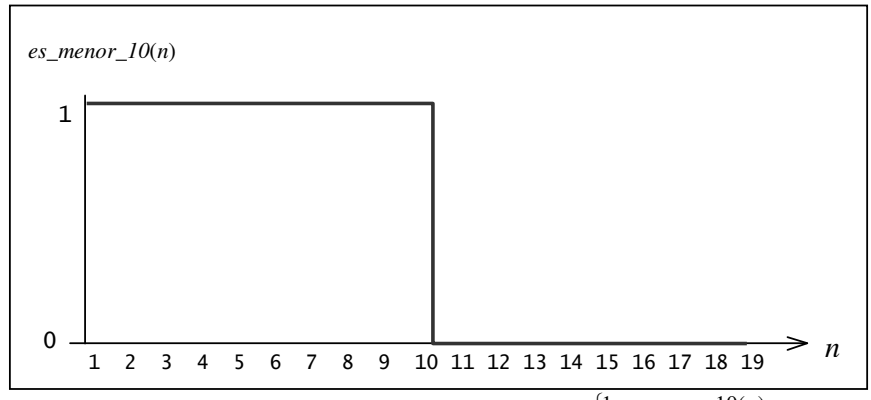


Figura 2.1 Conjunto nítido $\mu_A(n) = \begin{cases} 1 & menor_10(n) \\ 0 & en \quad otro \quad caso \end{cases}$

La teoría de conjuntos difusos extienden este concepto definiendo una *membresía parcial*. Es decir, que un elemento *x* del universo de discurso μ tendrá asociado un valor $y = \mu_A(x)$ que es el *grado de pertenencia* del elemento *x* al conjunto difuso *A* definido en el intervalo [0,1].

Por ejemplo, si se toma el universo de discurso μ de los pesos de las personas adultas femeninas, y el conjunto difuso *A* representa la etiqueta lingüística "flaco", se puede decir con seguridad que un peso de 40 kg. pertenece a ese conjunto. Esto sería que su grado de pertenencia es 1, o $\mu_A(40) = 1$. Al mismo tiempo podría asegurarse que un peso de 70 kg. no pertenece al conjunto "flaco", es decir, $\mu_A(70) = 0$. Pero para un peso de 60 kg. puede decirse que pertenece al conjunto *A* en un grado de 0.5 ($\mu_A(60) = 0.5$), ya que no es del todo seguro si pertenece o no al conjunto "flaco". (Figura 2.2)

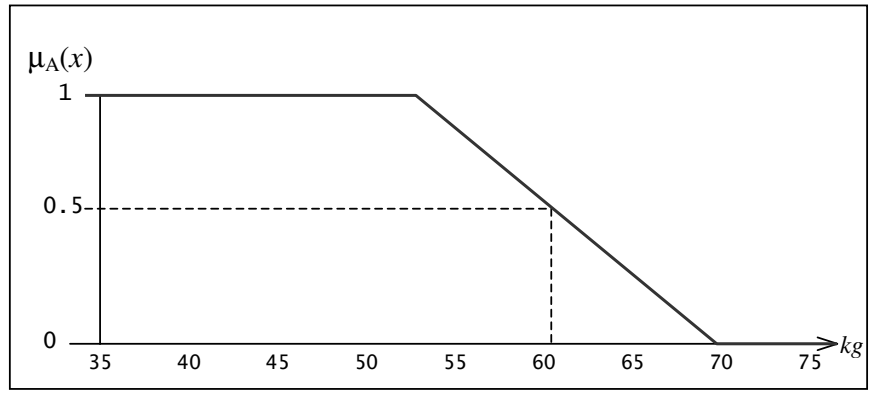


Figura 2.2 Conjunto difuso "flaco" del universo de personas adultas femeninas.

Entonces, los conjuntos difusos representan expresiones lingüísticas del lenguaje natural como flaco, gordo, alto, bajo, lento, rápido, joven, viejo, lejos, cerca, etc. Un elemento del universo puede pertenecer a más de un conjunto difuso al mismo tiempo con distinto grado de pertenencia.

Un conjunto difuso *A* del universo μ puede representarse como un conjunto de pares ordenados, en el que cada par consiste en un elemento genérico *x* y su grado de pertenencia a *A*.

$$A = \{(x, \mu_A(x)) \mid x \in \mu\}$$

Una variable lingüística *x* del universo μ se define como

$$T(x) = \{T_x^1, T_x^2, \ldots, T_x^k\}$$
$$\mu(x) = \{\mu_x^1, \mu_x^2, \ldots, \mu_x^k\}$$

donde $T(x)$ es el conjunto de términos de x, es decir, el conjunto de nombres de los valores lingüísticos de x, con cada T_x^i representando un número difuso cuya función de membresía μ_x^i está definida en μ [Kulkarni01]. En el ejemplo anterior, x indicaría pesos y $T(x)$ se referiría a los conjuntos flaco, gordo, normal.

2.1.1 Función Membresía

Una función membresía es una curva que define qué grado de pertenencia entre 0 y 1 le corresponde a cada punto del espacio de entrada.

La figura 2.1 muestra una función de membresía de un conjunto nítido, mientras que la figura 2.2 corresponde a una función de membresía de un conjunto difuso. Esta última permite observar una transición suave del valor de pertenencia entre los elementos que pertenecen y los que no pertenecen al conjunto A. Este concepto puede extenderse a muchos conjuntos difusos.

Existen varios tipos de funciones de membresía, a saber, triangular, trapezoidal, campana generalizada, Gausiana, sinosoidales, etc. Justamente la diferencia entre estas es la transición en el grado de pertenencia.

2.1.1.1 Función Triangular

Las curvas triangulares dependen de tres parámetros a, b y c y están definidas por:

$$f(x;a,b,c) = \begin{cases} 0 & para \quad x < a \\ \dfrac{x-a}{b-a} & para\ a \le x < b \\ \dfrac{c-x}{c-b} & para\ b \le x \le c \\ 0 & para \quad x > c \end{cases}$$

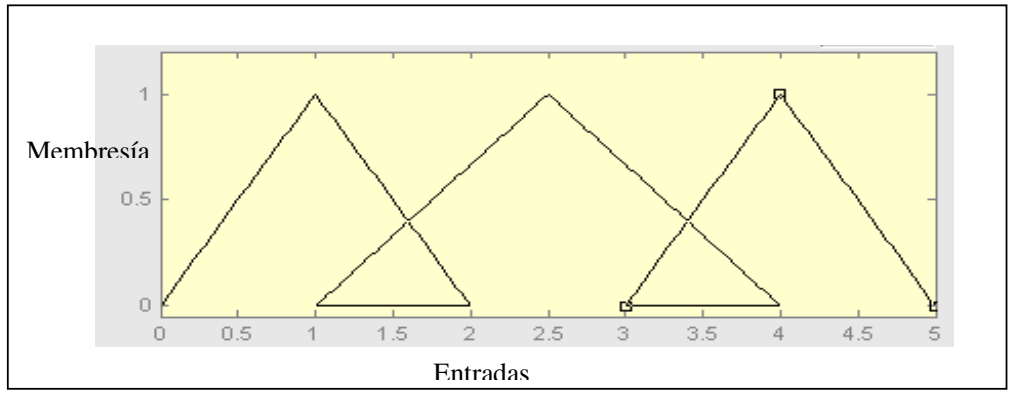


Figura 2.3 Curva triangular

2.1.1.2 Función Trapezoidal

Las curvas trapezoidales dependen de cuatro parámetros *a*, *b*, *c* y *d*.

$$f(x;a,b,c,d) = \begin{cases} 0 & para \quad x < a \\ \dfrac{x-a}{b-a} & para \quad a \le x < b \\ 1 & para \quad b \le x < c \\ \dfrac{d-x}{d-c} & para \quad c \le x < d \\ 0 & para \quad d \le x \end{cases}$$

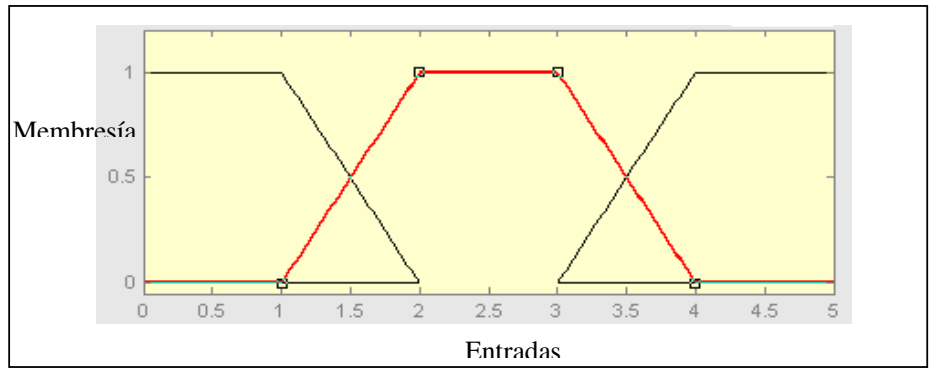


Figura 2.4 Curva trapezoidal

2.1.1.3 Función Gausiana

Las curvas Gausianas depende de dos parámetros, σ y *c* y se representa como:

$$f(x;\sigma,c) = \exp\left[\frac{-(x-c)^2}{2\sigma^2}\right]$$

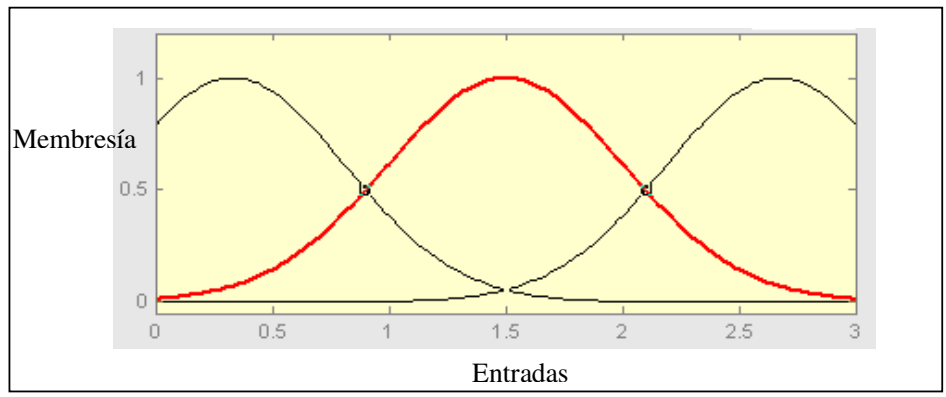


Figura 2.5 Curva Gausiana

2.1.2 Operaciones sobre conjuntos difusos

Las operaciones que se pueden realizar con conjuntos difusos son las mismas que se realizan con conjuntos nítidos. Las operaciones de Unión, Intersección y Complemento que se llevan a cabo en los conjuntos nítidos, se corresponden con OR, AND y NOT respectivamente en los conjuntos difusos (Figura 2.6).

Dados los conjuntos A y $B \in \mu$, se define Unión de A y B como $A \cup B$ y la componen todos los elementos que pertenecen a A o a B.

En conjuntos nítidos: $\mu_{A \cup B}(x) = 1 \; si \; x \in A \; o \; x \in B$

En lógica difusa: $\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)]$

La intersección entre A y B queda definida como $A \cap B$ y contiene a todos los elementos que estén en A y en B simultáneamente.

En conjuntos nítidos: $\mu_{A \cap B}(x) = 1 \; si \; x \in A \; y \; x \in B$

En lógica difusa: $\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)]$

El complemento de A se define como $\overline{A}$ y lo componen los elementos que no se encuentran en A.

En conjuntos nítidos: $\mu_{\overline{A}}(x) = 1 \; si \; x \notin A$

En lógica difusa: $\mu_{\overline{A}}(x) = 1 - \mu_A(x)$

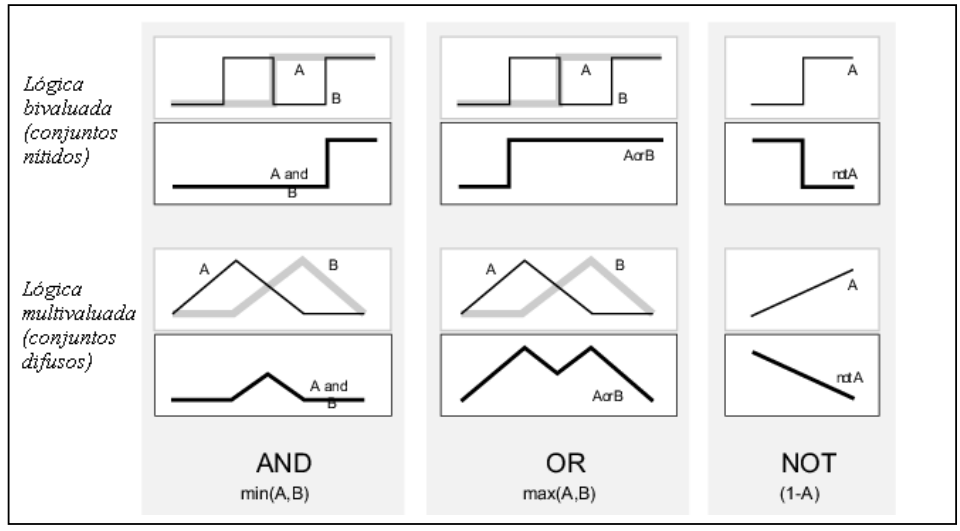


Figura 2.6 Operaciones sobre conjuntos nítidos y difusos [MathWorks01]

2.2 Reglas si-entonces

Las reglas if-then o sí-entonces son la base del sistema de inferencia difuso ya que relacionan los conjuntos difusos de entrada con los de salida.

Una regla básica tiene la forma: "SI *x* es *A*, ENTONCES *y* es *B*".
x pertenece al universo de discurso μ e *y* pertenece al conjunto de valores de verdad *V*. Tiene además una función membresía

$$\mu_{A\to B}(x, y)$$

cuyo valor pertenece al intervalo [0,1].

La parte izquierda de la regla "*x* es *A*" se denomina *antecedente* o *premisa* y la parte que le sigue al entonces – "*y* es *B*" – se conoce como *conclusión* o *consecuente*.

Para interpretar una regla IF-THEN, primero debe evaluarse el antecedente lo que implica fusificar las variables de entrada y aplicar cualquier operador difuso que sea necesario si fueran más de una variable (and, or). En otras palabras, se examina una variable lingüística de entrada ("temperatura") y su grado de pertenencia a un conjunto difuso ("temperaturas altas"). Esto produce un *grado de cumplimento* de cada regla [Romero00]. Luego, para obtener la salida de la regla que es un conjunto difuso completo, se evalúa la implicación entre el grado de cumplimiento y la función de pertenencia. Es decir, se aplica el resultado del antecedente al consecuente lo que consiste específicamente en evaluar la función de membresía. Por lo tanto, el consecuente será cierto en el mismo grado en que lo es el antecedente, ya que el conjunto difuso de salida es truncado según el método de implicación utilizado.

Existen varias formas de evaluar una función de membresía $\mu_{A\to B}(x, y)$ de una regla IF-THEN, es decir, existen varios métodos de implicación, a saber:

- Zadeh (1965)
$$\mu_{A\to B}(x, y) = \max(\min(\mu_A(x), \mu_B(y)), 1 - \mu_A(x))$$
- Mamdani (1977)
$$\mu_{A\to B}(x, y) = \min(\mu_A(x), \mu_B(y))$$
- Larsen (1980)
$$\mu_{A\to B}(x, y) = \mu_A(x).\mu_B(y)$$

Un caso genérico de una regla de implicación es:

R_i = *If* (x_1 *is* T_{x1}. *and…,* x_p *is* T_{xp}) *Then* (y_1 *is* T_{y1}, *and…,*y_p *is* T_{yp})

Las reglas SI-ENTONCES ampliadas cuya forma general es "*si…entonces…sino…*" pueden interpretarse como sigue:

$$\mu_{A\to B,C}(x,y,z) = \max(\min(\mu_A(x),\mu_B(y)),\min(1-\mu_A(x),\mu_C(z)))$$

2.3 Sistema de Inferencia Difuso

Un sistema de inferencia difuso (SID) o fuzzy inference system (FIS) define una proyección no lineal entre el vector de variables de entrada y el escalar de salida, a través de la utilización de reglas difusas. El proceso de proyección involucra funciones de membresía de entrada y salida, operadores lógicos, reglas if-then, agregación de conjuntos de salida y defusificación. Todo esto está incluido en los cuatro componentes de un SID (Figura 2.7):

- *bloque fusificador*: toma las variables de entrada y determina el grado de pertenencia a cada conjunto difuso a través de funciones de membresía.
- *base de reglas*: contiene las reglas lingüísticas if-then.
- *motor de inferencia*: realiza la proyección entre los conjuntos difusos de entrada y los de salida. Para esto, determina el grado de cumplimiento del antecedente de cada regla. Si el antecedente tuviera más de una cláusula se utilizan los operadores lógicos para obtener un número representativo del resultado de la regla. Es posible que se disparen varias reglas al mismo tiempo, en ese caso, deben agregarse. Es decir, se toman los conjuntos difusos de salida de las reglas disparadas paralelamente y se combinan obteniendo un único conjunto difuso de salida.
- *defusificador*: toma el conjunto difuso de salida agregado y da como resultado un número nítido. Esto es posible a través de varios métodos de defusificación, como el centro de gravedad, máximo, media del máximo y height.

También es posible, y por lo general se realiza, agregar dos bloques más, uno de pre-procesamiento y otro de post-procesamiento. Estos permiten, por ejemplo, la adaptación de las entradas para el control digital difuso a través de muestreo y retención. También realizan filtrado de ruido de medición y derivación o integración de la señal de entrada.

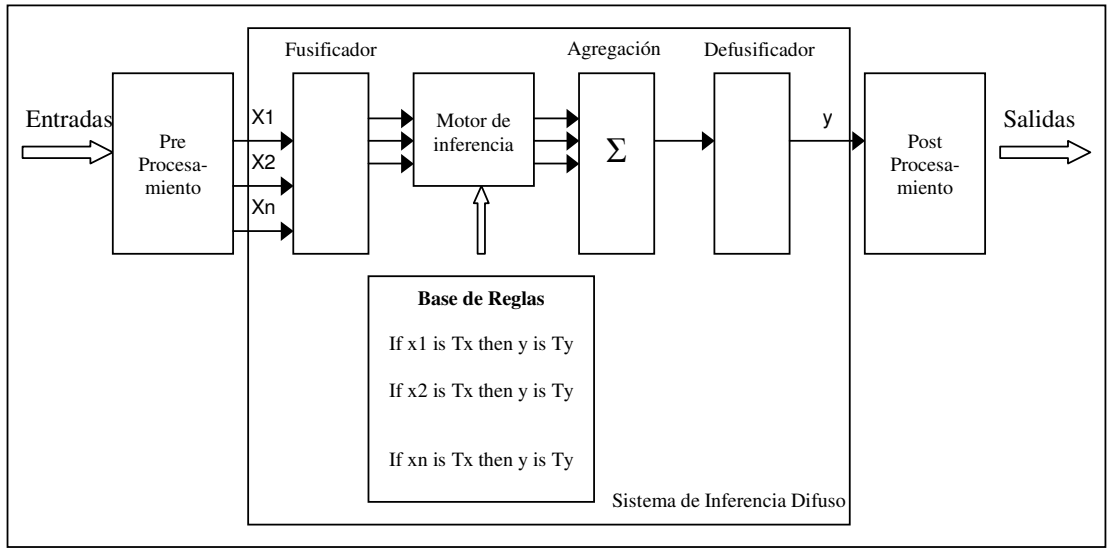


Figura 2.7 Sistema de inferencia difuso con pre y post procesamiento

2.3.1 Proceso de inferencia difuso

El proceso de inferencia difuso (PID) consiste en cinco pasos que se describirán a continuación y define una superficie de control $y = f(x_1,x_2)$ que se muestra en la figura 2.8.

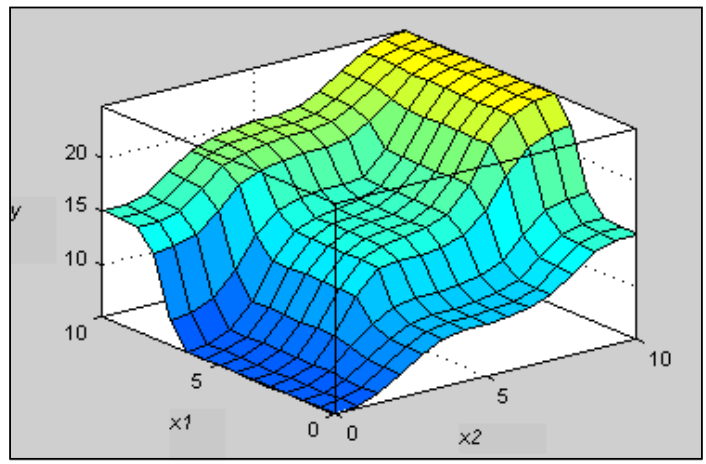


Figura 2.8 Superficie de control $y = f(x_1,x_2)$

Paso 1: Fusificación de entradas

Consiste en tomar las variables de entrada y determinar el grado de pertenencia a cada conjunto difuso a través de la función membresía.

Paso 2: Aplicación de operadores difusos

Con las variables ya fusificadas se conoce el grado de cumplimiento que otorga esa parte del antecedente a toda la regla. Si una regla tiene más de una cláusula, se utilizan los operadores difusos que correspondan como and y or para obtener el grado de cumplimiento de la regla compuesta.

Paso 3: Aplicación del método de implicación

Teniendo como entrada el grado de cumplimiento del antecedente, se procede a obtener la salida de aquellas reglas que fueron disparadas (es decir, cuyo grado de cumplimiento es mayor a 0). El método de implicación utilizado puede ser cualquiera de los descritos en la sección 2.2, siendo los más comunes el del producto (Larsen) y el del mínimo (Mamdani). Se obtiene como resultado un conjunto difuso de salida.

Paso 4: Agregación de salidas

Consiste en la unificación de las salidas de las reglas disparadas. Ocurre solo una vez por cada variable de salida. La entrada para este proceso de agregación son los conjuntos de salida obtenidos en el paso anterior y su resultado es un único conjunto difuso de salida que será utilizado luego por el paso de defusificación. Este proceso es conmutativo, es decir, no importa el orden en que las reglas fueron ejecutadas. El método de agregación mayormente utilizado es el del máximo definido como:

$$\mu_y(w) = \max(\mu_y^1(w), \mu_y^2(w))$$

donde $\mu_y^1(w)$ y $\mu_y^2(w)$ son dos conjuntos difusos de salida y $\mu_y(w)$ es el conjunto único de salida resultado de este paso.

Paso 5: Defusificación

El proceso de defusificación tiene como entrada el conjunto difuso de salida agregado obtenido en el paso anterior. Su resultado es un número nítido que se determina a través de uno de los varios métodos de defusificación posibles, como el centro de gravedad, máximo, media del máximo y height.

En la figura 2.9 pueden observarse estos cinco pasos a través de un ejemplo obtenido de [MathWorks01] con dos variables de entrada, una de salida y tres reglas difusas.

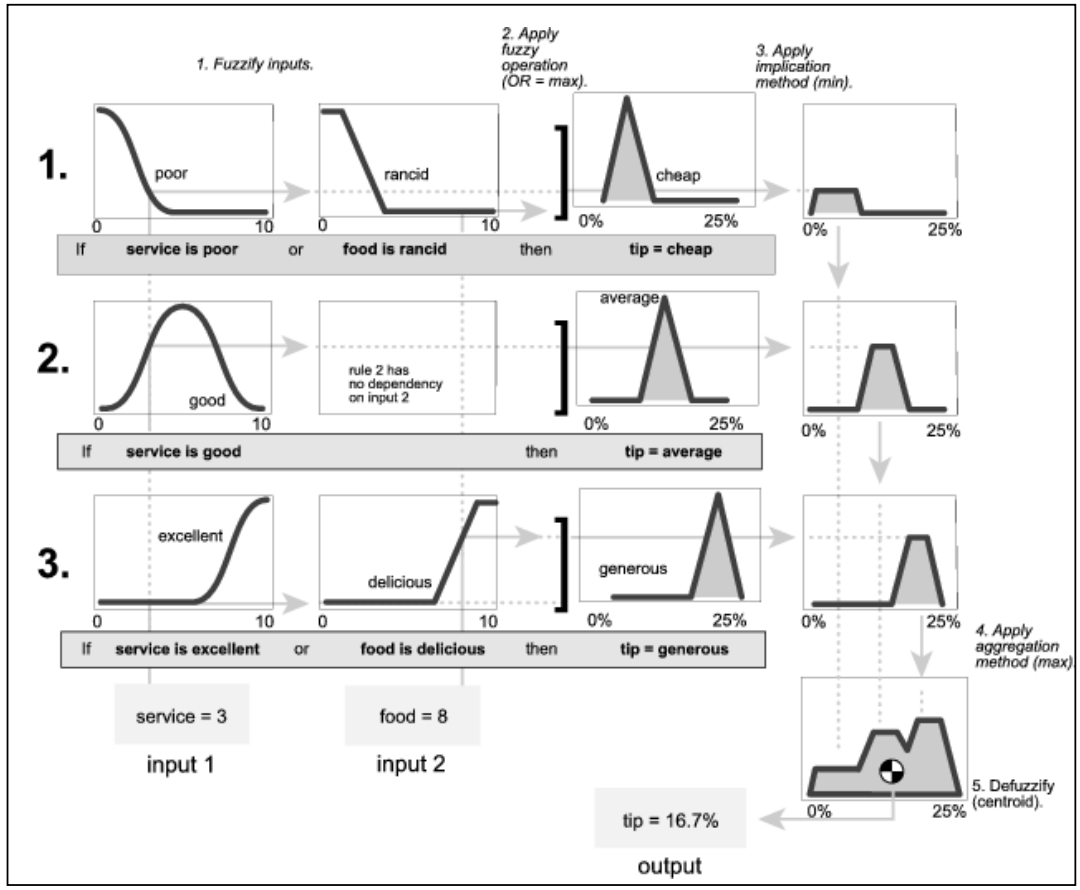


Figura 2.9 Proceso de Inferencia Difuso [MathWorks01]

2.4 Defusificación

El proceso defusificador es el último paso del proceso de inferencia difusa y consiste en obtener un número nítido que se utilizará como acción de control a partir del conjunto difuso retornado por el proceso de agregación. Los métodos de defusificación propuestos son varios, pero el más utilizado es el del Centroide. Los otros son el máximo, media del máximo y height.

2.4.1 Método defusificador Centroide

Este método defusificador determina el centro de gravedad o centroide y_i' de B, siendo B el conjunto difuso agregado, y lo utiliza como valor de salida del sistema de inferencia difusa.

El centroide está dado por

$$y' = \frac{\int_s y_i \mu_B(y) dy}{\int_s \mu_B(y) dy}$$

donde S es la superficie $\mu_B(y)$. Para obtener una aproximación de y' se suele utilizar variables discretas y la operación de sumatoria en lugar de la integral.

$$y' = \frac{\sum_{i=1}^{n} y_i \mu_B(y_i)}{\sum_{i=1}^{n} \mu_B(y_i)}$$

Este método defusificador encuentra un punto de balance en la región difusa que sea la solución calculando el peso medio de la región de salida difusa (Figura 2.10).

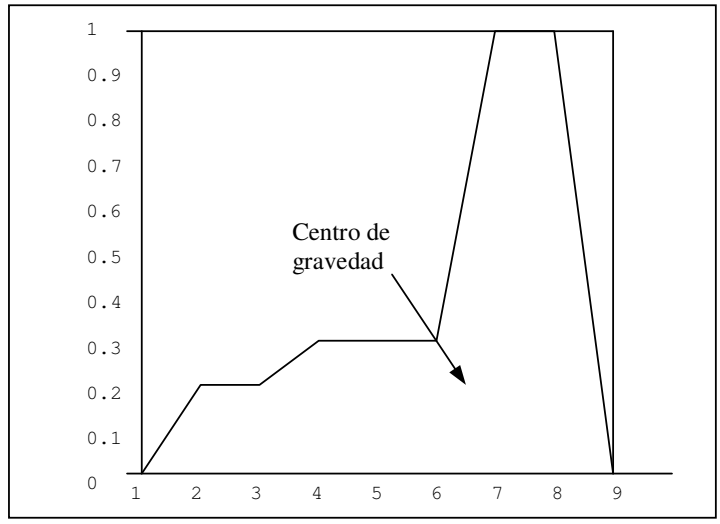


Figura 2.10 Centro de gravedad en el conjunto difuso de salida

2.4.2 Otros métodos defusificadores

El método del máximo busca el punto *y* del conjunto de salida agregado para el cual $\mu_B(y)$ sea el máximo (Figura 2.11).

El método de la media de los máximos busca el punto medio entre los centros de las cimas más altas de la región difusa de salida (Figura 2.12).

El método Height está definido por

$$y_h = \frac{\sum_{i=1}^{m} y_i' \mu_B(y_i)}{\sum_{i=1}^{m} \mu_B(y_i)}$$

donde m representa el número de conjuntos de salida difusos obtenidos después de la implicación y y' representa el centroide de la región difusa *i*.

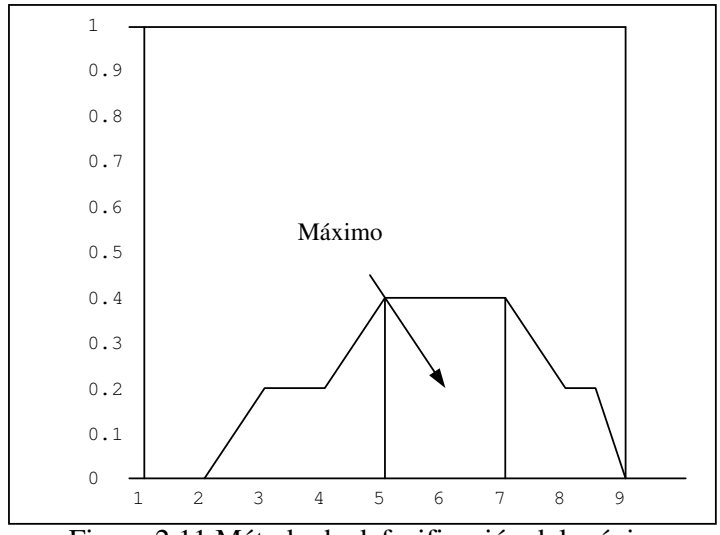


Figura 2.11 Método de defusificación del máximo

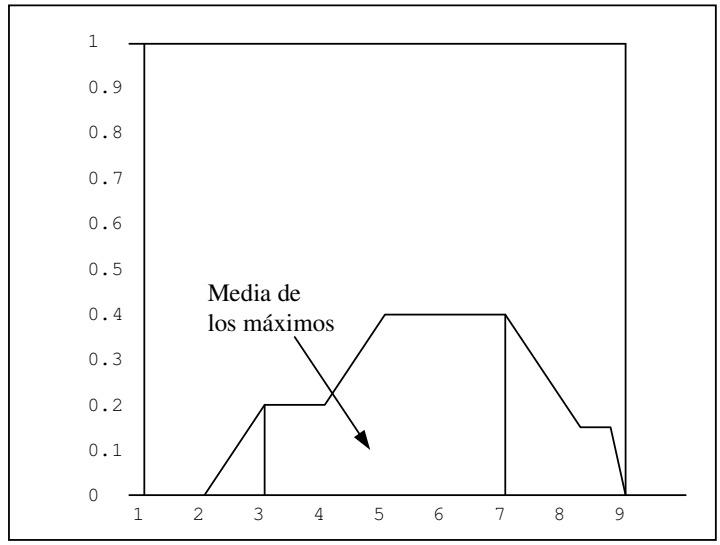


Figura 2.12 Método de defusificación de la media de los máximos

Capítulo 3

3 Ubicación del Objetivo y Cálculo de la Distancia

Un *agente móvil* se define como "todo aquello que puede considerarse que percibe su ambiente mediante sensores y que responde o actúa en tal ambiente por medio de efectores" [Russell96].

Cuando hablamos de agentes móviles robóticos estamos refiriéndonos a dispositivos mecánicos que, al igual que los agentes móviles biológicos como los animales, interactúan con su ambiente a través de los sensores y efectores que poseen (Figura 3.1).

Los sensores son aquellos órganos o dispositivos que le permiten al agente percibir el estado del mundo. Por ejemplo, los ojos, nariz, oído en los agentes biológicos o las cámaras de video, sensores de tacto, sonar en los agentes robóticos.

Los efectores, en cambio, le permiten al agente interactuar y modificar el ambiente, por ejemplo, las patas, la voz en los agentes biológicos o los motores con ruedas, síntesis de voz en los agentes robóticos.

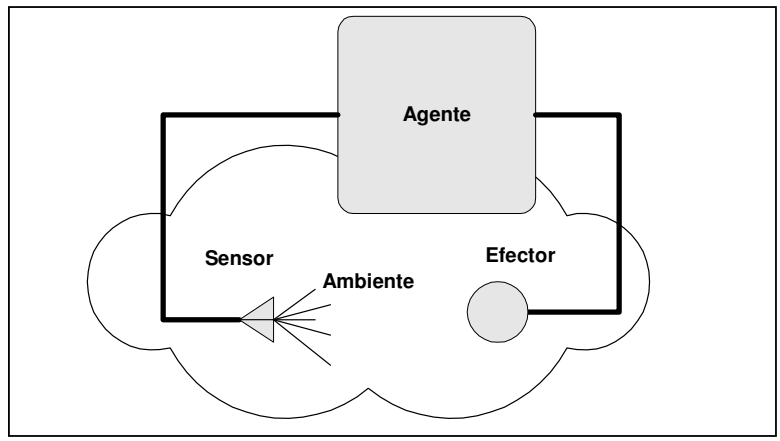


Figura 3.1 Sensores y efectores de un agente móvil en relación con el ambiente

Para lograr la navegación de un agente robótico es primordial detectar la presencia de objetos en el ambiente. Estos objetos, según la aplicación, pueden

representar obstáculos que deben ser esquivados o bien objetivos a los cuales debe dirigirse el robot.

En el presente trabajo el agente deberá encontrar en la imagen capturada a través de sus cámaras de video un objeto de características de color bien contrastantes con el fondo que representará un objetivo al cual debe dirigirse.

Como sensores del agente robótico se utilizarán dos cámaras de video conocidas como CMUcam2 desarrolladas por Robotics Institute de Carnegie Mellon University [RoweVS03]. La resolución de las imágenes adquiridas por las cámaras es de 88x143 píxeles y el ángulo de visión de cada una de ellas es de 40º. Para simular la visión estereoscópica, las mismas se disponen en forma paralela al eje central de visión y separadas por una línea base L de 6,5 cm. Para poder obtener la disparidad entre las imágenes de cada cámara y lograr así conocer la distancia del objeto al agente móvil, es necesario ubicar el objeto en la imagen. Para ello, la misma se dividirá en 5 cuadrantes de 8º cada uno. (Figura 3.2).

Para la obtención de la distancia del agente móvil al objeto se utilizarán las funcionalidades del módulo de procesamiento provisto por las CMUcam, las cuales se describirán en detalle a continuación en el presente capítulo. En primera instancia se describirá la obtención de una medida de disparidad que dará idea de la distancia real del objeto en el ambiente y que se utilizará luego como entrada al modelo difuso propuesto y descrito en el capítulo 4 usado para guiar al robot hacia el objetivo. Y por último se detallará el cálculo de la distancia relativa al objeto a través de triangulación que será utilizado cuando se determine que el objeto está cerca del robot, para mejorar la precisión de los movimientos.

3.1 Obtención de las coordenadas del objeto y cálculo de disparidad

Como se ha dicho anteriormente, el objeto que debe identificarse en la imagen posee un color contrastante respecto al ambiente para que pueda ser distinguido. Las CMUcam2 proporcionan un comando llamado *Track Color* cuyo objetivo es apartar un color determinado de una imagen y extraer información sobre la ubicación de una región en la imagen que contenga justamente ese color [RoweVS03].

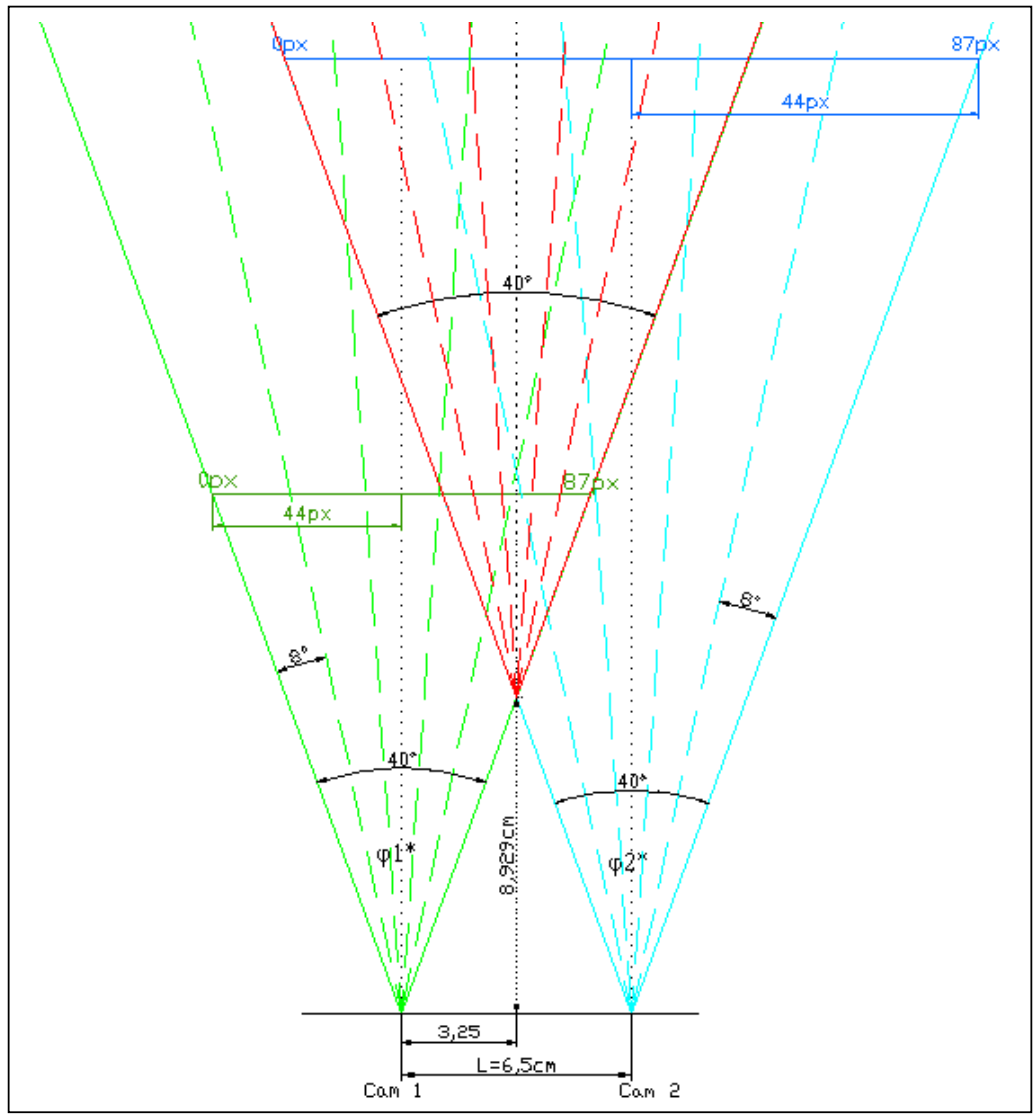


Figura 3.2 Visión estereoscópica con dos cámaras

Cada color es representado unívocamente por un valor de rojo, verde y azul que indica cuánto de cada canal se encuentra mezclado para formar el color final. Esto se conoce como valor RGB (*R*ed, *G*reen, *B*lue). Para que las cámaras puedan identificar un color, es necesario definir un valor mínimo y un valor máximo permitido para cada uno de los tres canales de colores. Todos los colores formados por esa combinación de valores serán los que se buscarán detectar en la imagen. Estos límites se definen para trabajar con un margen que tenga en cuenta las variaciones de luz y la no uniformidad de los objetos.

Una vez definidos los límites, se lleva a cabo el proceso de búsqueda que consiste en un algoritmo simple aplicado a cada cuadro independientemente. A grandes rasgos, el algoritmo examina secuencialmente y comenzando en la

esquina superior izquierda de la imagen, cada píxel fila por fila. Si el píxel que está evaluando pertenece al rango definido como el color deseado, lo marca como válido. Además, se evalúa la posición del píxel para saber si se trata del más alto, del más bajo, del más a la izquierda o el más a la derecha de los píxeles ya rastreados hasta el momento. Si se trata de un píxel ubicado afuera de los límites del objeto hasta allí calculados, entonces se expanden dichos límites de forma que lo contengan. De esta manera se va formando un rectángulo que demarca la zona donde se encuentra un objeto del color especificado. Finalmente, cuando todos los píxeles del cuadro hayan sido examinados, se calcula la posición del objeto.

3.1.1 Obtención de coordenadas del objeto

La Interfaz de Usuario proporcionada por las cámaras llamada CMUcam2GUI la cual fue adaptada para esta aplicación aprovechando su condición de software libre de código abierto, permite observar en el *Color Panel* la información otorgada por el comando Track Color.

El cuadro que demarca la zona de ubicación del objeto es representado por un rectángulo azul y el centro del mismo es mostrado con un punto rojo. Se pueden observar también las coordenadas del centroide como un par ordenado (x,y) donde el valor de *x* es el píxel de ubicación horizontal que se encuentra entre 0 y 87, y el valor de *y* representa la ubicación en forma vertical y se encuentra entre 0 y 142. El valor de *x* es el que interesa a los fines de determinar la distancia al objeto. También se muestran las coordenadas de los vértices superior izquierdo e inferior derecho del rectángulo que encierra al objeto, para poder tener una idea de la dimensión del mismo.

En la figura 3.3 se muestran los resultados del comando Track Color (b) aplicado al objeto rojo de la imagen (a).

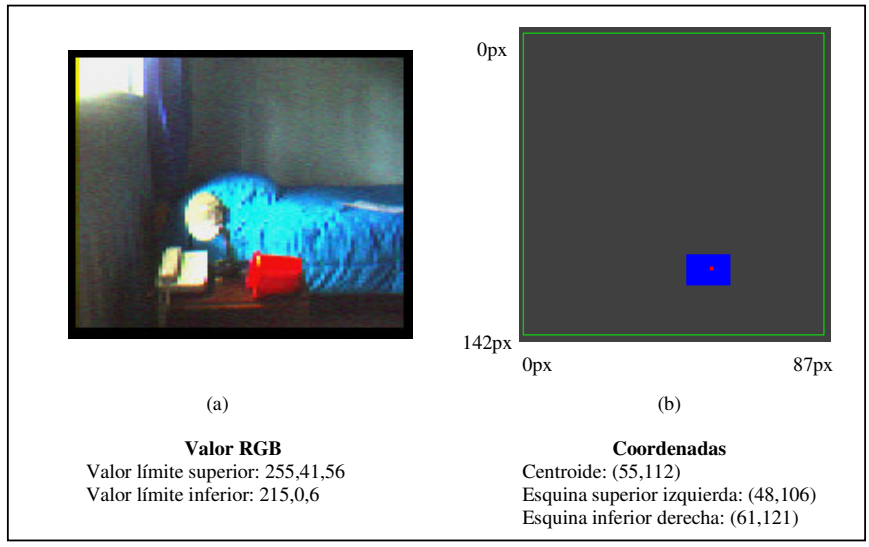


Figura 3.3 Obtención de coordenadas el objeto a través del comando Track Color (a) Imagen original. (b) Resultado de aplicar Track Color a la imagen original.

3.1.2 Ubicación en los cuadrantes de cada cámara

Como se dijo anteriormente, las imágenes se encuentran divididas en 5 cuadrantes de 8º cada uno. Esta división también puede efectuarse en base a los píxeles horizontales de la misma por lo que quedan definidos los siguientes límites para cada cuadrante:

Cuadrante	Píxel Mínimo	Píxel Máximo
1	0	17
2	18	34
3	35	51
4	52	68
5	69	87

Conociendo las coordenadas del objeto, y teniendo en cuenta especialmente el valor *x* del par ordenado que representa la posición horizontal del objeto, se puede determinar qué cuadrantes estarán activados. En este caso se tienen en cuenta los valores de los vértices del rectángulo que envuelve al objeto, y se activarán en forma consecutiva aquellos cuadrantes que estén incluidos en los cuadrantes límites. Por ejemplo, si las coordenadas de la esquina superior izquierda es (33,25) se activaría como cuadrante límite inicial el cuadrante 2, y si

la esquina inferior derecha es (58,65) se activaría como cuadrante límite final el número 4. Por lo tanto los cuadrantes activados de la imagen para esa cámara serían los número 2, 3 y 4.

En el ejemplo de la figura 3.3 los cuadrantes activados fueron el número 3 y 4. Esto se representa como el conjunto de valores booleanos (0,0,1,1,0) cuya posición indica el número de cuadrante, el valor 0 indica que el objeto no se encuentra en ese lugar y el valor 1 indica su presencia en correspondiente cuadrante.

3.1.3 Cálculo de disparidad

Los datos obtenidos anteriormente, es decir, las coordenadas del objeto y la activación de los cuadrantes, deben calcularse para ambas imágenes obtenidas una con cada cámara del modelo de visión estereoscópica.(Figura 3.4)

Luego se calcula la disparidad entre ambas imágenes. Para ello se realiza la siguiente ecuación:

$$Disparidad = \frac{\sum_{i=1}^{n} ABS(CI_i - CD_i)}{n}$$

Es decir, la sumatoria de las diferencias absolutas de cada par de cuadrantes dividido la cantidad total de cuadrantes.

Para el ejemplo de la figura 3.4 donde tenemos *CI*=(0,0,0,1,0) y *CD*=(0,0,1,1,0) la disparidad es:

$$\frac{\text{ABS}(0 - 0) + \text{ABS}(0 - 0) + \text{ABS}(0 - 1) + \text{ABS}(1 - 1) + \text{ABS}(0 - 0)}{5} = 0{,}2$$

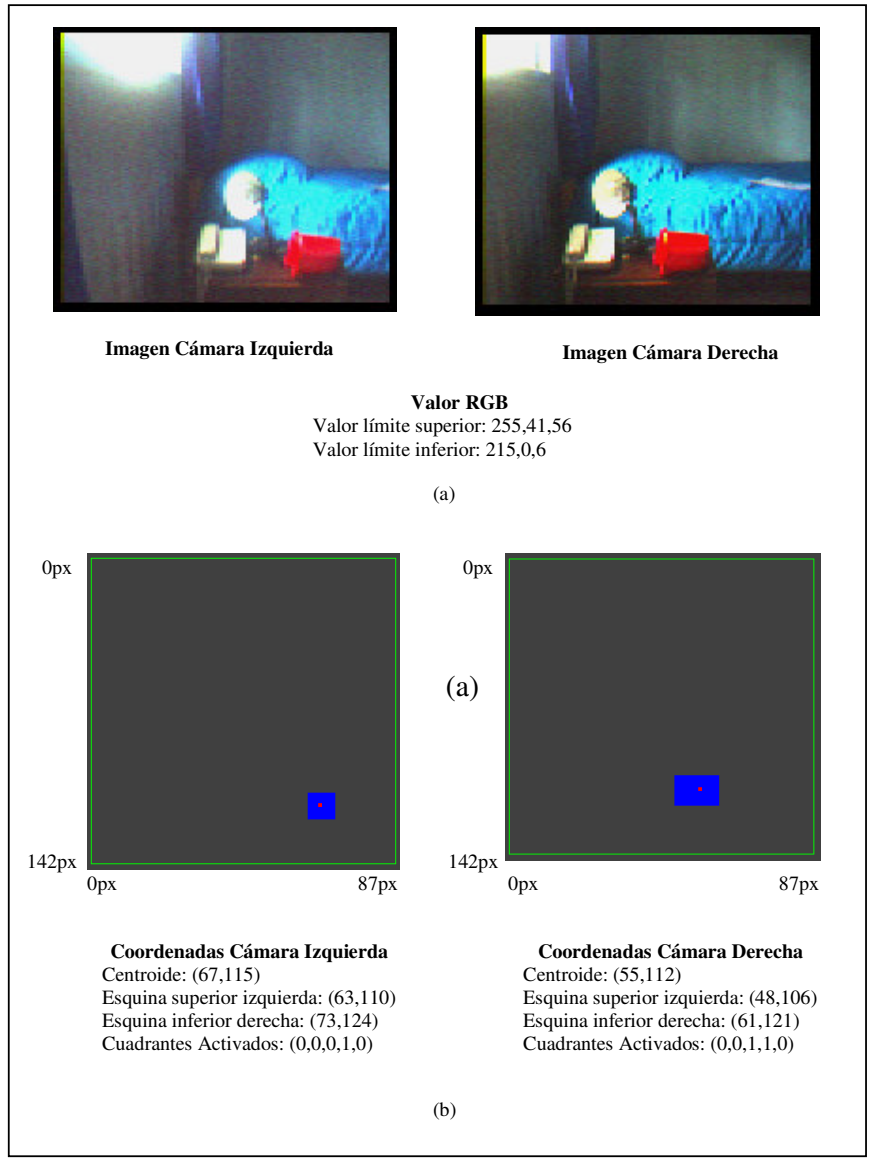


Figura 3.4 Obtención de coordenadas el objeto a través del comando Track Color (a) Imágenes originales. (b) Resultado de aplicar Track Color a las imágenes originales.

3.2 Obtención de la distancia al objeto por Triangulación

Teniendo en cuenta el sistema de geometría de las dos cámaras en estéreo - descrito en la figura 3.2 – es posible calcular la distancia relativa del centro de las cámaras al objetivo. Esta técnica se conoce con el nombre de triangulación ya que para realizar los cálculos se utilizan teoremas correspondientes a la trigonometría. Es importante resaltar que para poder calcular la distancia al objetivo, ambas cámaras deben poder verlo.

3.2.1 Obtención de coordenadas del objeto

Para comenzar con los cálculos que permitirán obtener la distancia relativa al objetivo, es necesario conocer las coordenadas del píxel central del objetivo para cada una de las imágenes captadas por cada cámara. Esto se obtiene a través del comando Track Color del módulo de procesamiento de las cámaras CMUcam2 como ya se ha explicado detalladamente en la sección 3.1.1.

A los efectos de calcular la distancia, se ha dicho que solo importa el desplazamiento horizontal del objeto. Es decir, que el valor de *x* es el que se tiene en cuenta y consiste en la distancia desde el inicio del primer cuadrante o píxel 0 al centro del objetivo. Se denominarán estos valores como *a1* para la cámara 1 o Izquierda y *b1* para la cámara 2 o Derecha. (Figura 3.5)

También se calcula la distancia desde el centro del ángulo de visión al objetivo, denominando a estos segmentos como *a* y *b* para la cámara 1 y 2 respectivamente, de la siguiente manera:

$$a/b = ABS(x - 44\text{px})$$

Los 44px de la ecuación se refieren a la mitad de la resolución en píxeles de la coordenada *x*.

Por último se calcula para la cámara 1 la distancia entre el objetivo y el último píxel del quinto cuadrante (87px) a la que llamaremos *a´* de la siguiente manera:

$$a' = 87\text{px} - x$$

y para la cámara 2 la distancia entre el primer píxel del primer cuadrante (0px) denominada *b´* que es igual a *b1*.

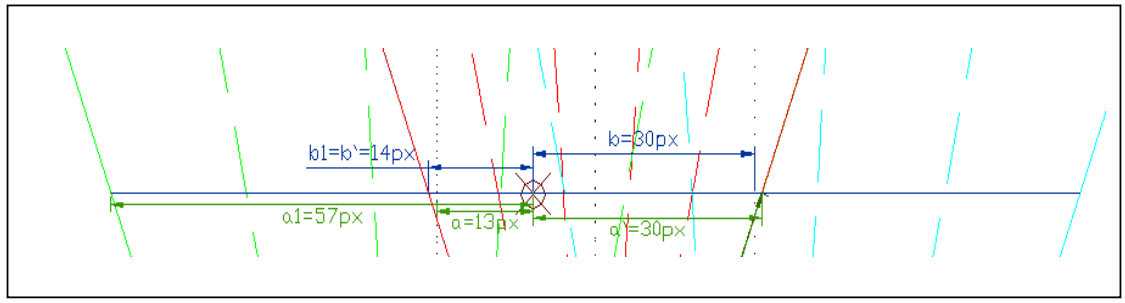


Figura 3.5 Ejemplo de datos relacionados a la ubicación del objetivo.
Para la cámara izquierda el valor de x es 57 y para la derecha es 14.

3.2.2 Cálculo de la distancia relativa al objeto

Partiendo de los datos relacionados a la ubicación del objetivo calculados en la sección anterior, se procede a obtener el segmento *D* correspondiente a la distancia relativa del centro de las cámaras al objeto.

En primer lugar se obtiene la distancia relativa en píxeles desde el centro cada cámara al objetivo. Estos segmentos se denominan *A* y *B* para las cámaras 1 y 2 respectivamente. (Figura 3.6) Para ello se utiliza el teorema del Coseno como sigue:

$$A^2 = A1^2 + a'^2 - 2 * A1 * a' * \cos 70^\circ$$
$$B^2 = B1^2 + b'^2 - 2 * B1 * b' * \cos 70^\circ$$

donde 70º se refiere al ángulo formado entre el plano de la imagen y, para la cámara 1, el segmento que define el final del quinto cuadrante, y para la cámara 2, el segmento que define el inicio del primer cuadrante. *A1* y *B1* hacen referencia a los segmentos correspondientes al final del quinto cuadrante de la cámara 1 y al inicio del primer cuadrante de la cámara 2 hasta la línea que determina la profundidad del objeto. Tanto *A1* como *B1* se calculan como 44px divido el seno de φ*/2 (es decir, sen20º).

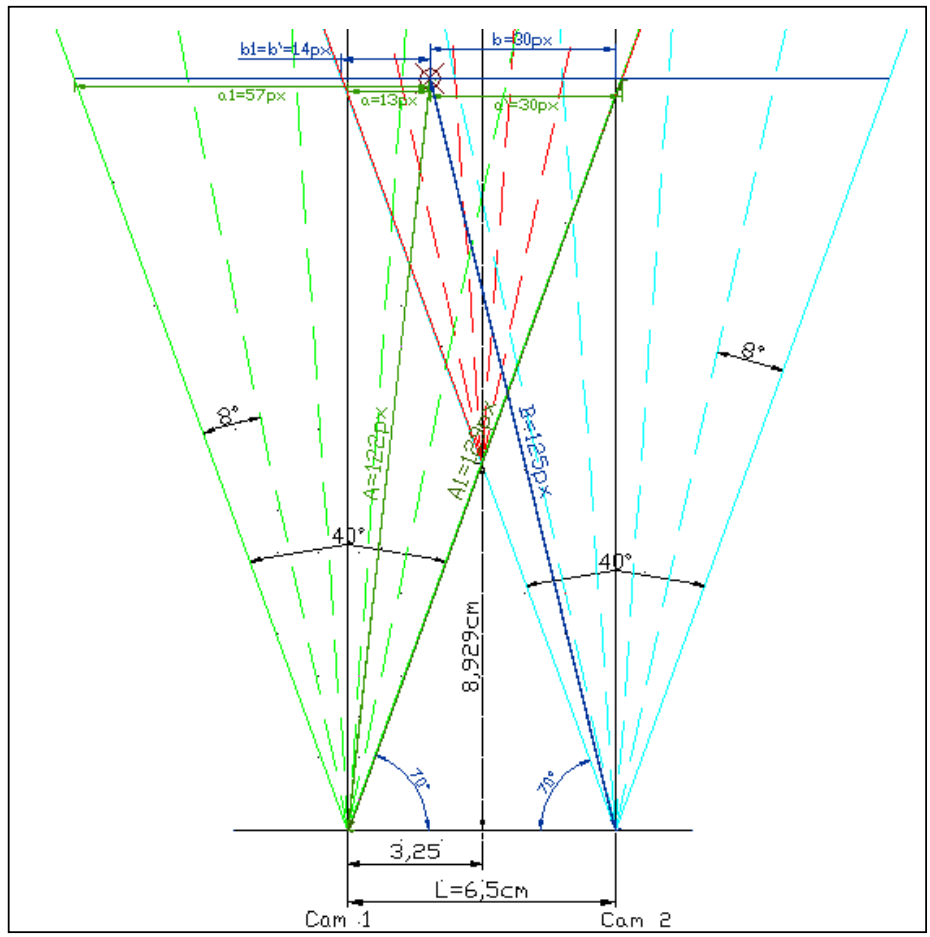


Figura 3.6 Ejemplo de cálculo de la distancia al objetivo (Pasos intermedios)

Luego , para obtener el ángulo comprendido entre el segmento *A1* y *A* denominado γ1 y el ángulo comprendido entre el segmento *B1* y *B* llamado γ2, se utiliza el teorema del Seno:

$$sen(\gamma 1) = \frac{a' * sen(70^\circ)}{A} \qquad sen(\gamma 2) = \frac{b' * sen(70^\circ)}{B}$$

Teniendo como dato estos dos últimos ángulos, se pueden conocer los ángulos φ1 y φ2, que son los comprendidos entre la base de las cámaras y los segmentos *A* y *B* respectivamente.

$$\varphi 1 = 90^\circ - \varphi^*/2 + \gamma 1 \qquad \varphi 2 = 90^\circ - \varphi^*/2 + \gamma 2$$

siendo φ*/2 en este caso igual a 20º.

Con estos dos ángulos se puede obtener por ser suplementarios el ángulo α que es el comprendido entre los dos segmentos *A* y *B*.

$$\alpha = 180^\circ - \varphi 1 - \varphi 2$$

Luego, conociendo α, φ1 y φ2 se puede calcular los segmentos A y B en centímetros, ya que sabemos que la distancia entre ambas cámaras, es decir la línea base L mide 6,5 cm. Esto se obtiene haciendo uso del teorema del Seno:

$$A = \frac{sen(\varphi 2)}{sen(\alpha)} L \qquad\qquad B = \frac{sen(\varphi 1)}{sen(\alpha)} L$$

Por último, la distancia relativa al objetivo en el plano x-y denominada D en centímetros se obtiene con el teorema del Coseno como sigue (Figura 3.7):

$$D^2 = A^2 + \left(\frac{L}{2}\right)^2 - 2 * A * \frac{L}{2} * \cos(\varphi 1)$$

O bien

$$D^2 = B^2 + \left(\frac{L}{2}\right)^2 - 2 * B * \frac{L}{2} * \cos(\varphi 2)$$

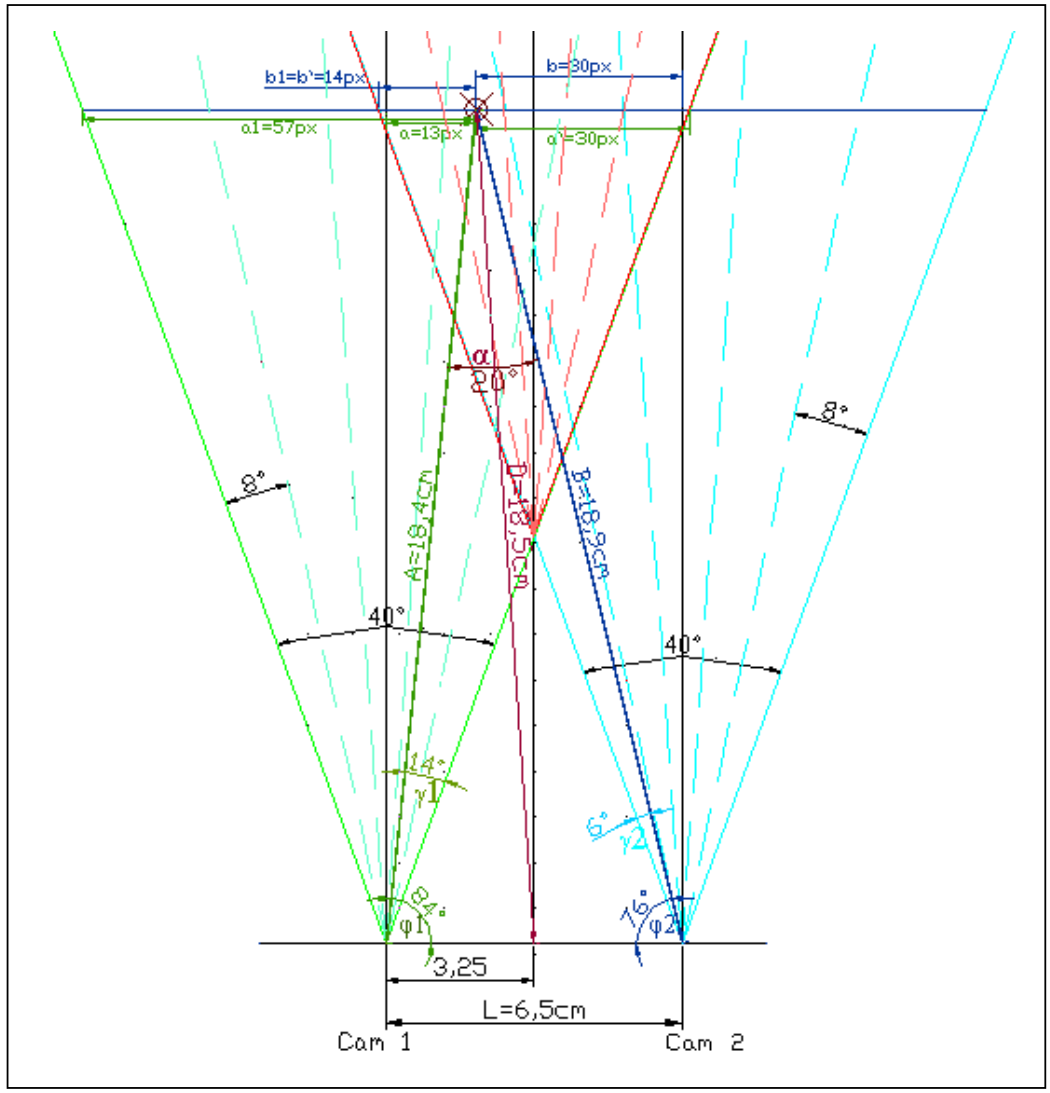


Figura 3.7 Ejemplo de cálculo de la distancia al objetivo.

3.2.3 Cálculo del ángulo de giro

Por último, debe obtenerse el ángulo de giro para que el agente móvil se dirija en forma directa al objetivo. Es decir, el ángulo comprendido entre el segmento que demarca el centro de las cámaras y el segmento D que indica la distancia del centro de las cámaras al objetivo (Figura 3.8).

Para esto primero se calculan los ángulos comprendidos entre la línea base de las cámaras y el segmento D. Se denominan a dichos ángulos como α1 y α2 para las cámaras 1 y 2 respectivamente y se calculan de la siguiente manera a través del teorema del Coseno.

$$\cos(\alpha 1) = \frac{-A^2 + \left(\frac{L}{2}\right)^2 + D^2}{2\frac{L}{2}D} \qquad \cos(\alpha 2) = \frac{-B^2 + \left(\frac{L}{2}\right)^2 + D^2}{2\frac{L}{2}D}$$

Como comprobación se sabe que ambos ángulos son suplementarios, lo que significa que su suma es igual a 180º.

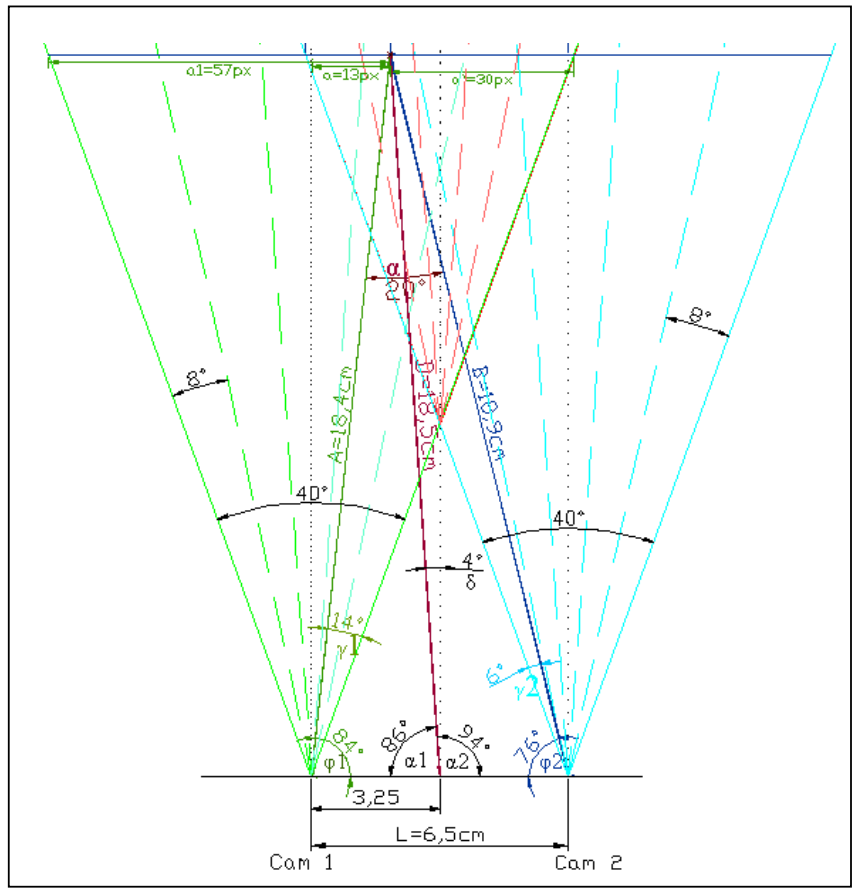


Figura 3.8 Ejemplo de cálculo del ángulo de giro.

La diferencia entre 90º y aquel ángulo α que supere el ángulo recto será la medida del ángulo δ que estamos buscando. Además con esto se puede determinar también la dirección a la cual debe girar el agente. Es decir:

Si $\alpha 1 > 90º$ entonces $\delta = \alpha 1 - 90º$ con giro a la derecha

Si $\alpha 2 > 90º$ entonces $\delta = \alpha 2 - 90º$ con giro a la izquierda

Sino $\delta = 0º$

Capítulo 4

4 Modelo Difuso

La lógica difusa es ampliamente utilizada en el control de la navegación robótica ya que el ambiente donde un robot se moviliza es incierto, dinámico y cambiante. Por lo tanto, es necesario modelar el comportamiento difuso característico de los humanos que permita, mediante la descripción del agente robótico y del ambiente, representar de forma vaga las situaciones a las que se enfrenta el robot. Por ejemplo, saber qué tan lejos se ubica un objeto sin necesidad de conocer precisamente la distancia en metros. Este comportamiento es posible de describir mediante la lógica difusa, garantizando además robustez frente a situaciones de incertidumbre.

Los sistemas de inferencia difusa permiten describir situaciones de forma intuitiva y natural mediante la utilización de etiquetas lingüísticas, haciendo muy simple el diseño del modelo difuso. Además de ésta, se destaca la ventaja de lograr en el agente una movilidad suave e ininterrumpida gracias a que la complejidad computacional de inferencia de la base de reglas es baja. Contribuye a esto el hecho de que no es necesaria una gran cantidad de reglas ya que los resultados de las reglas son interpolados por el sistema de inferencia difusa.

En conclusión, la razón principal para utilizar lógica difusa en el control de navegación robótica es la habilidad de activar parcialmente comportamientos y lograr una transición suave entre los mismos, permitiendo así una movilidad continua y un modelo robusto en circunstancias inciertas [Saffiotti01].

El modelo difuso propuesto (Figura 4.1) se basa en el esquema conocido como "*reglas de contexto*" el cual consiste en la definición de reglas difusas que indican el nivel de activación del *comportamiento* según el valor de verdad del *contexto*. Las mismas tienen la siguiente forma general:

IF *contexto* THEN *comportamiento*

En cuanto a las variables difusas de entrada se utilizarán cinco denominadas Cuadrante1, Cuadrante2, Cuadrante3, Cuadrante4 y Cuadrante5

(C*i*), correspondientes a los cinco cuadrantes que en fueron divididas las imágenes. En cada variable de entrada hay tres etiquetas lingüísticas referidas a los conjuntos difusos de Lejos, Mediano y Cerca, los cuales describen la distancia borrosa entre el objeto y el agente móvil. La combinación de los posibles estados de distancia de los distintos cuadrantes describirán el contexto, formando así el antecedente de las reglas difusas.

Mediante la inferencia se pueden obtener varias conclusiones, que luego de ser agregadas, arrojan una solución en los conjuntos difusos de salida. Estos son representados por dos variables difusas llamadas Izquierdo y Derecho las cuales hacen referencia a los motores (efectores) del agente ubicados a cada lado del mismo. Cada variable tiene cinco etiquetas lingüísticas correspondientes a los conjuntos difusos de Mucho_Adelante, Poco_Adelante, Quieto, Poco_Reversa y Mucho_Reversa referidos a las velocidades y direcciones de los motores. Estas variables indican el comportamiento que tomará el agente definido en el consecuente de las reglas.

Una vez agregadas las salidas y teniendo una única solución, se procede a la defusificación para obtener así un valor nítido que pueda ser aplicado a los efectores.

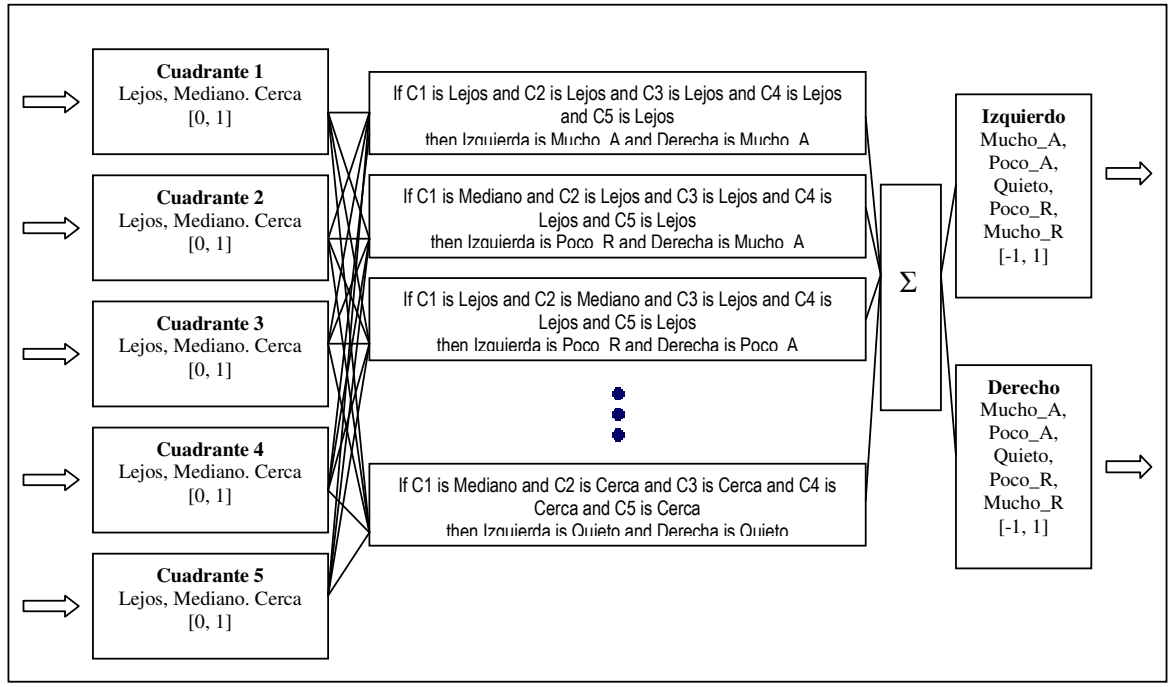


Figura 4.1 Modelo Difuso para la navegación de agentes móviles

4.1 Definición del Modelo Difuso

Uno de los modelos de sistema de inferencia más utilizado en la robótica es el modelo de Mamdani. En este trabajo se aplicará dicho modelo (Figura 4.2).

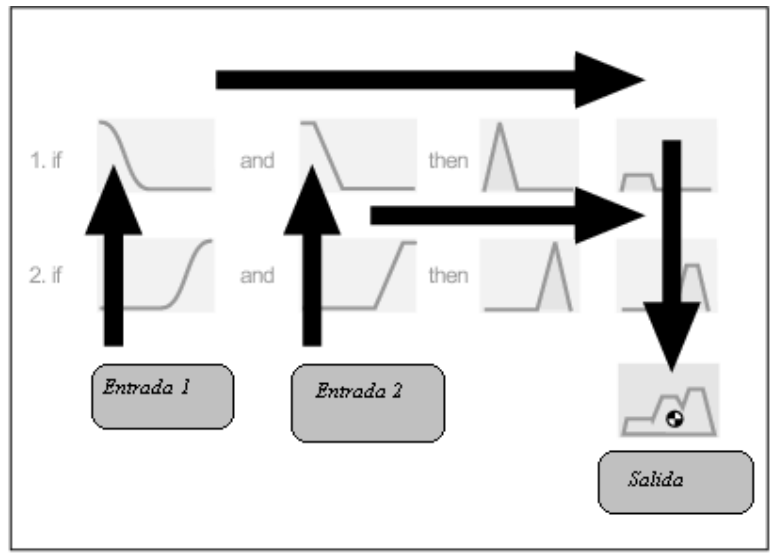


Figura 4.2 Modelo de sistema de inferencia difusa Mamdani [MathWorks01]

4.1.1 Operadores Lógicos

Para el operador lógico AND se utiliza la función Mínimo (min).

$$\mu_{E_A and E_B}(x) = \min\left[\mu_{E_A}(x), \mu_{E_B}(x)\right]$$

Para el operador lógico OR se utiliza la función Máximo (max).

$$\mu_{E_A or E_B}(x) = \max\left[\mu_{E_A}(x), \mu_{E_B}(x)\right]$$

4.1.2 Método de Implicación

El método de implicación es el propuesto por Mamdani, es decir, el Mínimo.

$$\mu_{E \to S}(x, y) = \min(\mu_E(x), \mu_S(y))$$

4.1.3 Método de Agregación

El método de agregación utilizado es el de la Unión de los conjuntos difusos de salida obtenidos.

$$\mu_y(w) = \max(\mu_y^1(w), \mu_y^2(w))$$

4.1.4 Método de Defusificación

El método de defusificación que se utiliza es el del Centroide, es decir, aquel que encuentra el centro de gravedad de la superficie difusa de salida.

La siguiente ecuación es una aproximación a dicho método que permite calcularlo más rápidamente.

$$S' = \frac{\sum_{i=1}^{n} y_i \mu_S(y)}{\sum_{i=1}^{n} \mu_S(y)}$$

4.1.5 Entradas Difusas

El sistema de inferencia difuso cuenta con cinco variables de entradas difusas.

$$E = (C1, C2, C3, C4, C5)$$

Cada entrada *Ci* hace referencia al cuadrante *i* de los cinco en los que está dividida la imagen.

El universo de discurso de las entradas es la distancia entre el agente móvil y el objeto, obtenida de la disparidad encontrada mediante la primer técnica explicada en el capítulo anterior. Sus valores posibles se encuentran en el intervalo [0, 1], donde los valores cercanos a cero se refieren a objetos lejanos y los cercanos a 1 a los cercanos.

$$U_E = [0, 1]$$

Las etiquetas lingüísticas para las entradas difusas están definidas como LEJOS para los objetos que estén ubicados lejos del agente móvil, MEDIANO para los cuales su distancia sea intermedia y CERCA para los que están cercanos al agente.

$$T(E) = \{LEJOS, MEDIANO, CERCA\}$$

Los conjuntos difuso de entrada quedan definidos para todas las variables de entrada de la siguiente manera (Figura 4.3):

$$LEJOS = [0, 0, 0.1, 0.25]$$
$$MEDIANO = [0.15, 0.35, 0.45, 0.65]$$
$$CERCA = [0.55, 0.7, 1, 1]$$

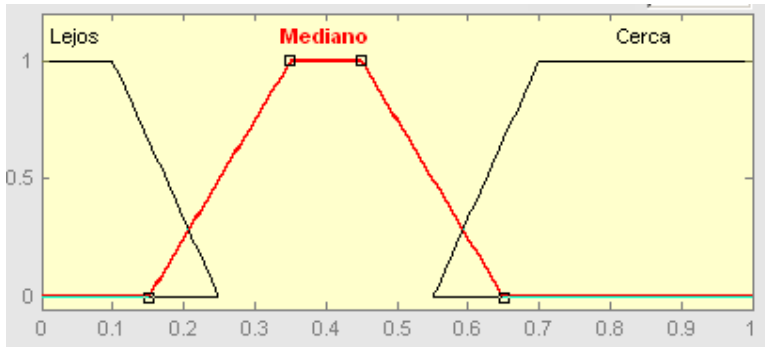


Figura 4.3 Conjuntos difusos de entrada para las variables Cuadrante 1 a Cuadrante 5

4.1.6 Salidas Difusas

El sistema de inferencia difuso cuenta con dos variables de salida difusas.

$$S = (IZQUIERDO, DERECHO)$$

Estas variables hacen referencia a los efectores del agente móvil. La variable IZQUIERDO se refiere al motor izquierdo y la variable DERECHO al motor derecho del agente.

El universo de discurso para las variables de salida son los valores defusificados que representan tensiones eléctricas para el movimiento de los efectores. Los valores posibles están en el intervalo de [-1, 1], donde un valor cercano a –1 significa un movimiento veloz en dirección reversa, un valor de 0 detiene los efectores y los valores cercanos a 1 producen un movimiento veloz hacia delante.

$$U_S = [-1, 1]$$

Las etiquetas lingüísticas para las salidas difusas están definidas como MUCHO_R que se refiere a un movimiento veloz hacia atrás, POCO_R referido a

un movimiento lento hacia atrás, QUIETO cuando el efector se detiene, POCO_A referido a un movimiento lento hacia adelante y MUCHO_A cuando el efector se mueve velozmente hacia adelante.

$$T(S) = \{MUCHO_R, POCO_R, QUIETO, POCO_A, MUCHO_A\}$$

Los conjuntos difusos de salida quedan definidos para ambas variables de las siguiente manera (Figura 4.4):

$$MUCHO_R = [-1, -1, -0.6]$$
$$POCO_R = [-0.8, -0.4, 0]$$
$$QUIETO = [-0.3, 0, 0.3]$$
$$POCO_A = [0, 0.4, 0.8]$$
$$MUCHO_A = [0.6, 1, 1]$$

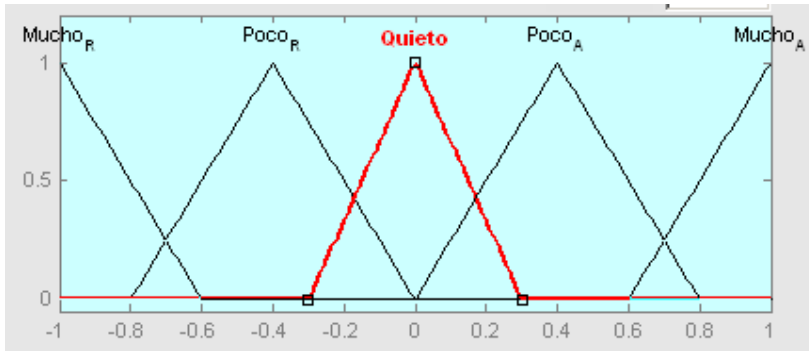


Figura 4.4 Conjuntos difusos de salida para las variables Izquierdo y Derecho

4.1.7 Función Membresía

Las funciones membresía para las variables de entrada son de tipo trapezoidales y quedan definidas de las siguiente manera:

Conjunto difuso *LEJOS*:

$$f(LEJOS, 0, 0, 0.1, 0.25) = \begin{cases} 0 & para\ LEJOS < 0 \\ \dfrac{LEJOS - 0}{0 - 0} & para\ 0 <= LEJOS < 0 \\ 1 & para\ 0 <= LEJOS < 0.1 \\ \dfrac{0.25 - LEJOS}{0.25 - 0.1} & para\ 0.1 <= LEJOS < 0.25 \\ 0 & para\ 0.25 <= LEJOS \end{cases}$$

Conjunto difuso *MEDIANO*:

$$f(MEDIANO, 0.15, 0.35, 0.45, 0.65) = \begin{cases} 0 & para\ MEDIANO < 0.15 \\ \dfrac{MEDIANO - 0.15}{0.35 - 0.15} & para\ 0.15 <= \text{MEDIANO} < 0.35 \\ 1 & para\ 0.35 <= MEDIANO < 0.45 \\ \dfrac{0.65 - MEDIANO}{0.65 - 0.45} & para\ 0.45 <= MEDIANO < 0.65 \\ 0 & para\ 0.65 <= MEDIANO \end{cases}$$

Conjunto difuso *CERCA*:

$$f(CERCA, 0.55, 0.7, 1, 1) = \begin{cases} 0 & para\ CERCA < 0.55 \\ \dfrac{CERCA - 0.55}{0.7 - 0.55} & para\ 0.55 <= CERCA < 0.7 \\ 1 & para\ 0.7 <= CERCA < 1 \\ \dfrac{1 - CERCA}{1 - 1} & para\ 1 <= CERCA < 1 \\ 0 & para\ 1 <= CERCA \end{cases}$$

4.1.8 Reglas Difusas

Las reglas de inferencia difusa se encargan de guiar el razonamiento del Sistema de Inferencia Difuso. Con pocas reglas se puede obtener una función suavizada sin cambios bruscos ya que un valor cualquiera *x* puede pertenecer a varios conjuntos difusos al mismo tiempo lo que permite tener una transición suave. En otros sistemas de inferencia como la lógica de primer orden, para obtener una función suavizada se debe definir una gran cantidad de reglas.

Se definen 39 reglas que permitirán al agente móvil acercarse al objetivo en el escenario.

La base de reglas es la siguiente:

R1 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Mucho_A) (Derecha is Mucho_A)
R2 If (C1 is Mediano) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Poco_R) (Derecha is Mucho_A)
R3 If (C1 is Lejos) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Poco_R) (Derecha is Poco_A)

R4 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Mediano) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Poco_A) (Derecha is Poco_A)
R5 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Mediano) and (C5 is Lejos)
then (Izquierda is Poco_A) (Derecha is Poco_R)
R6 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Mediano)
then (Izquierda is Mucho_A) (Derecha is Poco_R)
R7 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Poco_R) (Derecha is Poco_A)
R8 If (C1 is Lejos) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Poco_A)
R9 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Lejos)
then (Izquierda is Poco_A) (Derecha is Quieto)
R10 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Mediano) and (C5 is Mediano)
then (Izquierda is Poco_A) (Derecha is Poco_R)
R11 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Poco_A)
R12 If (C1 is Lejos) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Lejos)
then (Izquierda is Poco_A) (Derecha is Poco_A)
R13 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Mediano)
then (Izquierda is Poco_A) (Derecha is Quieto)
R14 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Lejos)
then (Izquierda is Poco_A) (Derecha is Poco_A)
R15 If (C1 is Lejos) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Mediano)
then (Izquierda is Poco_A) (Derecha is Poco_A)
R16 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Mediano)
then (Izquierda is Poco_A) (Derecha is Poco_A)
R17 If (C1 is Cerca) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R18 If (C1 is Lejos) and (C2 is Cerca) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R19 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Cerca) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R20 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Cerca) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R21 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R22 If (C1 is Cerca) and (C2 is Cerca) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R23 If (C1 is Lejos) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R24 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R25 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Cerca) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R26 If (C1 is Cerca) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R27 If (C1 is Lejos) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R28 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R29 If (C1 is Cerca) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Lejos)
then (Izquierda is Quieto) (Derecha is Quieto)
R30 If (C1 is Lejos) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R31 If (C1 is Cerca) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R32 If (C1 is Cerca) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Mediano)
then (Izquierda is Quieto) (Derecha is Quieto)
R33 If (C1 is Cerca) and (C2 is Cerca) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Mediano)
then (Izquierda is Quieto) (Derecha is Quieto)
R34 If (C1 is Cerca) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Mediano) and (C5 is Mediano)
then (Izquierda is Quieto) (Derecha is Quieto)
R35 If (C1 is Cerca) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Mediano)

then (Izquierda is Quieto) (Derecha is Quieto)
R36 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Mediano) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R37 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Mediano) and (C4 is Cerca) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R38 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)
R39 If (C1 is Mediano) and (C2 is Cerca) and (C3 is Cerca) and (C4 is Cerca) and (C5 is Cerca)
then (Izquierda is Quieto) (Derecha is Quieto)

4.2 Proceso de Inferencia Difusa (PID)

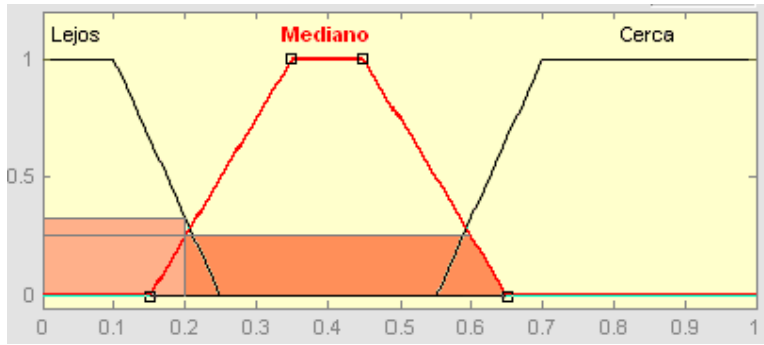
Se presenta a continuación un ejemplo completo del proceso de inferencia difusa donde se mostrarán todos los pasos a seguir para completar el razonamiento.

4.2.1 Entradas Difusas

Se tomará como ejemplo el siguiente el conjunto de variables difusas de entrada E = (0.2; 0.2; 0; 0; 0).

En primer lugar se considera el valor de entrada C1 (Cuadrante 1) para determinar a qué conjunto difuso corresponde y qué valor de membresía tiene en él. Su valor es 0.2 y se encuentra tanto en el intervalo [0, 0.25] que corresponde al conjunto difuso LEJOS, como en el intervalo [0.15, 0.65] que corresponde al conjunto difuso MEDIANO. Por lo tanto pertenece a ambos conjuntos con un grado de membresía determinado.

C1 = 0.2



Utilizando la función membresía del conjunto difuso LEJOS se determina el valor de pertenencia a este conjunto difuso $\mu_{C1}(0.2)$.

$$f(0.2, 0, 0, 0.1, 0.25) = \begin{cases} 0 & para\ 0.2 < 0 \\ \dfrac{0.2 - 0}{0 - 0} & para\ 0 <= 0.2 < 0 \\ 1 & para\ 0 <= 0.2 < 0.1 \\ \dfrac{0.25 - 0.2}{0.25 - 0.1} & para\ 0.1 <= 0.2 < 0.25 \\ 0 & para\ 0.25 <= 0.2 \end{cases}$$

En la función membresía se reemplaza LEJOS por el valor de C1, en este caso 0.2, y se obtiene el valor *f (0.2;LEJOS) = 0.333* porque C1 pertenece al intervalo 0.1 <= 0.2 < 0.25.

Por lo tanto, el valor de pertenencia del conjunto difuso LEJOS para el valor del Cuadrante 1 es $\mu_{C1}(0.2) = 0.333$.

Luego, utilizando la función membresía del conjunto difuso MEDIANO se determina el valor de pertenencia a este conjunto difuso $\mu_{C1}(0.2)$.

$$f(0.2, 0.15, 0.35, 0.45, 0.65) = \begin{cases} 0 & para\ 0.2 < 0.15 \\ \dfrac{0.2 - 0.15}{0.35 - 0.15} & para\ 0.15 <= 0.2 < 0.35 \\ 1 & para\ 0.35 <= 0.2 < 0.45 \\ \dfrac{0.65 - 0.2}{0.65 - 0.45} & para\ 0.45 <= 0.2 < 0.65 \\ 0 & para\ 0.65 <= 0.2 \end{cases}$$

En la función membresía se reemplaza MEDIANO por el valor de C1, en este caso 0.2, y se obtiene el valor *f (0.2;MEDIANO) = 0.25* porque C1 pertenece al intervalo 0.15 <= 0.2 < 0.35.

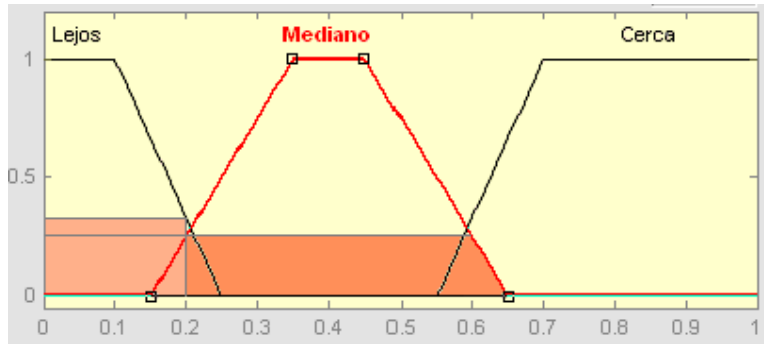
Por lo tanto, el valor de pertenencia del conjunto difuso MEDIANO para el valor del Cuadrante 1 es $\mu_{C1}(0.2) = 0.25$.

Luego se toma el valor de entrada del C2 (Cuadrante 2) para determinar a qué conjunto difuso pertenece y qué grado de membresía tiene en él. En este caso, como el valor C2 es igual a C1 y ambas variables están definidas de la misma manera al igual que sus funciones membresía, se obtienen los mismos resultados que para C1, es decir,

$$f(0.2;LEJOS) = 0.333$$

$$f(0.2;MEDIANO) = 0.25$$

$$C2 = 0.2$$

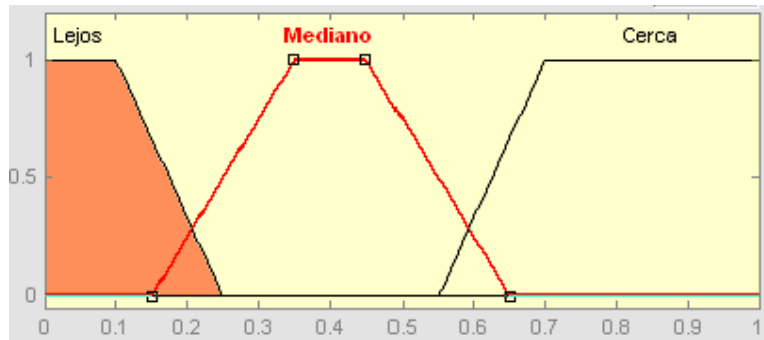


A continuación, se considera el valor de C3 (Cuadrante 3) y se determina a qué conjunto difuso pertenece y su nivel de membresía. Como el valor de C3 es 0, pertenece solo al intervalo [0, 0.25], por lo tanto corresponde a conjunto difuso LEJOS. Su grado de pertenencia se obtiene reemplazando en la función membresía LEJOS por el valor de C3, en este caso 0, y se obtiene el valor *f (0;LEJOS) = 1* porque C3 pertenece al intervalo 0 <= 0 < 0.1.

$$f(0, 0, 0, 0.1, 0.25) = \begin{cases} 0 & \text{para } 0 < 0 \\ \dfrac{0-0}{0-0} & \text{para } 0 <= 0 < 0 \\ 1 & \text{para } 0 <= 0 < 0.1 \\ \dfrac{0.25-0}{0.25-0.1} & \text{para } 0.1 <= 0 < 0.25 \\ 0 & \text{para } 0.25 <= 0. \end{cases}$$

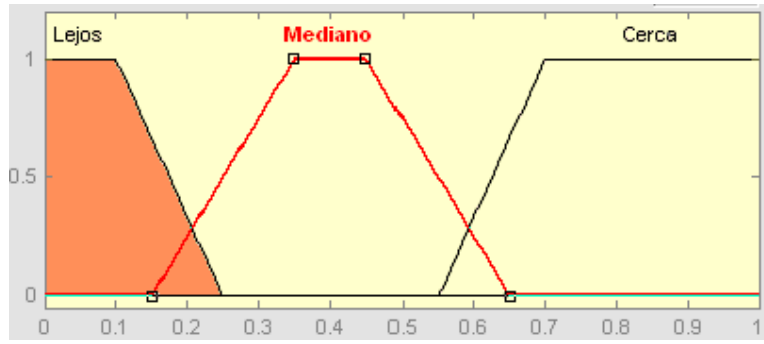
Por lo tanto, el valor de pertenencia del conjunto difuso LEJOS para el valor del Cuadrante 3 es $\mu_{C3}(0) = 1$.

C3 = 0

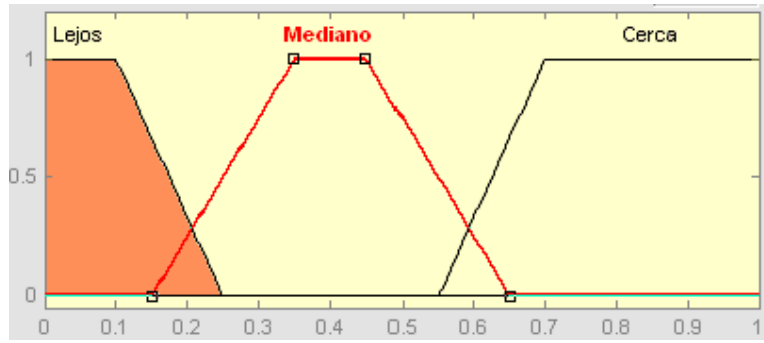


Al igual que el valor de C3, se toman los valores de C4 (Cuadrante 4) y C5 (Cuadrante 5), que como son iguales a C3 y las variables y sus funciones membresía están definidas de la misma manera, se sabe que todas pertenecen al conjunto difuso lejos y su grado de pertenencia es 1, es decir $\mu_{C4}(0) = 1$ y $\mu_{C5}(0) = 1$.

C4 = 0



C5 = 0



4.2.2 Operadores Difusos

Las entradas difusas fusificadas en el paso anterior dieron pertenencia a los siguientes conjuntos difusos para cada variable:

T_E = { (C1=LEJOS, C2=LEJOS, C3=LEJOS, C4=LEJOS, C5=LEJOS);
(C1=MEDIANO, C2=LEJOS, C3=LEJOS, C4=LEJOS, C5=LEJOS);
(C1=LEJOS, C2=MEDIANO, C3=LEJOS, C4=LEJOS, C5=LEJOS);
(C1=MEDIANO, C2=MEDIANO, C3=LEJOS, C4=LEJOS, C5=LEJOS)
}

Por lo tanto, con esta combinación de entradas se activarán las reglas 1, 2, 3 y 7, ya que sus antecedentes se corresponden.

R1 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Mucho_A) (Derecha is Mucho_A)

R2 If (C1 is Mediano) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Poco_R) (Derecha is Mucho_A)

R3 If (C1 is Lejos) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Poco_R) (Derecha is Poco_A)

R7 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)
then (Izquierda is Poco_R) (Derecha is Poco_A)

El paso siguiente es utilizar los operadores difusos para calcular el nivel de activación de los antecedentes:

Antecedente R1:

(C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)

El antecedente se encuentra relacionado por conjunciones por lo cual se utiliza el operador difuso min() para calcular su valor:

$$\mu_{E_A and E_B}(x) = \min\left[\mu_{E_A}(x), \mu_{E_B}(x)\right]$$

Los valores de las entradas y el grado de pertenencia para cada variable difusa son:

$\mu_{C1}(0.2) = 0.333$, $\mu_{C2}(0.2) = 0.333$, $\mu_{C3}(0) = 1$, $\mu_{C4}(0) = 1$, y $\mu_{C5}(0) = 1$

Calculando el mínimo se obtiene el grado de cumplimiento del antecedente de la regla 1:

Min[0.333,0.333,1,1,1]

Antecedente R1: 0.333

De la misma manera se calculan los antecedentes de las reglas 2, 3 y 7.

Antecedente R2:

(C1 is Mediano) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)

$\mu_{C1}(0.2) = 0.25$, $\mu_{C2}(0.2) = 0.333$, $\mu_{C3}(0) = 1$, $\mu_{C4}(0) = 1$, y $\mu_{C5}(0) = 1$

Antecedente R2: 0.25

Antecedente R3:

(C1 is Lejos) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)

$\mu_{C1}(0.2) = 0.333$, $\mu_{C2}(0.2) = 0.25$, $\mu_{C3}(0) = 1$, $\mu_{C4}(0) = 1$, y $\mu_{C5}(0) = 1$

Antecedente R3: 0.25

Antecedente R7:

(C1 is Mediano) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos)

$\mu_{C1}(0.2) = 0.25$, $\mu_{C2}(0.2) = 0.25$, $\mu_{C3}(0) = 1$, $\mu_{C4}(0) = 1$, y $\mu_{C5}(0) = 1$

Antecedente R7: 0.25

4.2.3 Inferencia Difusa

Se utiliza el método de Mamdani como método de implicación:

$$\mu_{E\to S}(x, y) = \min(\mu_E(x), \mu_S(y))$$

Recordamos las reglas activadas:

R1 If (C1 is Lejos) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos) then (Izquierda is Mucho_A) (Derecha is Mucho_A)

R2 If (C1 is Mediano) and (C2 is Lejos) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos) then (Izquierda is Poco_R) (Derecha is Mucho_A)

R3 If (C1 is Lejos) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos) then (Izquierda is Poco_R) (Derecha is Poco_A)

R7 If (C1 is Mediano) and (C2 is Mediano) and (C3 is Lejos) and (C4 is Lejos) and (C5 is Lejos) then (Izquierda is Poco_R) (Derecha is Poco_A)

Regla 1:

Implicación para ENTRADA $\longrightarrow$ IZQUIERDO

El valor de pertenencia de la salida difusa Izquierda es de 1

$$If\ 0.333\ then\ 1$$

$$\min(0.333, 1)$$

$$\mu_{E\to S}(IZQUIERDO) = 0.333$$

Implicación para ENTRADA $\longrightarrow$ DERECHO

El valor de pertenencia de la salida difusa Derecha es de 1

$$If\ 0.333\ then\ 1$$

$$\min(0.333, 1)$$

$$\mu_{E\to S}(DERECHO) = 0.333$$

Regla 2:

Implicación para ENTRADA $\longrightarrow$ IZQUIERDO

El valor de pertenencia de la salida difusa Izquierda es de 1

$$If\ 0.25\ then\ 1$$

$$\min(0.25, 1)$$

$$\mu_{E\to S}(IZQUIERDO) = 0.25$$

Implicación para ENTRADA $\longrightarrow$ DERECHO

El valor de pertenencia de la salida difusa Derecha es de 1

$$If\ 0.25\ then\ 1$$

$$\min(0.25, 1)$$

$$\mu_{E\to S}(DERECHO) = 0.25$$

Regla 3:

Implicación para ENTRADA $\longrightarrow$ IZQUIERDO

El valor de pertenencia de la salida difusa Izquierda es de 1

$$If\ 0.25\ then\ 1$$

$$\min(0.25, 1)$$

$$\mu_{E\to S}(DERECHO) = 0.25$$

Implicación para ENTRADA $\longrightarrow$ DERECHO

El valor de pertenencia de la salida difusa Derecha es de 1

$$If\ 0.25\ then\ 1$$

$$\min(0.25, 1)$$

$$\mu_{E\to S}(DERECHO) = 0.25$$

Regla 7:

Implicación para ENTRADA $\longrightarrow$ IZQUIERDO

El valor de pertenencia de la salida difusa Izquierda es de 1

$$If\ 0.25\ then\ 1$$

$$\min(0.25, 1)$$

$$\mu_{E\to S}(DERECHO) = 0.25$$

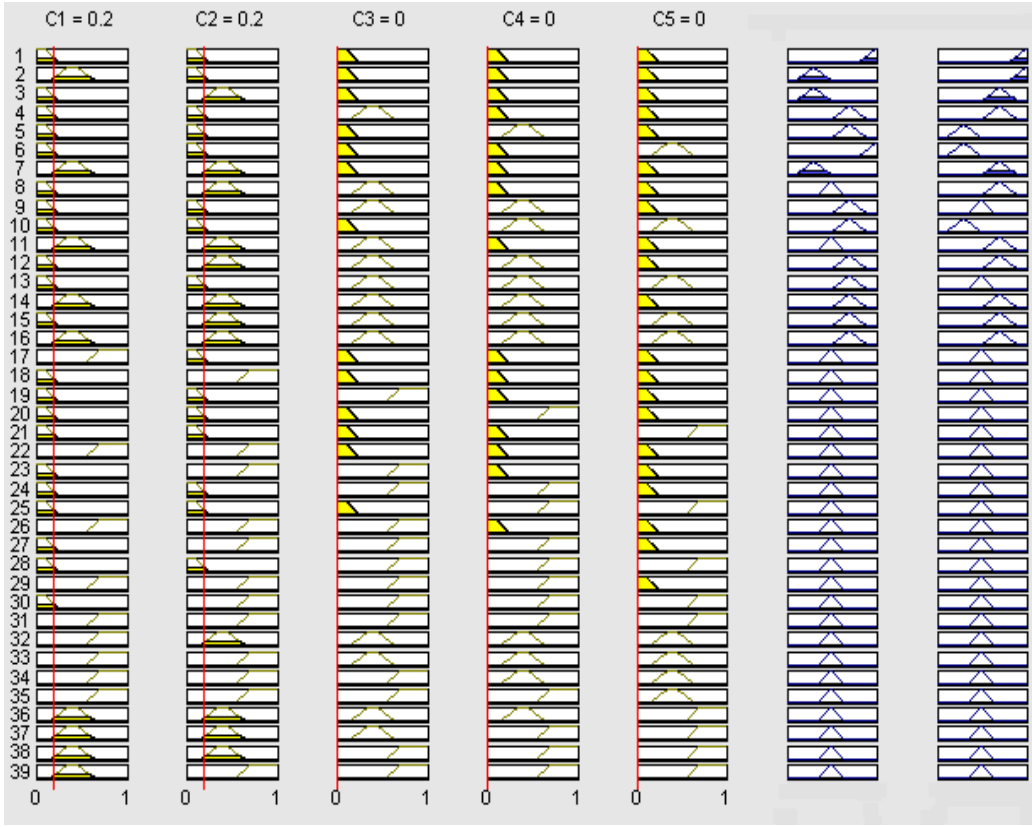
Implicación para ENTRADA $\longrightarrow$ DERECHO

El valor de pertenencia de la salida difusa Derecha es de 1

$$If\ 0.25\ then\ 1$$

$$\min(0.25, 1)$$

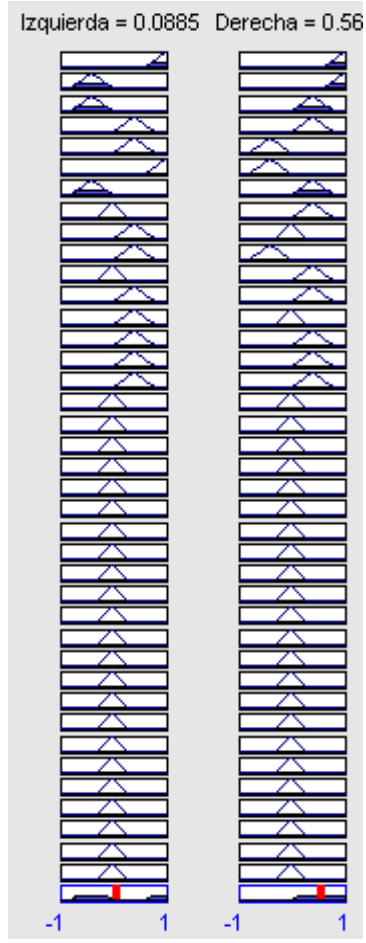
$$\mu_{E\to S}(DERECHO) = 0.25$$


C1 = 0.2
C2 = 0.2
C3 = 0
C4 = 0
C5 = 0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
0
1
0
1
0
1
0
1
0
1

4.2.4 Agregación

Como en este ejemplo se dispararon varias reglas de inferencia, deben agregarse los conjuntos difusos de salida. El método de agregación utilizado es el de la Unión de los conjuntos difusos de salida obtenidos.

$$\mu_y(w) = \max(\mu_y^1(w), \mu_y^2(w))$$



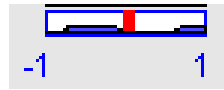
4.2.5 Defusificación

Para determinar el valor nítido de los conjuntos de salida IZQUIERDO y DERECHO se utilizará la aproximación del cálculo del centro de gravedad.

$$S' = \frac{\sum_{i=1}^{n} y_i \mu_S(y)}{\sum_{i=1}^{n} \mu_S(y)}$$

Valor nítido para la salida difusa IZQUIERDO

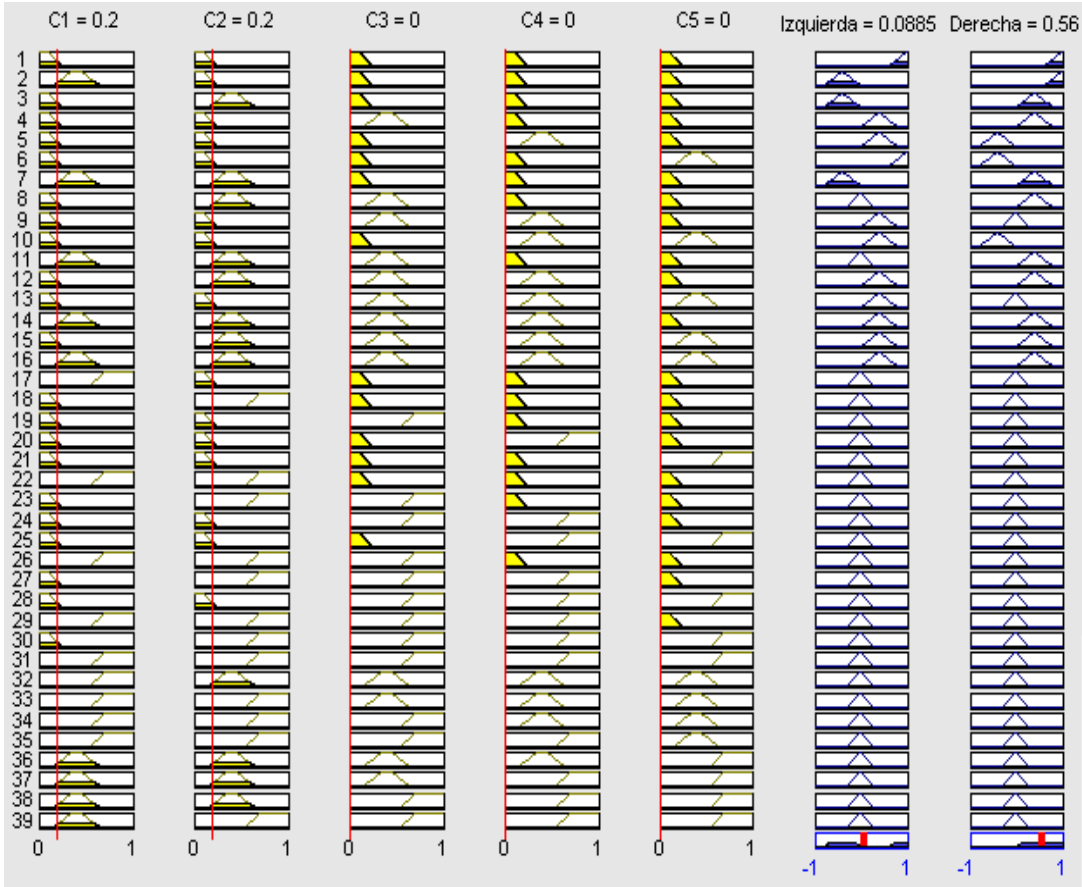
IZQUIERDO = 0.0885



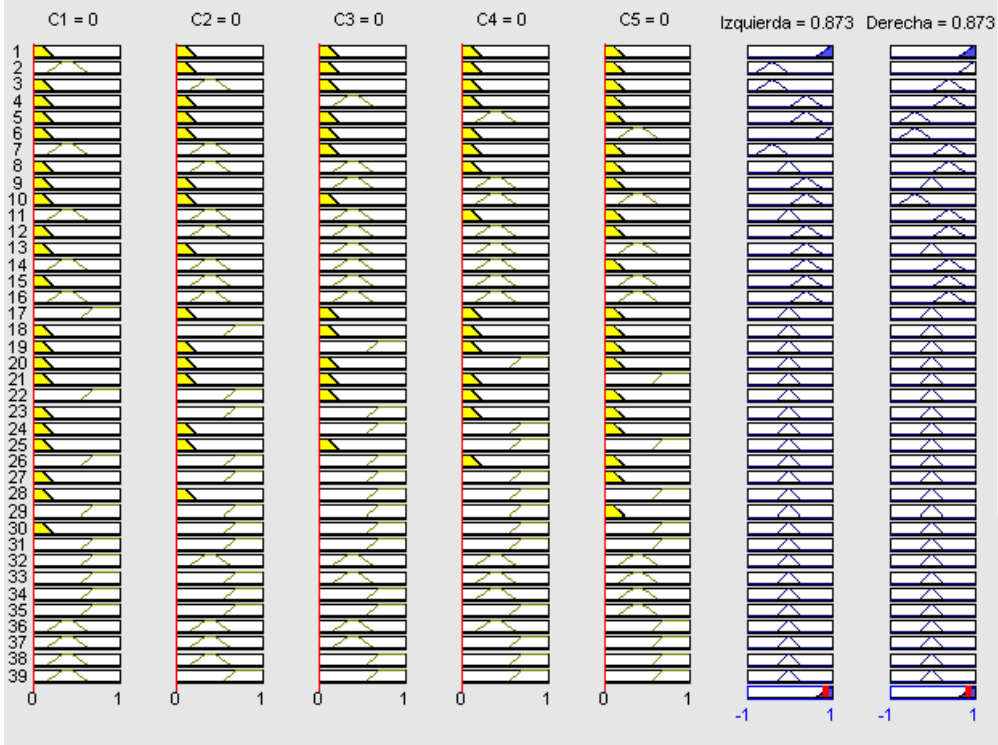
Valor nítido para la salida difusa DERECHO

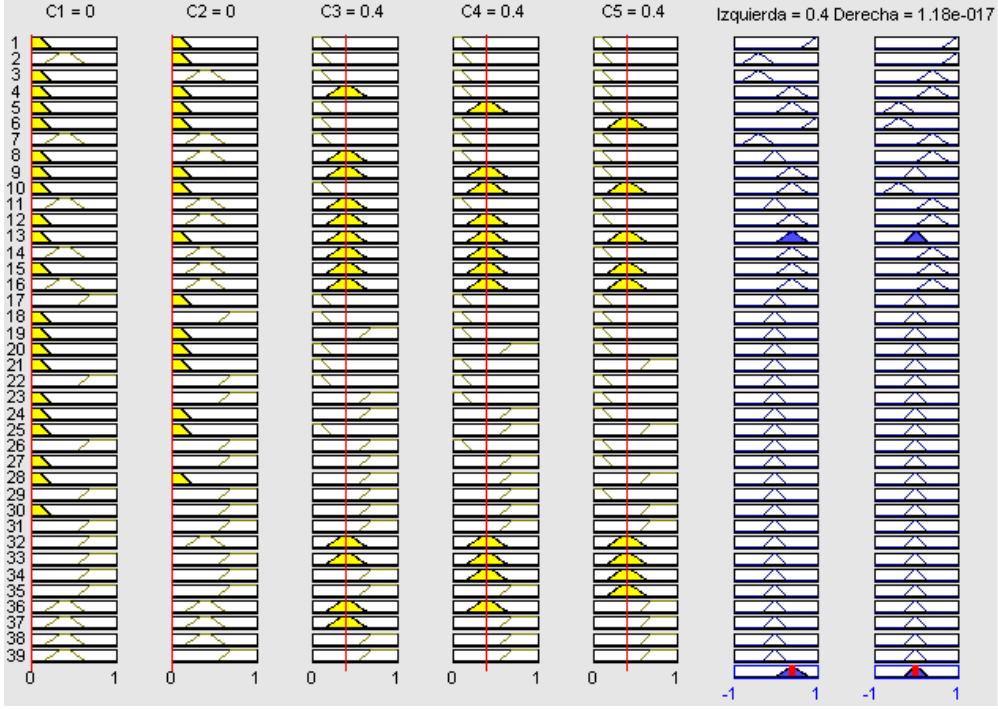
DERECHO = 0.56

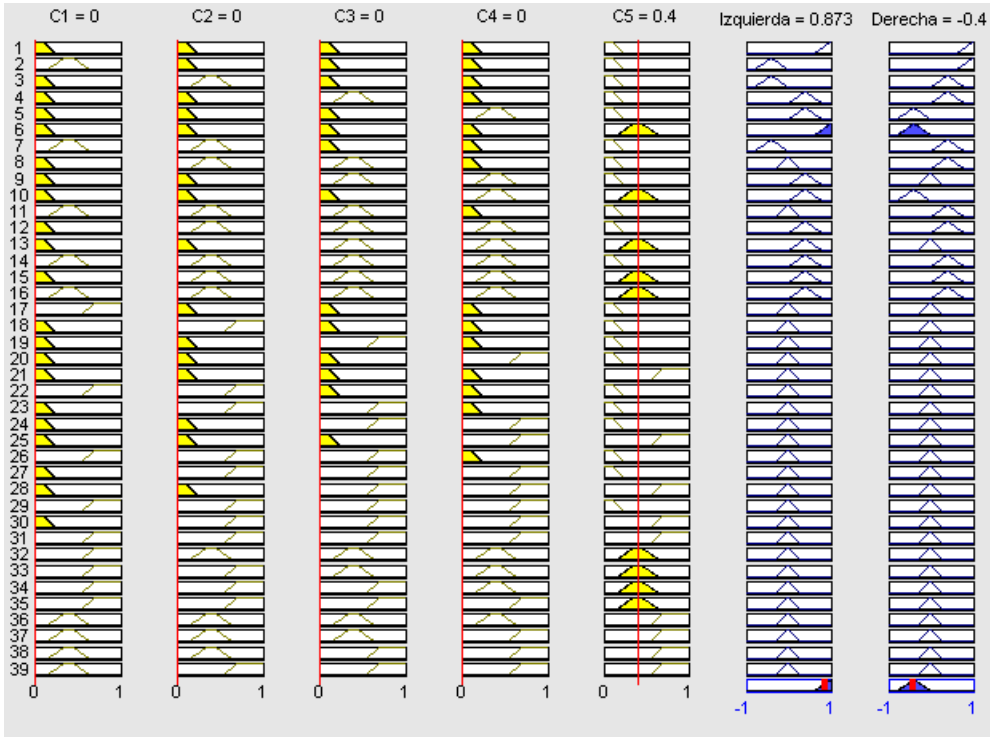
4.2.6 Proceso de Inferencia Difusa

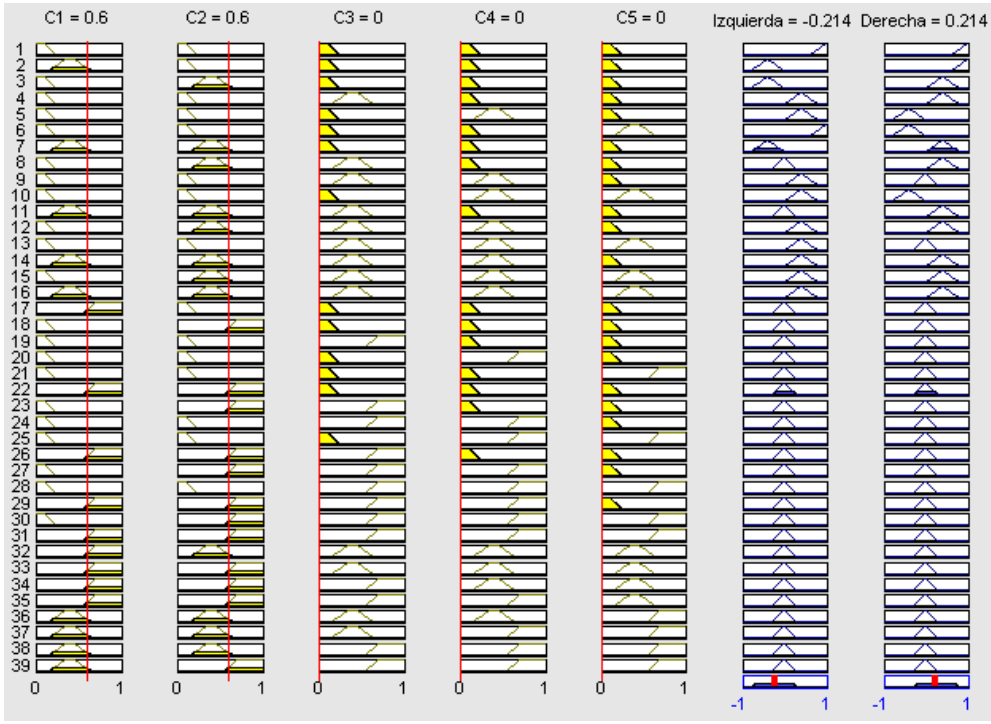


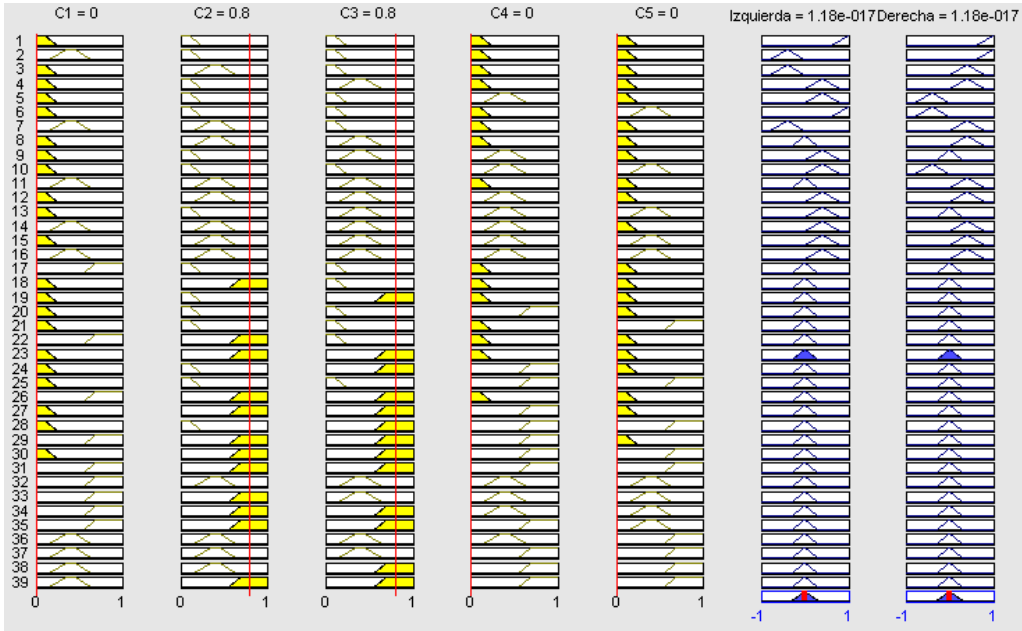
4.3 Ejemplos del PID

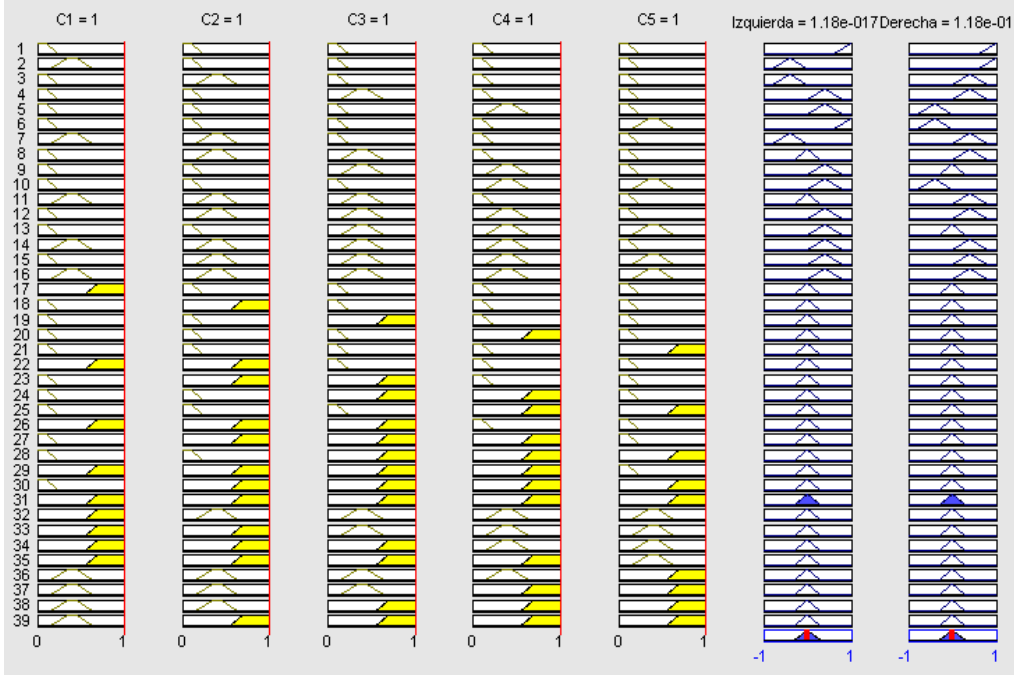





C1 = 0
C2 = 0
C3 = 0
C4 = 0
C5 = 0.4
Izquierda = 0.873
Derecha = -0.4


C1 = 0.6
C2 = 0.6
C3 = 0
C4 = 0
C5 = 0
Izquierda = -0.214
Derecha = 0.214


C1 = 0
C2 = 0.8
C3 = 0.8
C4 = 0
C5 = 0
Izquierda = 1.18e-017
Derecha = 1.18e-017


C1 = 1
C2 = 1
C3 = 1
C4 = 1
C5 = 1
Izquierda = 1.18e-017
Derecha = 1.18e-017

Capítulo 5

5 Integración de Técnicas de Procesamiento y Modelo Difuso

La integración de los resultados obtenidos por las diferentes técnicas de procesamiento de imágenes descritas en el capítulo 3 con el modelo difuso detallado en el capítulo 4, se realiza a través de distintos pasos, los cuales buscan hacer la navegación suave, continua y precisa.

A partir de la secuencia de pasos que se describirán en este capítulo, se logra una circulación suave hasta llegar cerca del objetivo y luego se ajustan los movimientos del agente para que logre llegar a la meta de manera más precisa.

5.1 Integración

En primer lugar se toma en cuenta el resultado de la activación de los cuadrantes de cada cámara y la disparidad obtenida de acuerdo a lo detallado en la sección 3.1 del capítulo 3 y a partir de una función que propaga dicha disparidad se activan los cuadrantes correspondientes a las variables de entrada del modelo difuso.

Luego, según el proceso de inferencia difusa, se procede a mover los motores en la dirección e intensidad correspondiente. Si, en cambio, el resultado del PID sugiere que los motores deben permanecer quietos, ya sea porque algún cuadrante del modelo difuso ha tomado el valor de "cerca", o por la agregación y defusificación de las reglas resultantes, se calcula la distancia relativa al objetivo a partir de la técnica de triangulación detallada en la sección 3.2 del capítulo 3.

El agente móvil se moverá en la dirección correspondiente (derecha o izquierda) hasta que el ángulo de giro sea igual a 0º lo que indicará que el robot se encuentra ubicado directamente en frente al objetivo a una cierta distancia expresada en centímetros. Por último se dirigirá hacia delante para encontrarse con el objetivo habiendo recorrido tantos centímetros como se haya calculado.

La figura 5.1 muestra la integración de los módulos.

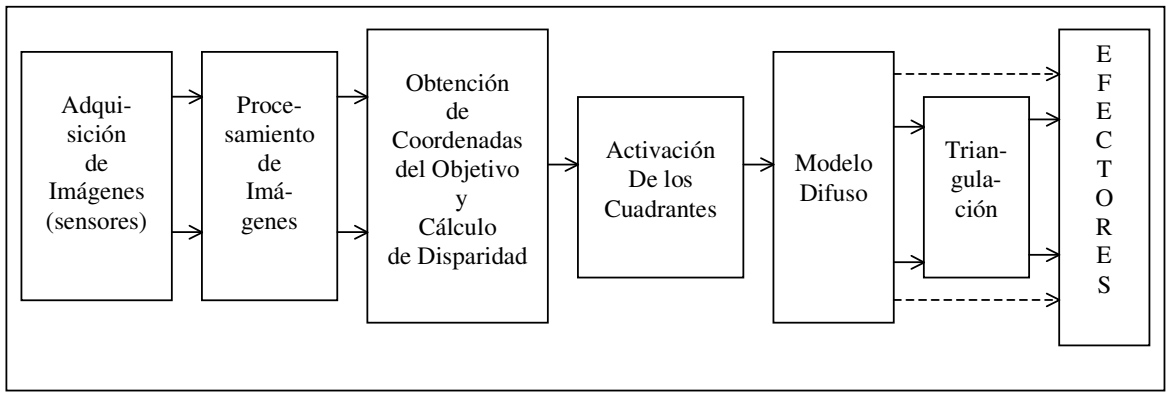


Figura 5.1 Integración de técnicas de procesamiento y modelo difuso

Como se mencionó anteriormente, es de esta manera que se logra una navegación suave hasta ubicar al agente cerca del objetivo para luego ajustar los movimientos permitiendo al robot llegar a la meta de forma más precisa.

5.1.1 Función de Activación de los Cuadrantes

La técnica que se realiza en un primer momento y que ha sido detallada en la sección 3.1 arroja como resultado un valor de disparidad calculado según la activación de los cuadrantes de cada imagen obtenida por las cámaras. Una vez conocida ésta se procede a calcular qué cuadrantes tienen que ser activados para propagar la información de la ubicación del objeto y la distancia. La siguiente función determina qué cuadrantes del modelo difuso deben ser activados:

$$C_i = \begin{cases} Disparidad & para\ CI_i >= 0{,}90\ ó\ CD_i >= 0{,}90 \\ 0 & en\ otro\ caso \end{cases}$$

i es la cantidad de cuadrantes en que están divididas las imágenes.

Cuando los dos pares CI_i y CD_i son menores a 0,90 se propaga el valor cero al cuadrante C_i del modelo difuso. Y se propaga el valor de disparidad en el correspondiente cuadrante del modelo difuso C_i cuando alguno de los pares CI_i y CD_i tiene un nivel mayor o igual a 0,90.

Por ejemplo, se tiene como resultado la activación de cada cuadrante de las imágenes *CI*=(0,0,0,1,0) y *CD*=(0,0,1,1,0), cuya disparidad es igual a 0.2. La función de propagación arrojaría la activación del tercer y cuarto cuadrante con el valor correspondiente a la disparidad y la no activación de los demás cuadrantes.

Es decir, que el conjunto de variables de entrada del modelo difuso para el ejemplo de la sección 3.1.3 sería E = (0; 0; 0,2; 0,2; 0).

5.1.2 Activación de los Efectores

Los efectores se activan de diferente forma para lograr la movilidad del agente y así producir un cambio en el ambiente. En este modelo de navegación, los efectores pueden recibir un comando que les indique a qué velocidad y en qué dirección deben activarse, ya sea directamente del modelo difuso como de los cálculos de triangulación.

Si en el modelo difuso se determina que el objetivo se encuentra a una distancia lejana o mediana del agente, a partir de la agregación de las reglas difusas activadas se logra una navegación suave. Esto se debe a que se activarán los efectores de manera que el agente vaya acercándose lentamente al objetivo. Es decir, los efectores son activados como resultado del modelo difuso.

Si a través del PID se determina que el agente está ubicado cerca del objetivo, sería más conveniente mejorar los movimientos de manera que el robot se acerque de forma precisa a la meta, ya sin depender de la lógica difusa. Esto es porque a través de realizar cálculos por triangulación, se puede conocer la distancia relativa al objeto de forma precisa y no incierta.

Por lo tanto, cuando ambos efectores tomen un valor correspondiente al conjunto difuso de salida Quieto, se procederá a calcular la distancia relativa al objeto y el ángulo y sentido de giro correspondientes. Este resultado puede darse porque el agente se encuentra cerca del objetivo y las reglas del modelo difuso determinen la quietud de los efectores, o bien porque a través de la agregación y defusificación se llega a la conclusión de que los motores deben activarse con una velocidad correspondiente dicho conjunto de salida.

Una vez conocidos los valores de la distancia y el ángulo de giro, se indicará a los efectores que, a través de sus movimientos, el agente gire tantos grados como mida dicho ángulo y en el sentido contrario al de desvío, dejando al agente enfrentado al objeto. Por último, se activarán ambos efectores hacia delante hasta que el agente recorra la distancia calculada como la relativa a la meta y se encuentre con ella. De esta manera, los efectores son activados como resultado del cálculo por triangulación.

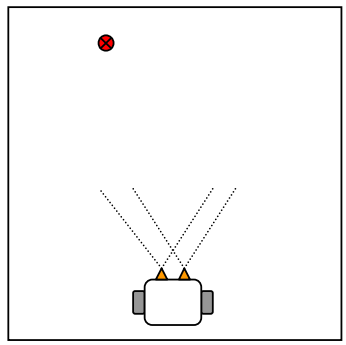
5.2 Navegación autónoma

La navegación autónoma consiste en la capacidad que posee un agente autónomo móvil de moverse en el ambiente sin la intervención humana. Los objetivos de la navegación autónoma pueden ser muy variados, como por ejemplo, alcanzar una meta, esquivar obstáculos, reaccionar en tiempo real a acontecimientos inesperados como la aparición repentina de obstáculos o construir, utilizar y mantener mapas del ambiente.

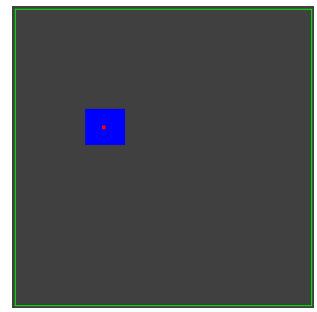
La finalidad de este modelo de navegación autónoma es la de alcanzar una y sólo una meta dada representada a través de un objeto con características de color muy contrastante con el ambiente.

5.2.1 Secuencia de Percepciones y Acciones

A continuación se muestra una secuencia de percepciones y acciones que toma un agente autónomo móvil que tiene dicho objetivo y está ubicado en un ambiente con dichas características. Su meta será el objeto rojo de la escena.

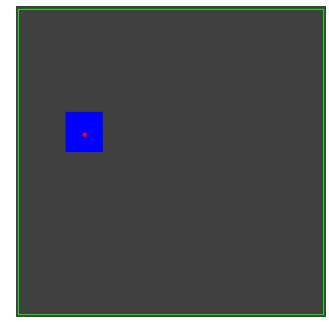


Cámara Izquierda



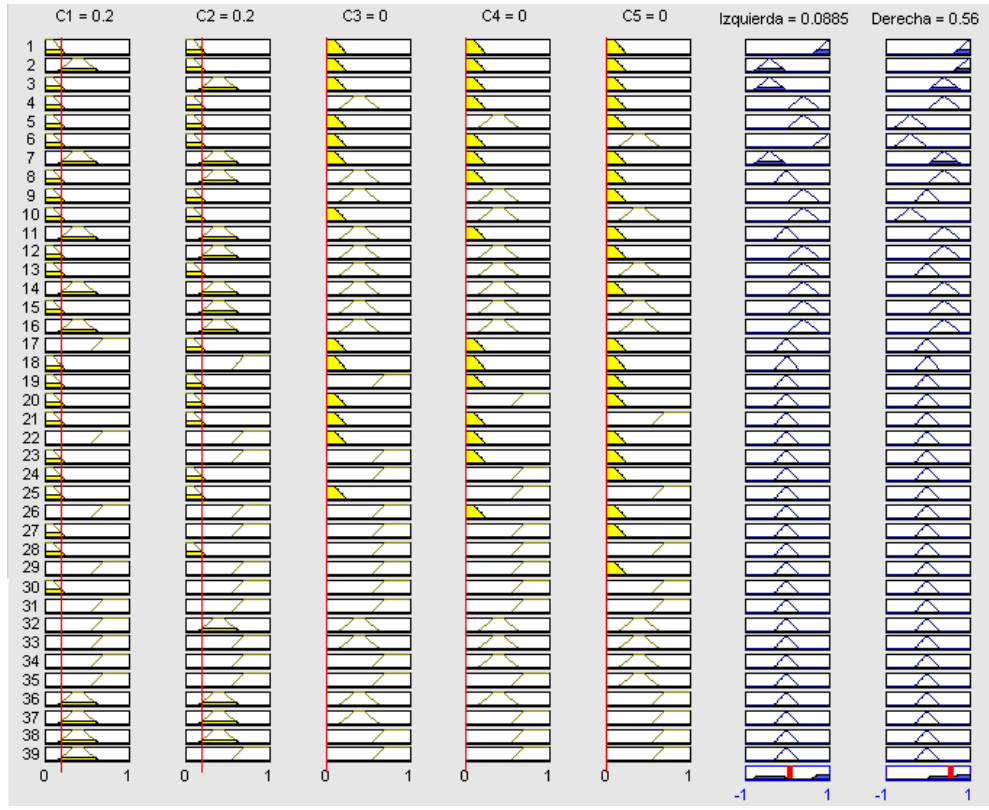
Centroide: (26,57)
Esquina superior izquierda: (21,49)
Esquina inferior derecha: (32,66)
Cuadrantes Activados: (0,1,0,0,0)

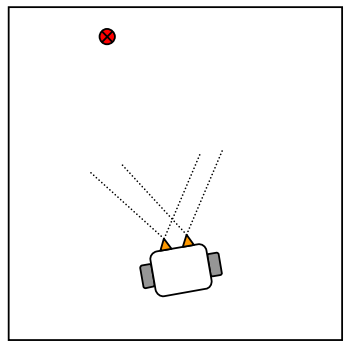
Cámara Derecha



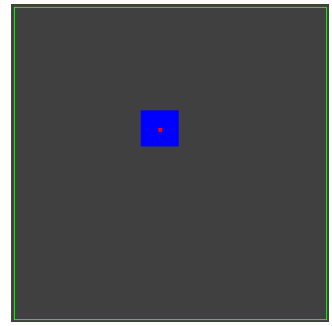
Centroide: (19,56)
Esquina superior izquierda: (14,49)
Esquina inferior derecha: (25,68)
Cuadrantes Activados: (1,1,0,0,0)

Disparidad: 0.2



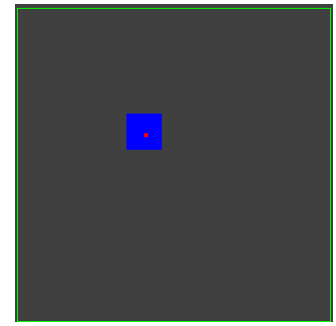


Cámara Izquierda



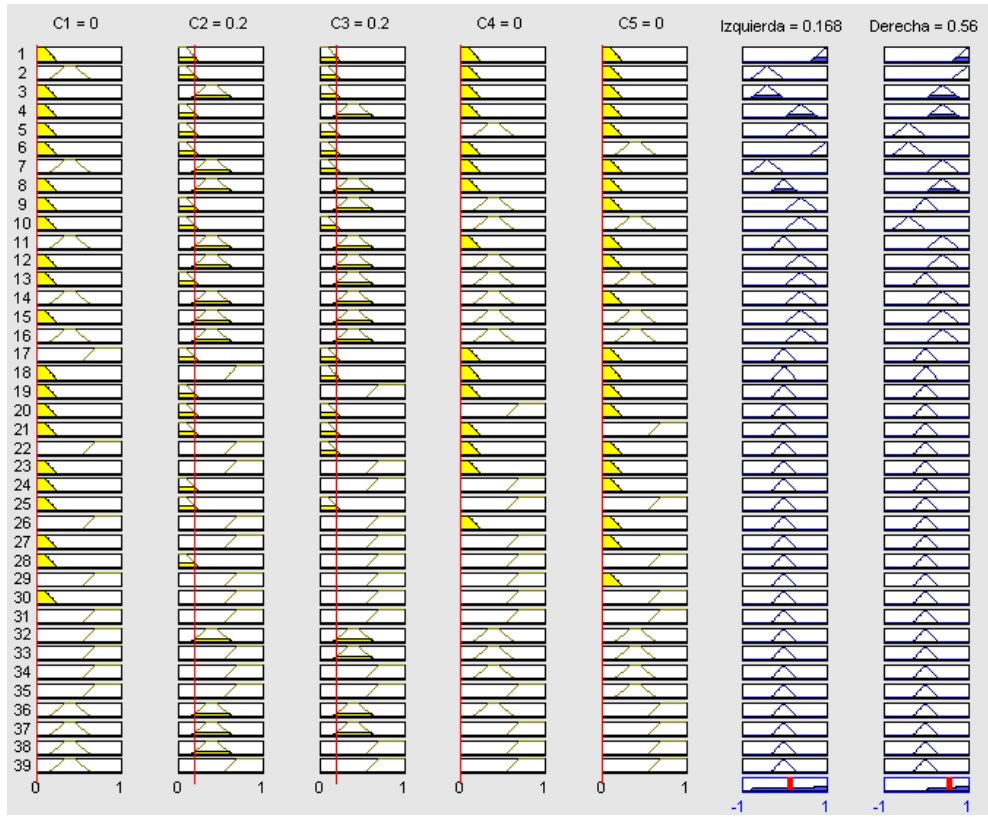
Centroide: (41,56)
Esquina superior izquierda: (36,48)
Esquina inferior derecha: (47,65)
Cuadrantes Activados: (0,0,1,0,0)

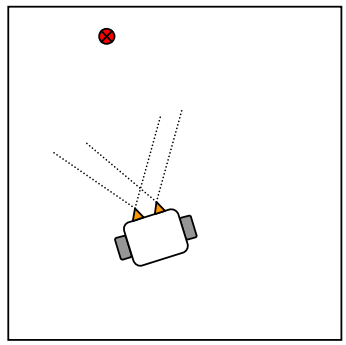
Cámara Derecha



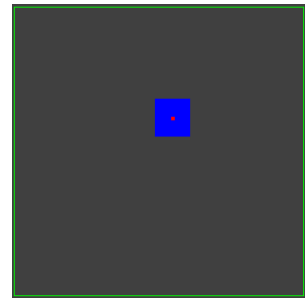
Centroide: (35,58)
Esquina superior izquierda: (31,49)
Esquina inferior derecha: (41,66)
Cuadrantes Activados: (0,1,1,0,0)

Disparidad: 0.2



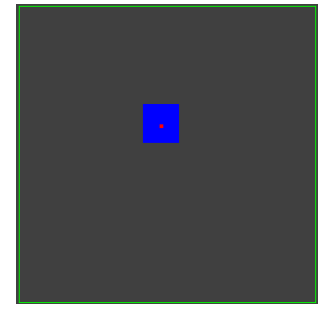


Cámara Izquierda



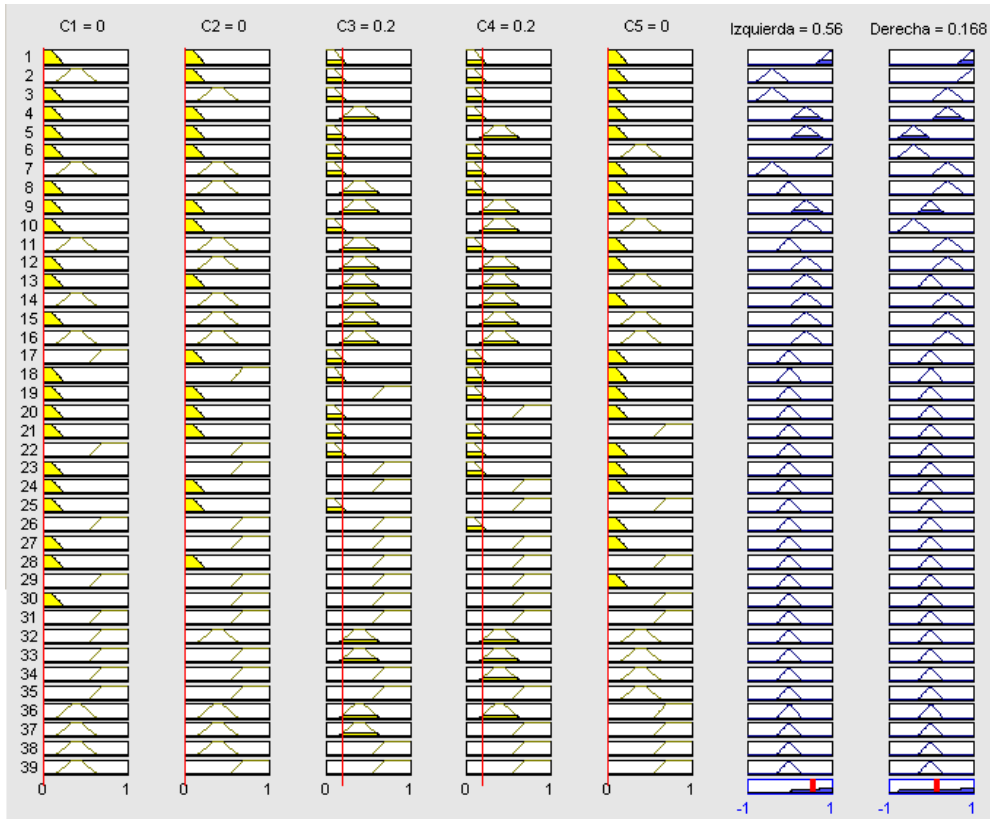
Centroide: (48,55)
Esquina superior izquierda: (43,46)
Esquina inferior derecha: (54,65)
Cuadrantes Activados: (0,0,1,1,0)

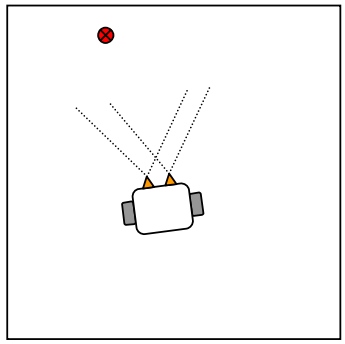
Cámara Derecha



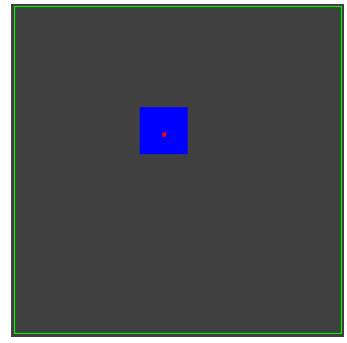
Centroide: (42,58)
Esquina superior izquierda: (37,48)
Esquina inferior derecha: (48,67)
Cuadrantes Activados: (0,0,1,0,0)

Disparidad: 0.2



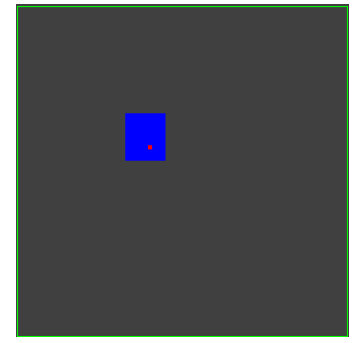


Cámara Izquierda



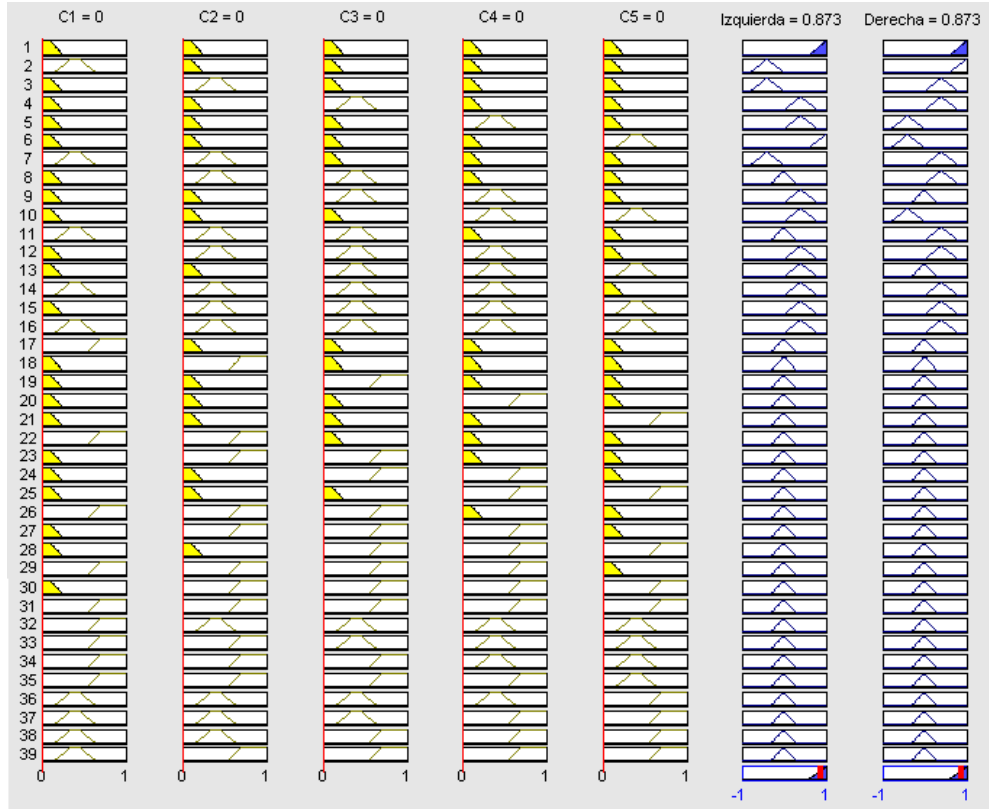
Centroide: (40,56)
Esquina superior izquierda: (34,45)
Esquina inferior derecha: (47,66)
Cuadrantes Activados: (0,1,1,0,0)

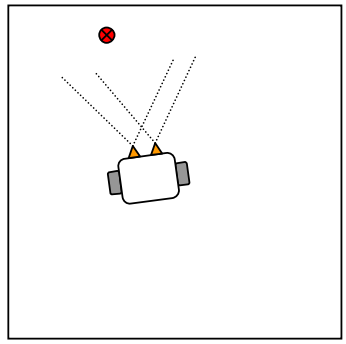
Cámara Derecha



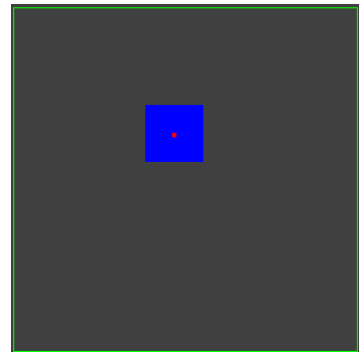
Centroide: (35,61)
Esquina superior izquierda: (29,47)
Esquina inferior derecha: (40,68)
Cuadrantes Activados: (0,1,1,0,0)

Disparidad: 0



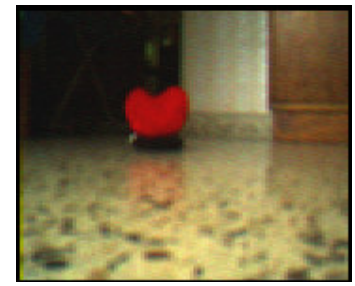


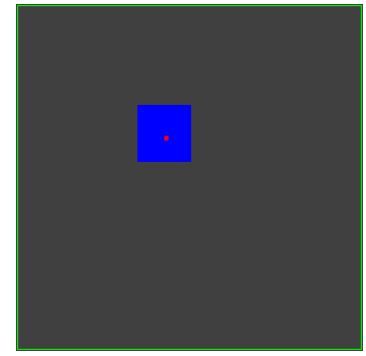
Cámara Izquierda



Centroide: (41,53)
Esquina superior izquierda: (34,41)
Esquina inferior derecha: (49,65)
Cuadrantes Activados: (0,1,1,0,0)

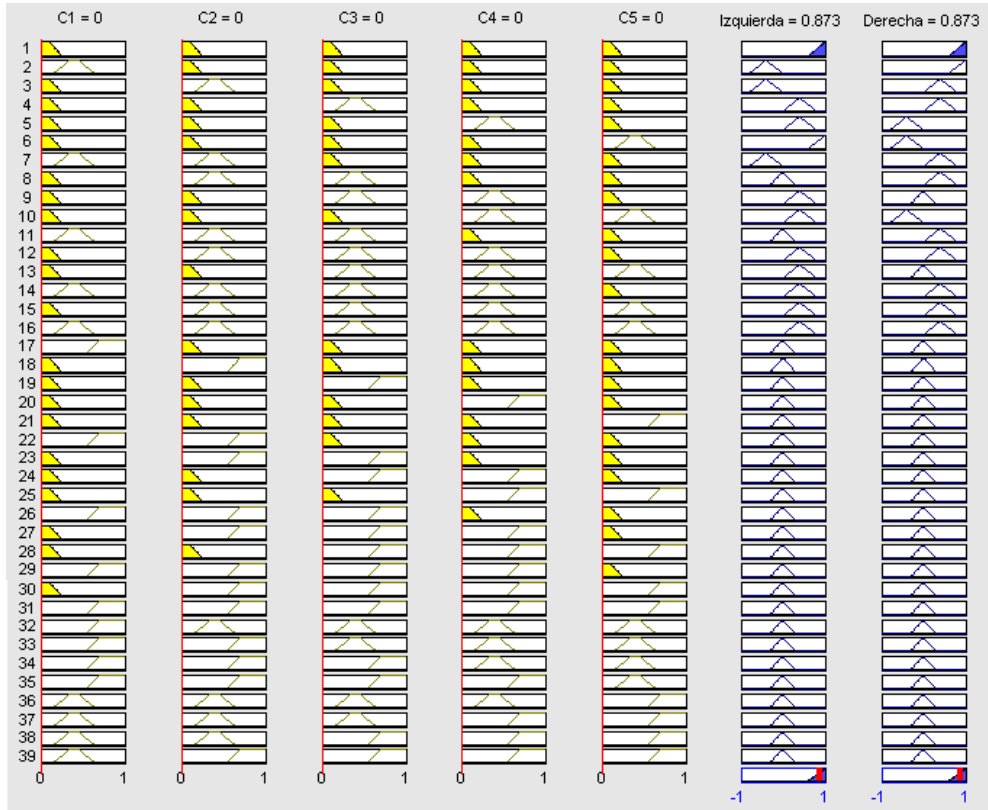
Cámara Derecha

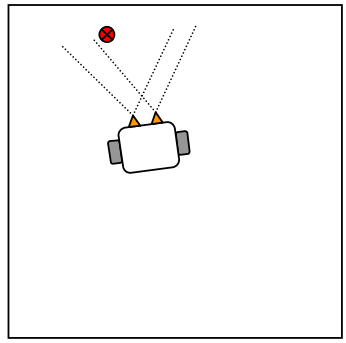




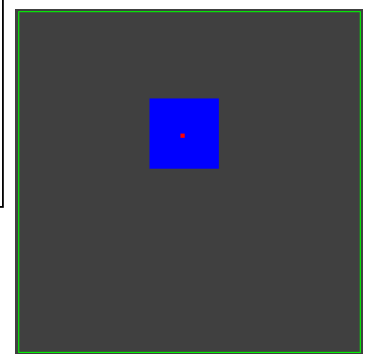
Centroide: (38,55)
Esquina superior izquierda: (31,42)
Esquina inferior derecha: (45,66)
Cuadrantes Activados: (0,1,1,0,0)

Disparidad: 0



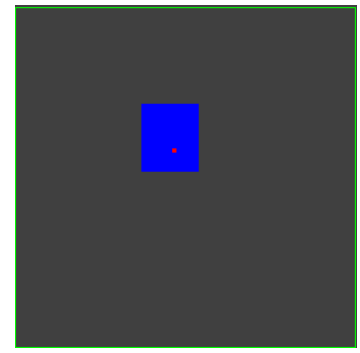


Cámara Izquierda



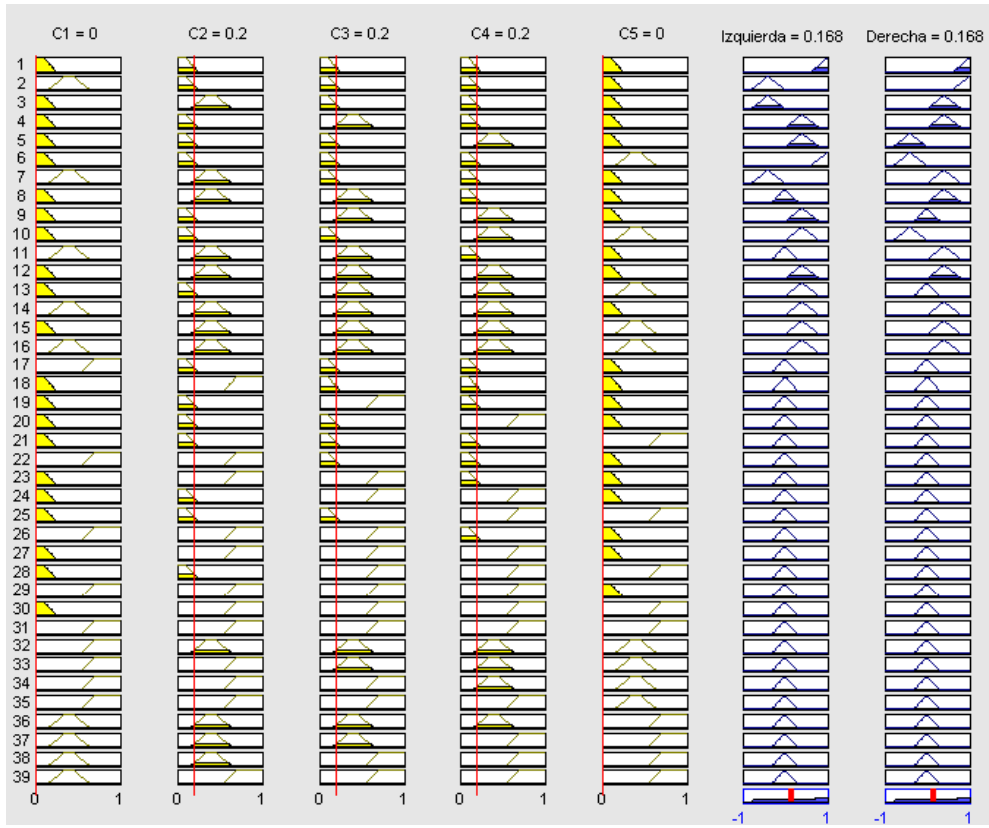
Centroide: (42,52)
Esquina superior izquierda: (34,37)
Esquina inferior derecha: (52,67)
Cuadrantes Activados: (0,1,1,1,0)

Cámara Derecha

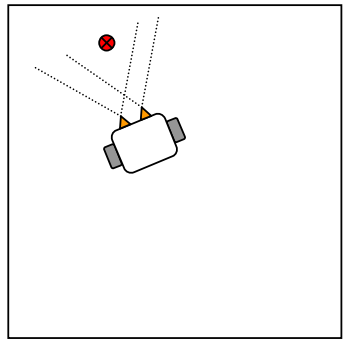


Centroide: (41,61)
Esquina superior izquierda: (33,41)
Esquina inferior derecha: (48,70)
Cuadrantes Activados: (0,1,1,0,0)

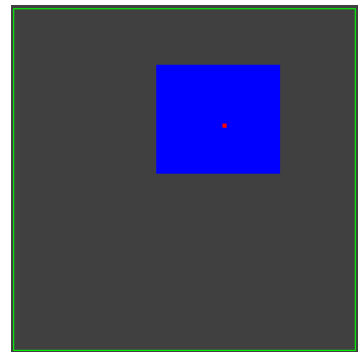
Disparidad: 0.2



Distancia: 71.495 cm
Angulo de Giro: 0.2336º
Dirección de Desvío: Derecha

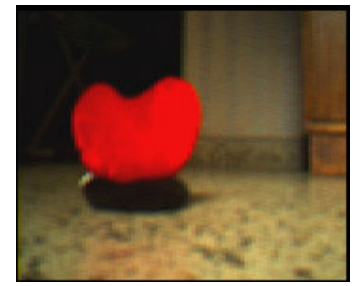


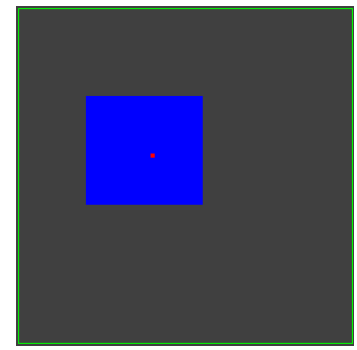
Cámara Izquierda



Centroide: (54,49)
Esquina superior izquierda: (37,24)
Esquina inferior derecha: (69,70)
Cuadrantes Activados: (0,0,1,1,1)

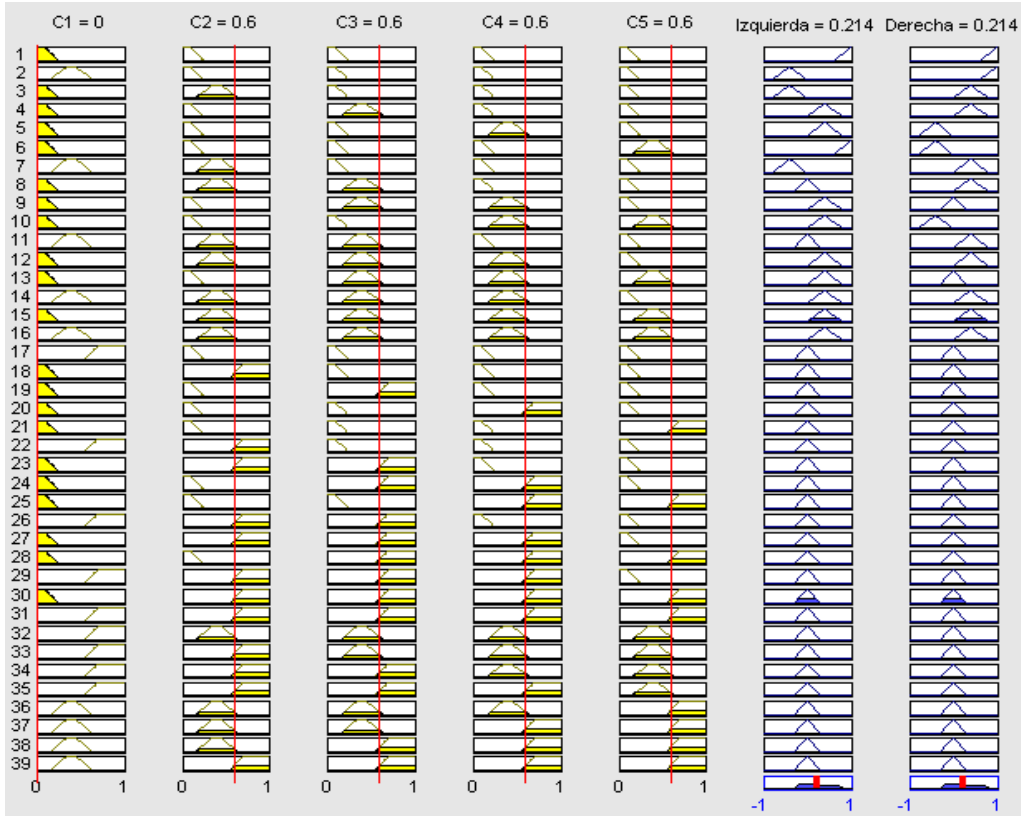
Cámara Derecha



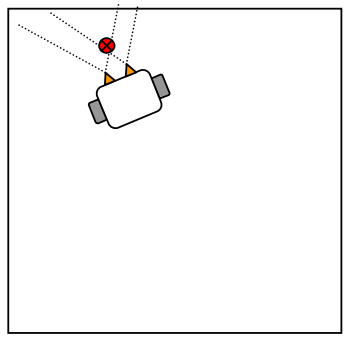


Centroide: (35,63)
Esquina superior izquierda: (18,38)
Esquina inferior derecha: (49,85)
Cuadrantes Activados: (0,1,1,0,0)

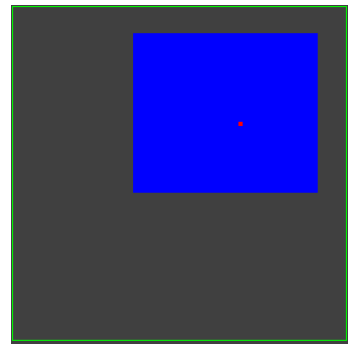
Disparidad: 0.6



Distancia: 36.056 cm
Angulo de Giro: -5.68E-14º
Dirección de Desvío: Derecha

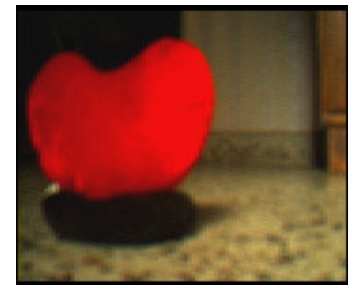


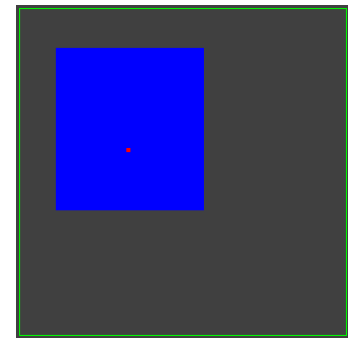
Cámara Izquierda



Centroide: (60,50)
Esquina superior izquierda: (32,12)
Esquina inferior derecha: (81,81)
Cuadrantes Activados: (0,1,1,1,1)

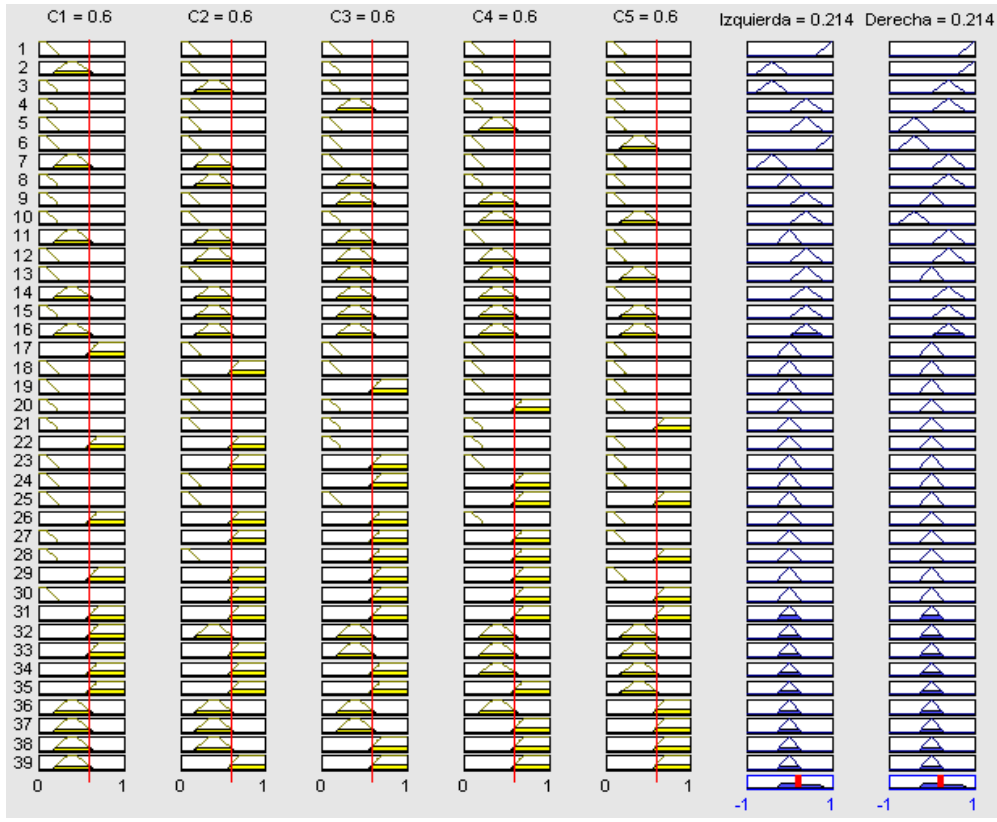
Cámara Derecha





Centroide: (29,61)
Esquina superior izquierda: (10,19)
Esquina inferior derecha: (51,90)
Cuadrantes Activados: (1,1,1,0,0)

Disparidad: 0.6



Distancia: 26.1186 cm
Angulo de Giro: 0.0º
Dirección de Desvío: Derecha

Según la secuencia ejemplificada anteriormente, el camino resultante hacia la meta se muestra en la figura 5.2.

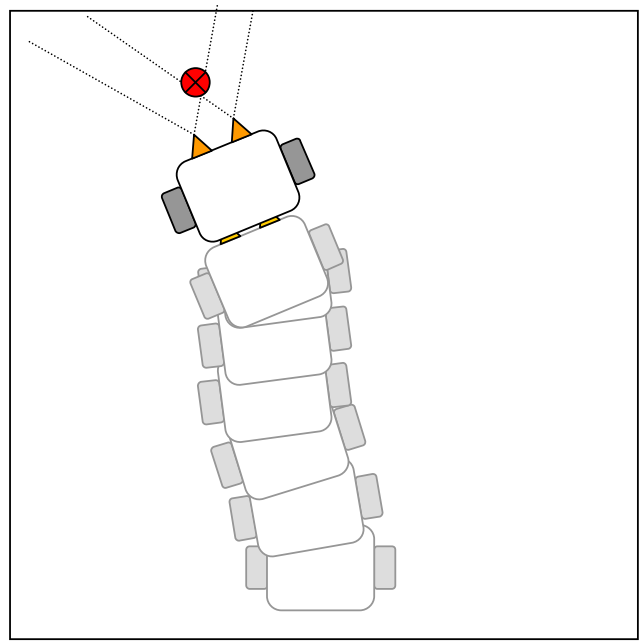


Figura 5.2 Ejemplo de secuencia de percepciones y acciones.

5.2.2 Observaciones

Durante las experiencias y pruebas realizadas se observó que las CMUcam2 son muy sensibles a la luz ambiente. Si bien se intentó estabilizar las fluctuaciones en los datos resultantes por los cambios en la claridad o luz externa haciendo uso del parámetro Auto-gain, los datos erróneos como distancia relativa y ángulo de giro debido a zonas oscuras o claras no se han podido evitar. Estos datos se pudieron corregir en tiempo real cambiando las zonas afectadas por las sombras o brillos a través de la introducción de otros elementos en el ambiente.

Cuando el ambiente está iluminado por luz artificial o está oscuro es prácticamente imposible discriminar un objeto, cualquiera sea su color. Si, en cambio el ambiente está iluminado por luz natural, los colores se distinguen más fácilmente. Ahora, si la luz artificial es blanca y además está direccionada a la zona captada por las cámaras, los objetos pueden encontrarse, siendo los mayormente distinguibles aquellos de tonos puros (es decir, cuando algún valor del RGB se encuentra cercano al máximo de 240).

En cuanto a la forma del objeto a detectar, se observa que aquellos que son más anchos que altos presentan la particularidad de ocupar más cuadrantes de cada imagen aún cuando están lejos del agente. Esto puede confundir a la hora de obtener la disparidad ya que, al ser similares los cuadrantes activados en cada cámara, aún estando cerca del agente la disparidad puede ser 0 o cercana a 0. Este inconveniente podría solucionarse dividiendo las imágenes en más cantidad de cuadrantes.

Por otro lado, cuando los objetos son más altos que anchos, el cálculo de disparidad es más claro, ya que a medida que se acerca el agente al objeto, los cuadrantes activados en cada imagen seguramente son diferentes, dando como resultado una disparidad mayor a cuando están lejos del mismo. Como contrapartida, cuando están demasiado cerca del agente, pueden entrar más rápidamente en la zona ciega de las cámaras y no ser vistos por las mismas.

5.3 Sistemas de Control para agentes autónomos

Un sistema de control para robots móviles consiste en una arquitectura que procesa información compleja para llevar a cabo tareas en tiempo real. Rodney Brooks planteó en el año 1985 un modelo de sistema de control en capas al cual llamó Subsumisión que le permite al agente operar en niveles de competencia incrementales [Brooks86]. Esta es una idea diferente a lo que se planteaba anteriormente, lo cual consistía en descomponer las tareas de un robot en capas verticales y totalmente dependientes unas de otras para el completo funcionamiento del mismo (Figura 5.3). La información fluía desde el ambiente hacia el robot y de él al ambiente nuevamente, y debía construirse cada capa para que el robot funcionara, además de que algún cambio en una de ellas debería propagarse a las demás.

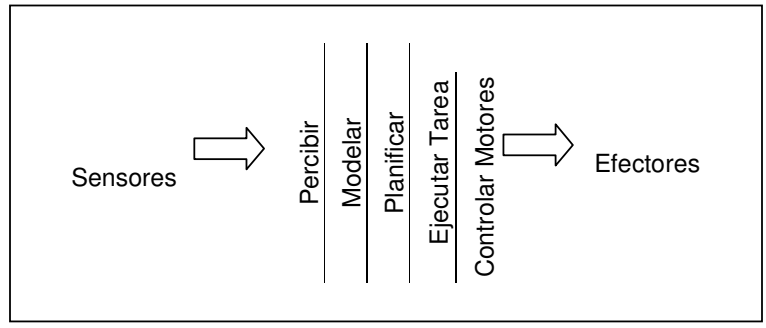


Figura 5.3 Sistema de control de la primera era de la robótica

El modelo de Subsumisión propone capas creadas como módulos asincrónicos, donde cada una se comunica con las otras y consisten en máquinas computacionales simples (Figura 5.4). Cada capa define un nivel de competencia, es decir, una especificación de un tipo de comportamiento deseado. Cuánto mayor sea el nivel de la capa, más específico será el comportamiento. Este sistema de control debe su nombre al hecho de que las capas de niveles más altos subsumen el papel de las capas de más bajo nivel suprimiendo sus salidas. Igualmente, los niveles más bajos siguen funcionando cuando los más altos toman el control. Debe tenerse en cuenta que este modelo es muy rígido al momento de modificar la arbitración de los comportamientos, ya que se plantea en tiempo de diseño no pudiendo modificar la estrategia en tiempo de ejecución.

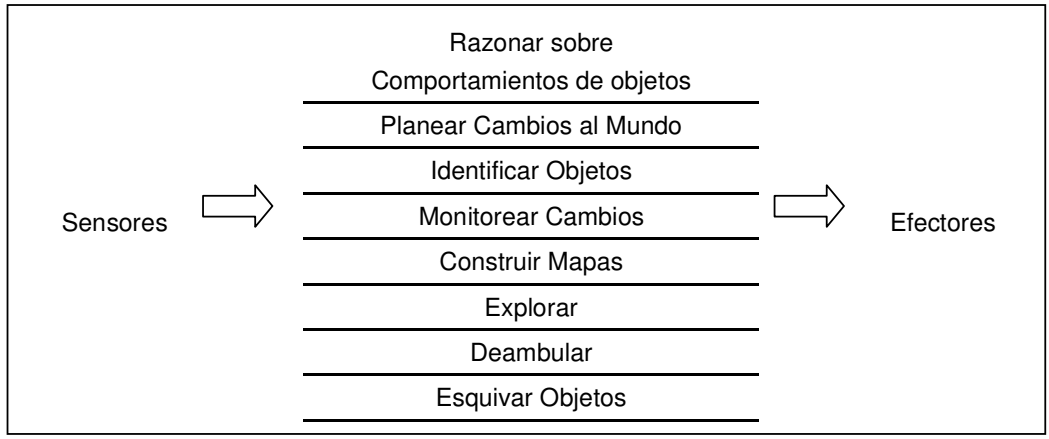


Figura 5.4 Sistema de control basado en comportamiento, Subsumisión

En 1997, Saffiotti y Ruspini toman las ideas de Brooks de un sistema de control organizado en comportamientos y desarrollan uno más flexible donde enfatizan:

- Cómo diseñar comportamientos simples que garanticen operaciones robustas aún con poca información en tiempo de diseño
- Cómo coordinar varias tareas que posiblemente sean contradictorias para lograr realizar una tarea más compleja
- Cómo asegurar la coherencia entre las representaciones de las distintas capas

Este modelo se denomina Context-dependent bleding (CDB) y permite mediante reglas de contexto difusas coordinar la activación de cada comportamiento o activarlos en forma simultánea según el contexto percibido por el agente autónomo [Saffiotti97].

5.3.1 Context-Dependent Blending – CDB

El CDB es un mecanismo general que permite expresar diferentes conductas a través de meta-reglas difusas o reglas de contexto. Estas proporcionan un medio para escribir estrategias de coordinación complejas de forma modular y jerárquica usando un formato lógico. (Figura 5.5)

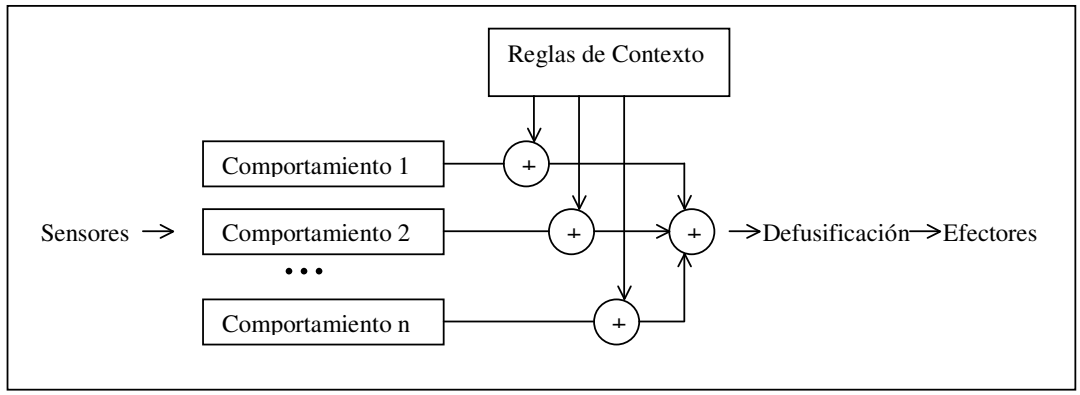


Figura 5.5 Sistema de control Context-Dependent Blending

Las reglas de comportamiento tiene la siguiente forma:

IF *CONEXTO* THEN *COMPORTAMIENTO*

Por ejemplo:

If Obstáculo then esquivar-obstáculo

If –(Obstáculo) then ir-a-la-meta

Es decir, de esta manera un agente autónomo podrá llevar a cabo diferentes comportamientos en forma simultánea y según el contexto en el que se encuentre en ese momento. Además, cada comportamiento es implementado independientemente con la única restricción de mantener la coherencia entre los ellos.

5.3.2 Modelo de Visión Estereoscópica Difusa implementado en CDB

El modelo planteado en este trabajo tiene como objetivo detectar un objeto en el ambiente y dirigir al agente hacia él. Es decir, se trata de un comportamiento independiente que puede ser implementado perfectamente en un sistema de control para robots como el planteado por Saffiotti. Esta implementación se describe en la figura 5.6.

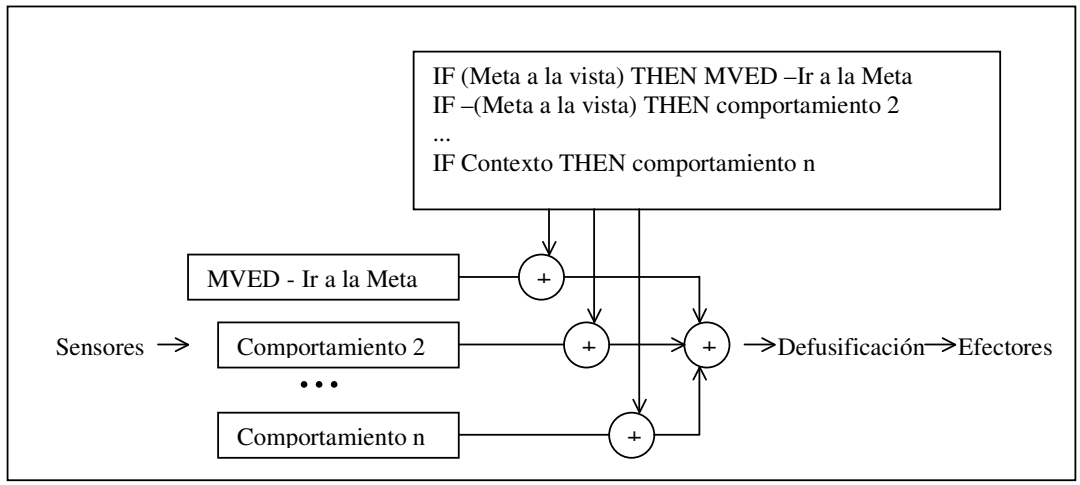


Figura 5.6 Implementación del Modelo de Visión Estereoscópica Difusa en el sistema de control CDB.

Conclusión

Uno de los comportamientos necesarios en un agente móvil es el de detectar objetos y ubicarlos en el ambiente en que se mueve. Esto puede ser por diferentes razones, como por ejemplo, para esquivar obstáculos, para dirigirse a una meta, descubrir intrusos u objetos no deseados, entre otros.

Este modelo presenta dos formas diferentes de determinar la distancia del objeto detectado y el agente autónomo, las cuales pueden ser utilizadas de manera independiente según la circunstancia.

A través del cálculo de disparidad y la integración al modelo difuso se determina la distancia de una manera aproximada permitiendo a través del razonamiento incierto lograr un modelo robusto y flexible que le brinda al agente la posibilidad de moverse hacia al objetivo en forma suave. Por sí mismos, los sistemas de inferencia difusa permiten interpolar resultados entre las reglas difusas produciendo estos movimientos suaves, siendo esto un punto importante para que los robots móviles puedan relacionarse con los seres humanos y otros seres vivos.

Por triangulación, según la geometría del sistema estereoscópico es posible conocer exactamente la distancia entre el objetivo y el agente. De la misma manera que se puede calcular el ángulo y dirección de giro para ubicar al robot directamente en frente al objeto. Cuando el agente esté cerca de la meta, si se utilizan efectores precisos, se puede guiar al robot hacia el objetivo con una cantidad mínima de movimientos y recorriendo la distancia exacta según estas medidas calculadas por funciones trigonométricas.

Otra ventaja de este modelo es la independencia entre los sensores y efectores. Es decir, cambiando los efectores se podría utilizar este mismo modelo para distintos propósitos. Por ejemplo, usando un efector de síntesis de voz, se podría informar el lugar de ubicación de un objeto y tomar así alguna decisión. O bien, conociendo la distancia y ángulo de giro exacto se podría alcanzar el objetivo de diferentes formas como por ejemplo disparando un proyectil o moviendo un brazo robótico. Además, podrían modificarse las reglas difusas para que el agente se comporte de otras maneras ante un objeto.

Trabajos Futuros

Los trabajos futuros consistirán en mejorar la percepción de las cámaras utilizadas o bien integrar este modelo a sensores más precisos y menos sensibles a ruidos y luminosidad externa.

Además, se proyecta implementar este modelo en plataformas robóticas que la universidad planea adquirir. En estos mismos agentes se podrían incorporar otros modelos de comportamiento, como por ejemplo el modelo Conexionista-Difuso propuesto por Longstaff [Longstaff02]. De esta manera, el robot podría dirigirse hacia una meta específica utilizando el modelo planteado en este trabajo y en su camino hacia ella esquivar obstáculos haciendo uso del modelo Conexionista-Difuso. Se obtendría así, un sistema de navegación más completo basado en CDB.

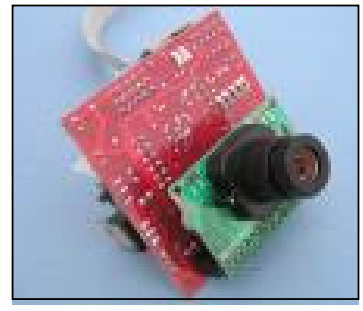
Y en relación con trabajos de investigación propiamente dichos, se avanzará con el estudio de sistemas de control basados en comportamiento, para intentar encontrar modelos para el control de los comportamientos de los agentes autónomos de la manera más eficiente y robusta posible.

Anexo A: CMUCam2

Las cámaras CMUcam2 consisten en un microcontrolador SX52 conectado a una cámara CMOS OV6620 sobre un chip que permite extraer datos de alto nivel de forma simple desde el flujo de video de la cámara. Esta plataforma se conecta a la computadora por un RS-232 o puerto serial TTL y ofrece, entre otras, las siguientes funcionalidades:

- Seguimiento de regiones de color definidos por el usuario de hasta 50 cuadros por segundo (50 fps).
- Seguimiento de movimiento utilizando diferencias entre cuadros hasta 26 fps.
- Determinación del centroide de cualquier área de datos.
- Generación de histogramas de 28 rangos para cada uno de los canales de color.
- Transferencia de mapas binarios de los píxeles capturados en una imagen en tiempo real.
- Control de 5 servos de salida.

Con las cámaras se puede trabajar completamente desde línea de comandos o bien utilizando una interfaz gráfica de usuario (GUI) que permite realizar el trabajo más fácilmente, la cual está desarrollada en el lenguaje Java.

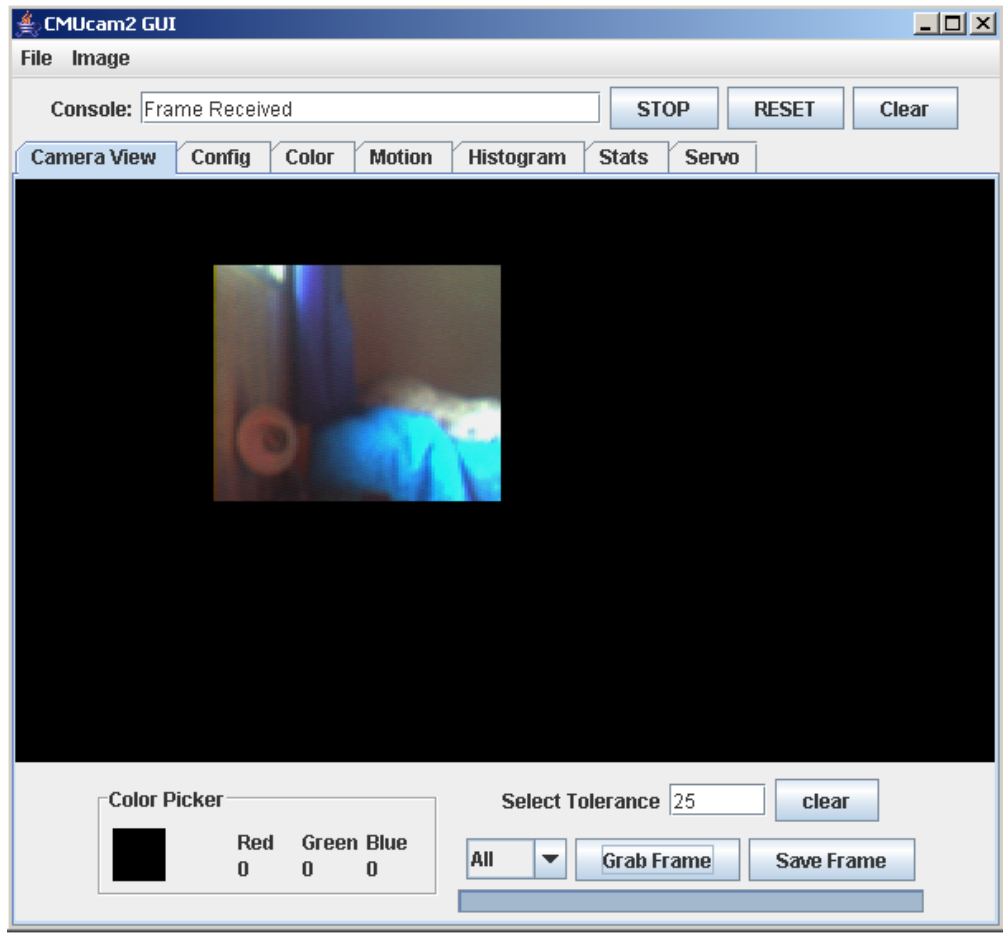


A.1 CMUcam2GUI

La interfaz gráfica provista por los desarrolladores de la CMUcam2 es un programa realizado en lenguaje Java, de código abierto y que puede obtenerse desde Internet. A continuación se describen brevemente cada una de sus funcionalidades.

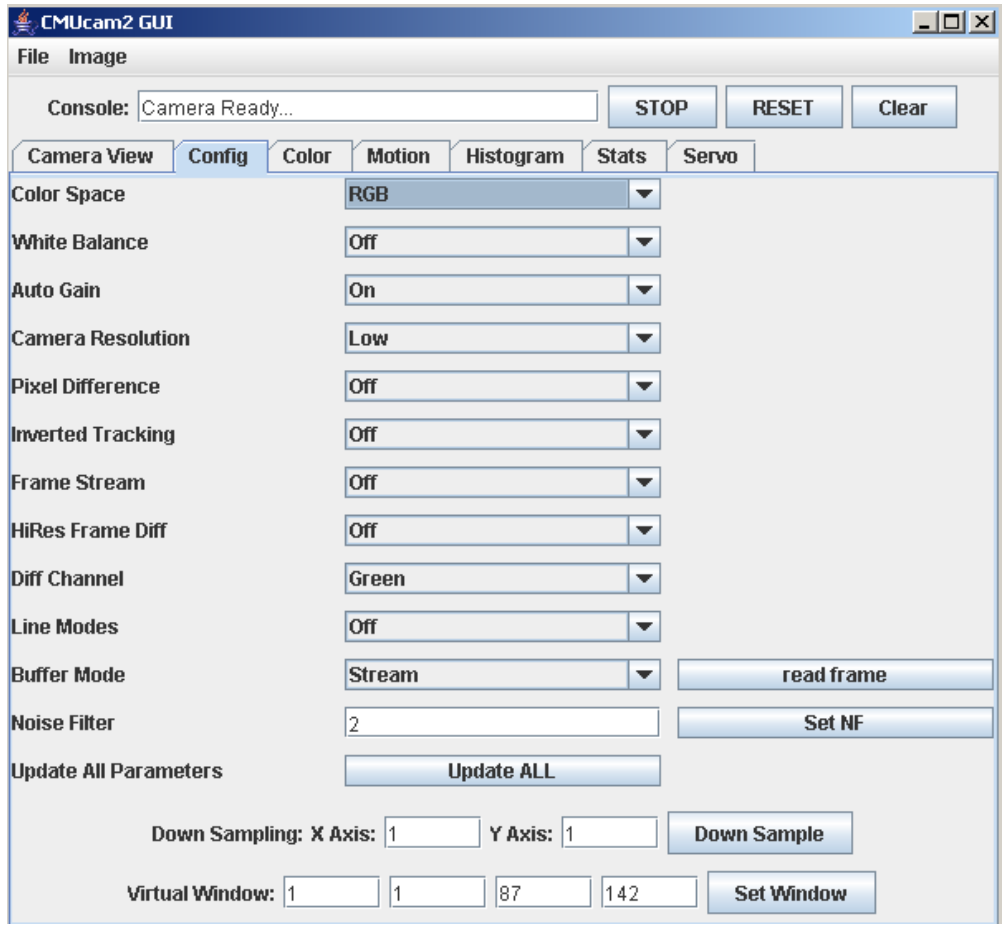
A.1.1 Panel Principal

En este panel se encuentran las opciones principales de la cámara, como son comenzar a grabar, detenerla, resetearla, limpiar la memoria de comandos y enviar los comandos escritos en el campo de consola al puerto de la cámara. Además se pueden seleccionar los canales de los cuales se desea recibir información y grabar la imagen capturada como jpg. También se tiene acceso desde aquí a los demás paneles.



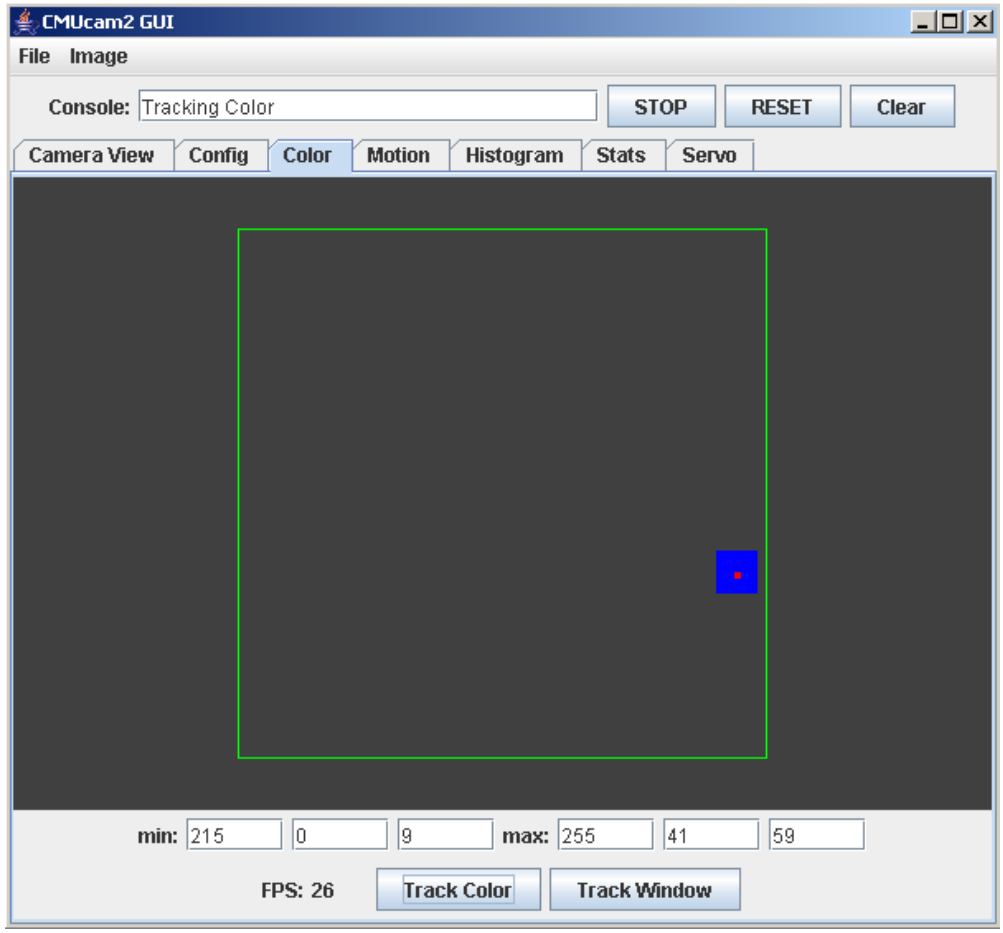
A.1.2 Panel de Configuración

En este panel se pueden configurar todos los parámetros de la cámara, excepto los relacionados a los servos. Cada opción se corresponde con un comando que puede ser enviado a la cámara junto a su valor por línea de comandos.



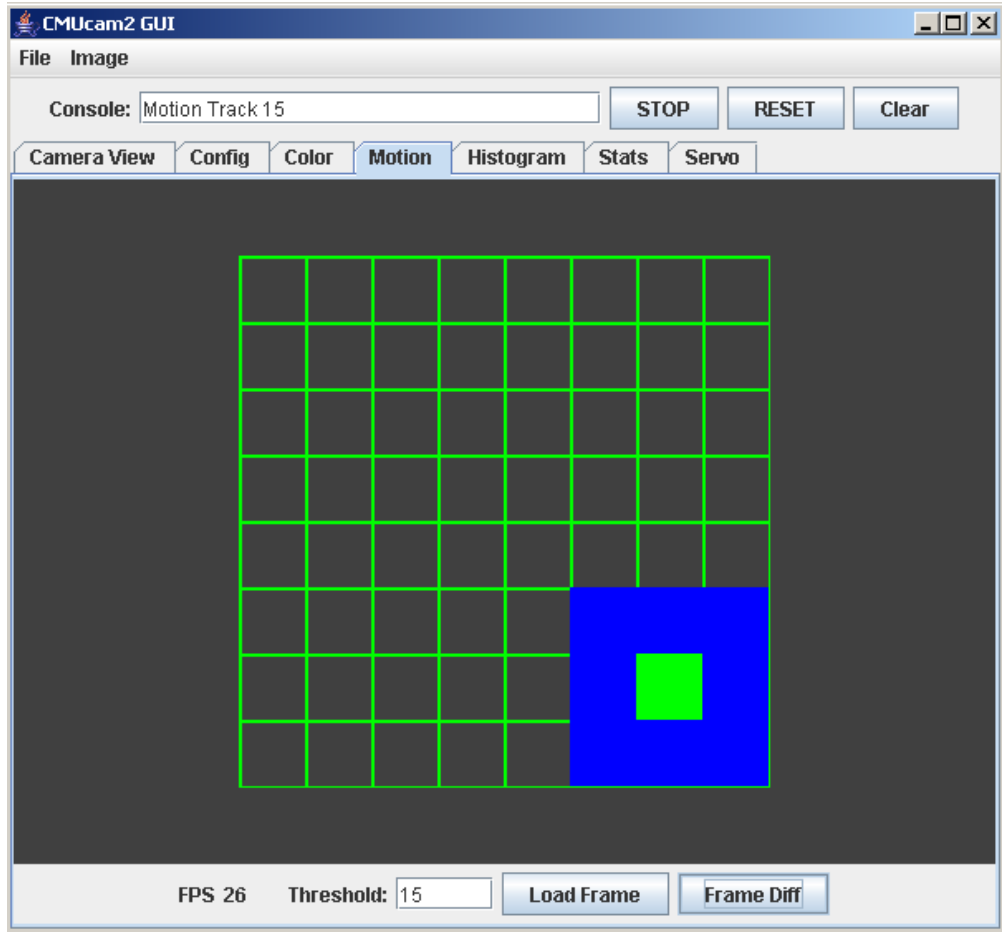
A.1.3 Panel de Color

En este panel se determinan los valores límites de color del área que se quiere detectar, y se ejecuta la función de ubicación de objetos por color. Además muestra en la pantalla la ubicación del objeto demarcando sus bordes y su centro.



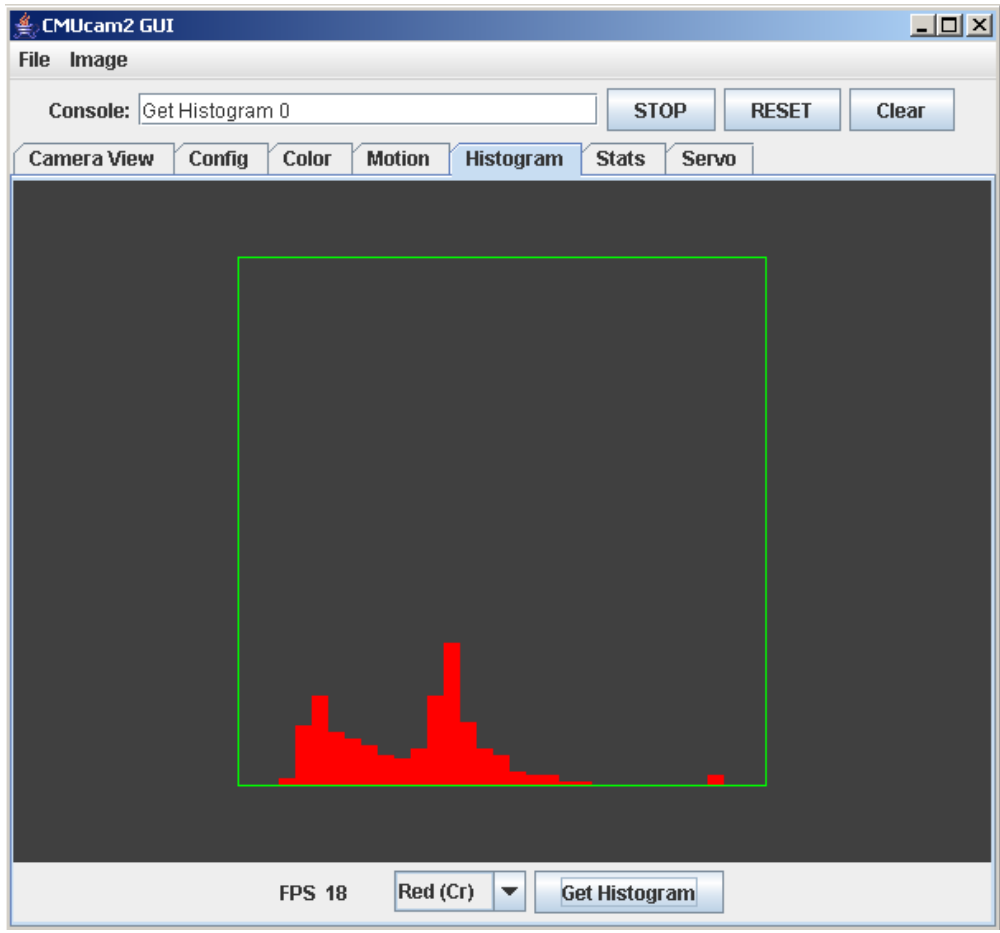
A.1.4 Panel de Movimiento

Este panel permite utilizar la función de diferencia entre cuadros o frames de la cámara. En la parte inferior se indica el umbral y se cargan las imágenes para luego obtener la diferencia entre ellas.



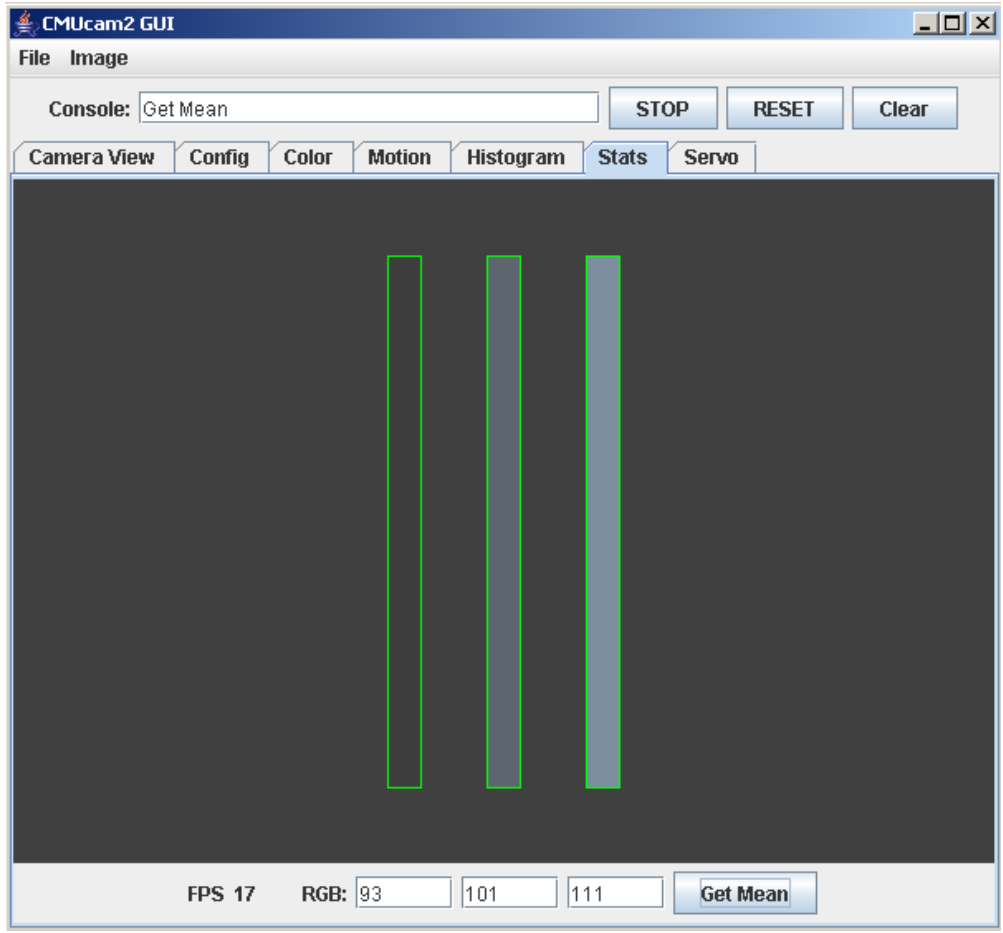
A.1.5 Panel de Histogramas

En este panel se observan los histogramas sobre los píxeles de las imágenes. Si la función de diferenciación está habilitada es posible obtener información sobre bordes brillantes en las imágenes. Esta función puede ser utilizada para determinar el foco y la textura encontrada en la escena.



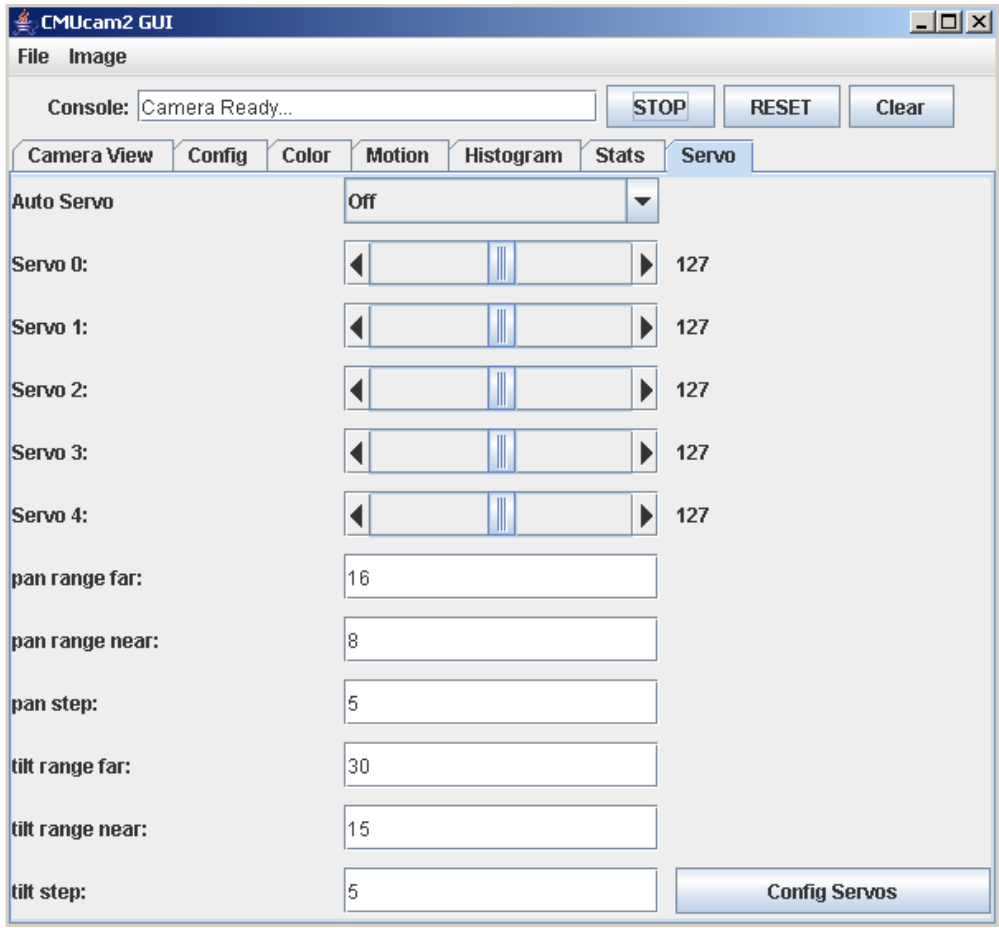
A.1.6 Panel de Estadísticas

Este panel permite utilizar la función de estadísticas de la cámara. En la primer columna se muestra el color promedio menos el desvío. En la segunda se observa el color promedio y la última columna muestra el color promedio más el desvío. Además se pueden conocer estos valores numéricos en el inferior de la pantalla.



A.1.7 Panel de Servos

En este panel se configuran y testean diferentes características de los servos.

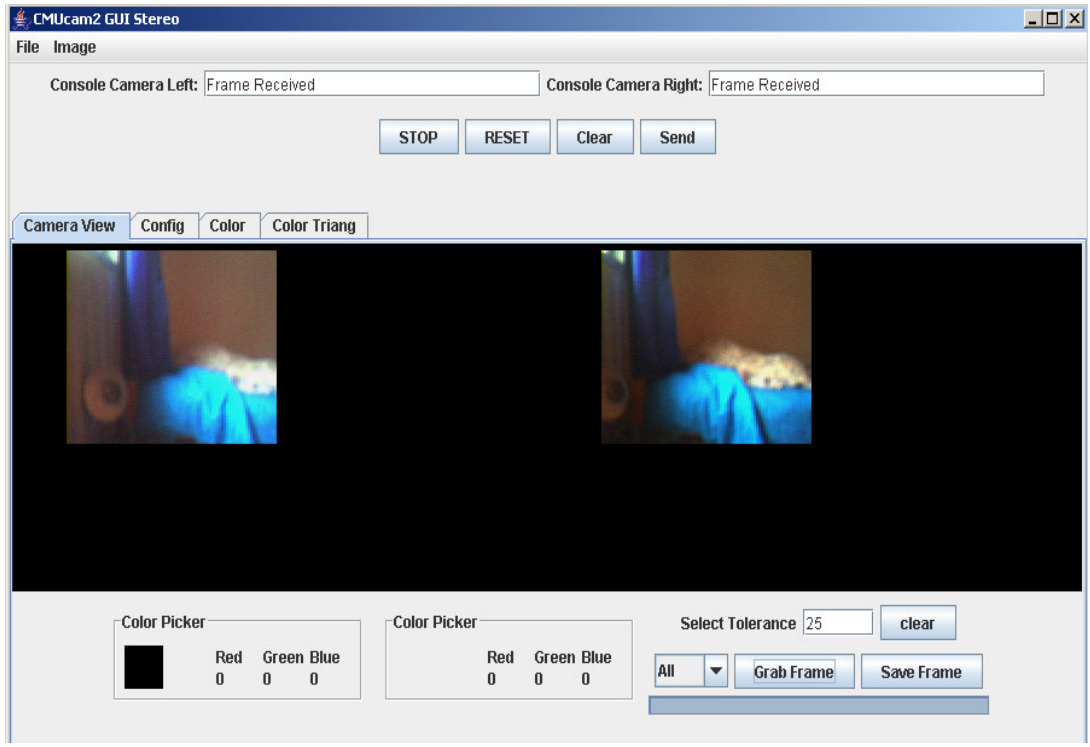


A.2 CMUcam2GUIStereo

La interfaz CMUcam2GUIStereo es una interfaz de usuario creada a partir de la CMUcam2GUI y adaptada para la utilización de dos cámaras en simultáneo para permitir la simulación de la visión estereoscópica. Su funcionalidad sólo contempla lo necesario para cumplir con los objetivos de este trabajo de investigación, los cuales se exponen brevemente a continuación.

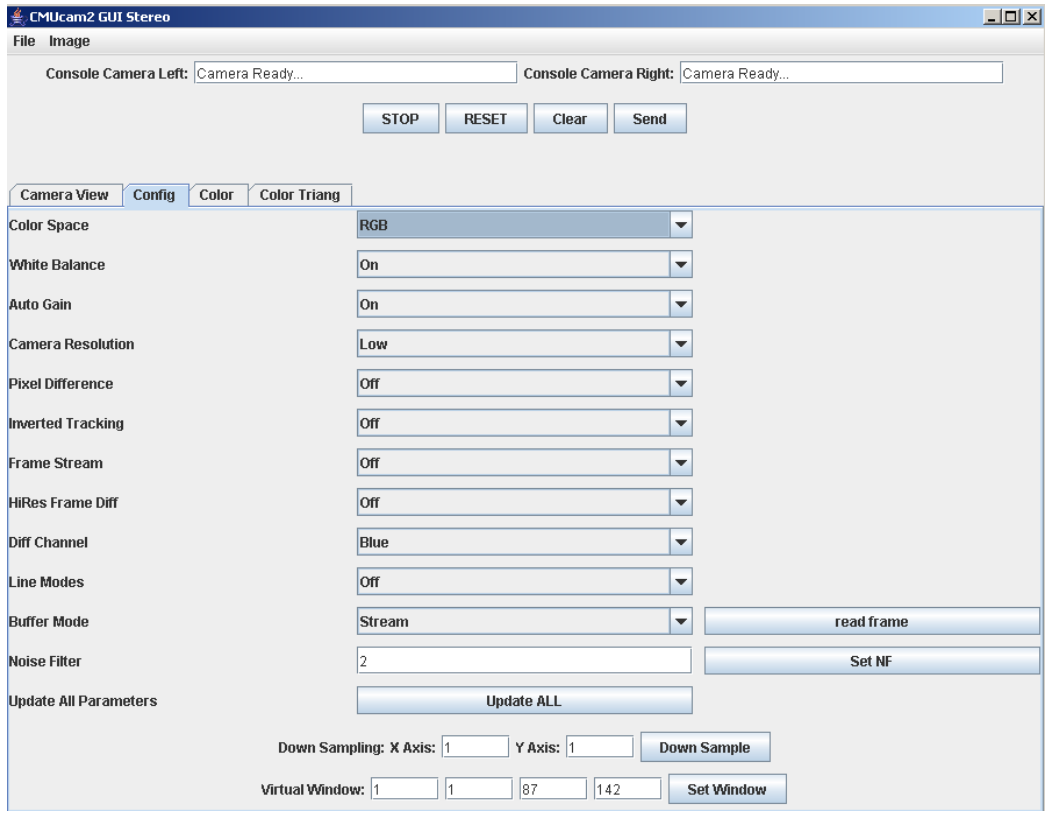
A.2.1 Panel Principal

En este panel se encuentran las mismas opciones que en la interfaz original solo que se ejecutan para ambas cámaras al mismo tiempo. Estas opciones incluyen grabar, detener, resetear, limpiar la memoria de comandos y enviar los comandos escritos en el campo de consola a cada puerto de cada cámara.



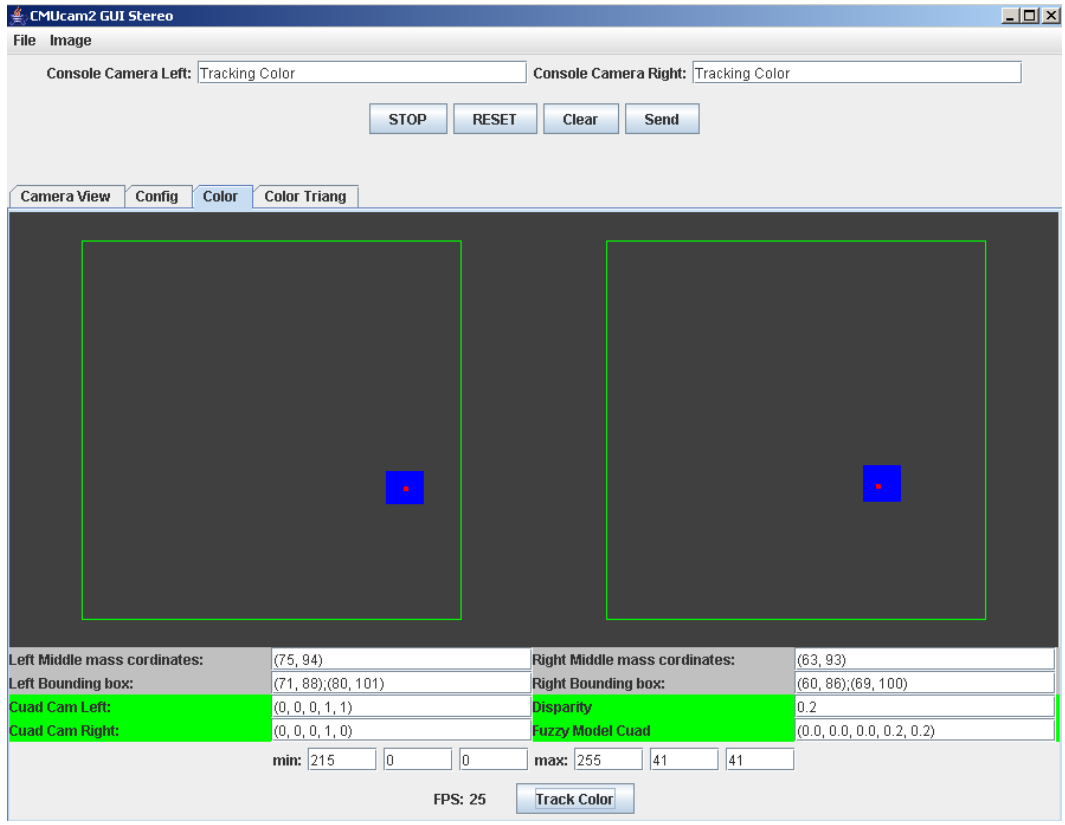
A.2.2 Panel de Configuración

Al igual que en la interfaz original, en este panel se pueden configurar todos los parámetros de las cámaras, excepto los relacionados a los servos. Cada opción se corresponde con un comando que puede ser enviado a las cámaras junto a su valor por línea de comandos.



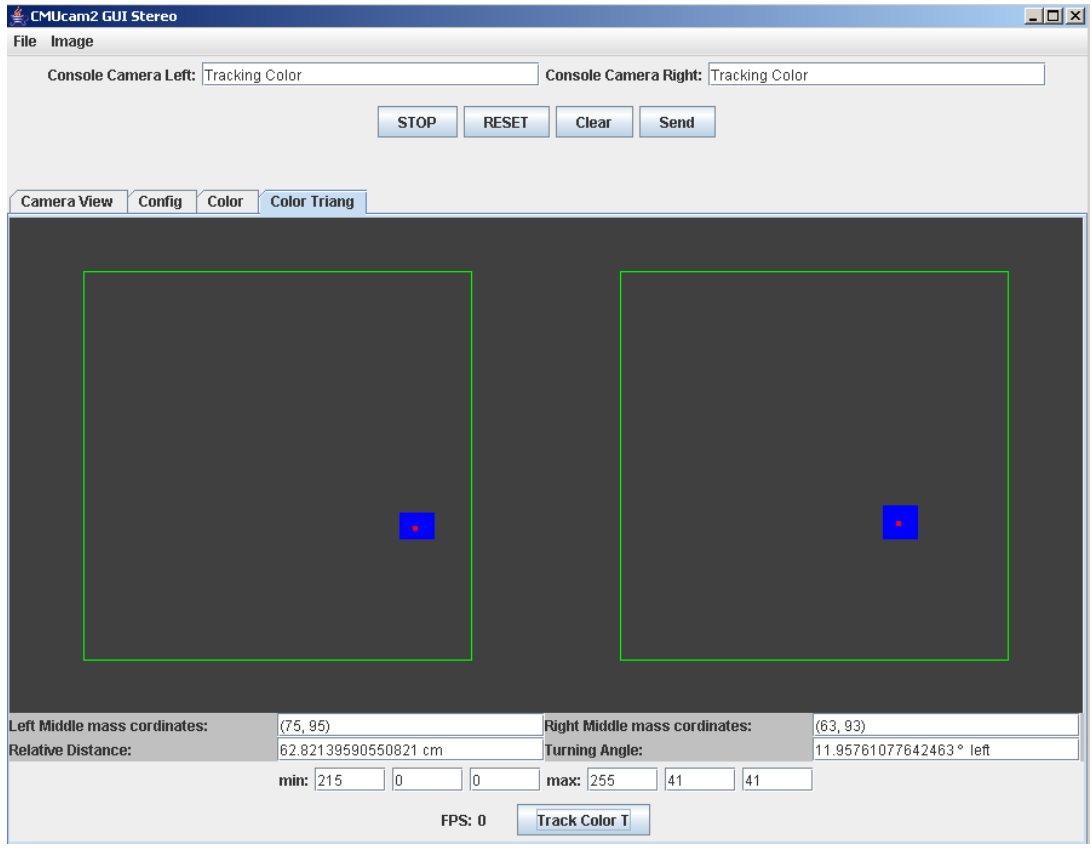
A.2.3 Panel de Color

En este panel se determinan los valores límites de color del área que se quiere detectar en ambas imágenes, y se ejecuta la función de ubicación de objetos por color. Además se muestra en la pantalla la ubicación del objeto demarcando sus bordes y su centro, que se precisan en el inferior del panel. También se muestran las activaciones de los cuadrantes de cada cámara, la disparidad y los valores de entrada al modelo difuso.



A.2.4 Panel de Color por Triangulación

En este panel se realizan las mismas acciones que en el anterior pero en el inferior del mismo se muestran la distancia relativa al objeto y el ángulo de giro con su dirección de desvío.



A.3 Parámetros de las CMUcam2

A.3.1 Tolerancia

El parámetro de Tolerancia indica el desvío que se utilizará para el cálculo de los límites superior e inferior para un valor de RGB de un píxel determinado que se tendrá en cuenta para realizar el rastreo por color.

Cuanto mayor es la tolerancia, más son los píxeles que se consideran como rastreados, por lo tanto el rectángulo que cubre al objeto es mayor. Además es fácilmente identificado siempre y cuando no haya en el ambiente otros objetos de tonos similares. Aumentan las posibilidades de que el rectángulo ocupe más cantidad de cuadrantes a medida que se acerca al objeto, y de que ambas cámaras reconozcan el objeto aunque sufran diferencias en las alteraciones por luminosidad.

Si la tolerancia es pequeña, es probable que a medida que se mueven las cámaras hacia el objeto, les sea imposible reconocerlo por la influencia de la luz externa que cambia a través del tiempo y del espacio recorrido.

A.3.2 Noise Filter

Este parámetro que puede tomar un valor de 0 a 255, indica cuántos píxeles activos consecutivos anteriores al píxel actual son requeridos antes de que el píxel deba ser detectado. Su valor por defecto es 2.

Cuanto mayor es el número de este parámetro, se toman en cuenta menos píxeles por lo tanto la zona del objeto buscado (el rectángulo azul) tiende a desaparecer. En cambio, cuanto menor es el valor, se tienen en cuenta más píxeles rastreados por lo que dicho rectángulo puede observarse sin dificultad.

Esto puede servir cuando en el ambiente hay varios objetos con valores similares al buscado, por lo que se reduce el ruido cuando se aumenta sensiblemente este valor, y se puede separar el objeto deseado de los que no lo son. Pero este valor no debería ser demasiado grande porque sino se perdería precisión, ni muy pequeño porque sino se considerarían todos los objetos de la escena que cumplan con la condición de color.

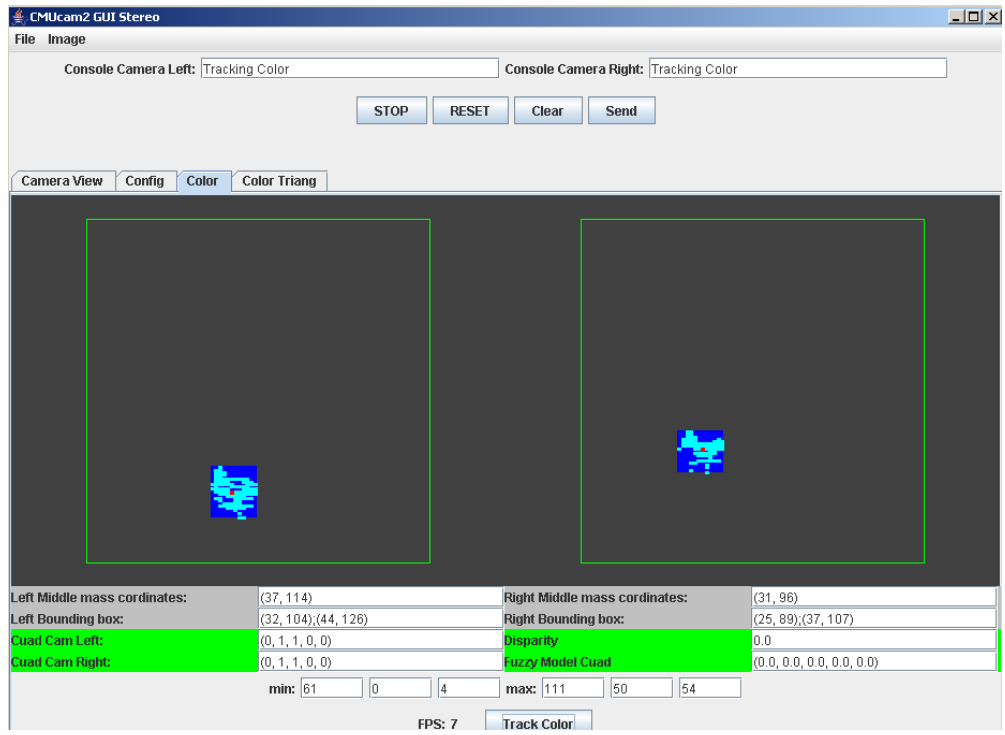
En las pruebas que se han desarrollado durante este trabajo, se ha tenido en cuenta un solo objetivo cuyo color sea diferenciable con el del resto del ambiente. Si existieran varios objetivos del mismo color en la escena sería conveniente usar

este filtro con un valor un poco mayor al que se propone por defecto pero no demasiado grande, no pudiendo especificar exactamente un número ya que dependería del color del objeto, del ambiente en general y de la luminosidad externa.

A.3.3 Line Mode 1: Bitmap of Tracked Region

El resultado del Color Track habiendo activado esta opción es un paquete igual al original pero al cual se le agregan bytes, cada uno de los cuales contiene el valor binario de los 8 píxeles que se están recibiendo desde la cámara desde la esquina superior izquierda a la esquina inferior derecha de la imagen. De esta manera, dentro del rectángulo azul que encierra el objeto, se verán ciertos puntos de color celeste que identifican los píxeles que efectivamente se encuentran dentro del rango del color establecido por el usuario para su rastreo. Es decir, se puede visualizar el contorno real del objeto además del mínimo rectángulo que lo cubre.

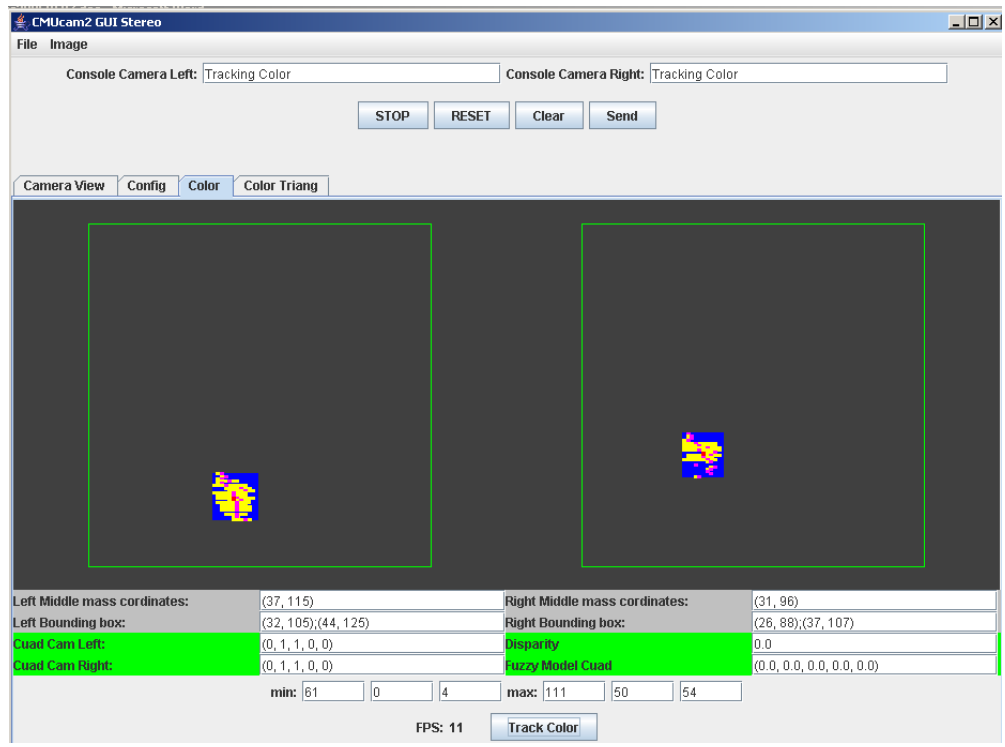
En cuanto al trabajo presentado, su funcionamiento es igual con esta opción activada, pero presenta la ventaja de que se puede observar mayor información.



A.3.4 Line Mode 2: Track Color Stats

Esta opción permite recibir en el mismo paquete resultado del comando Track Color, información adicional relacionada con estadísticas del grupo de píxeles de la zona rastreada. Particularmente, se muestra dentro del rectángulo azul que determina la zona del objeto buscado, líneas amarillas y puntos fucsias. El principio de las líneas amarillas indican el mínimo valor de píxel que cumple con la condición de pertenecer al rango de color definido por el usuario y el final indica el máximo valor de píxel que cumple dicha condición, para cada línea rastreada. El punto fucsia que se encuentra en línea determina el píxel cuya masa es la media de todos los puntos de ese segmento.

Al igual que en el caso anterior, esta opción activada tiene la ventaja de otorgar mayor información, aunque el funcionamiento del modelo presentado es el mismo.



A.3.5 Auto gain y White Balance

Auto gain es un comando interno que permite ajustar el nivel de brillo de la imagen para adaptarse al ambiente, es decir, intenta normalizar las partes claras y oscuras de la imagen.

White Balance pretende corregir los datos obtenidos a partir de los colores de la escena, ya que la luz del ambiente puede no ser blanca y la cámara vería los colores algo diferentes a los reales.

Ambas opciones permiten que la cámara regule la configuración correspondiente en forma automática cuadro a cuadro. Es decir, pueden activarse por un período para ajustar los valores y luego ser desactivadas.

En las pruebas realizadas no se han notado cambios en el comportamiento activando y desactivando estas opciones.

A.3.6 Inverted Tracking, Track Window e YcrCb

Estas opciones no tienen aplicación para este trabajo ya que, en el primer caso busca objetos fuera del rango determinado por el usuario (podría servir para la detección de obstáculos) y en el segundo caso busca objetos que tengan el color del centro de la imagen.

Color Track con RGB y con YCrCb funcionan de la misma manera, aún cambiando la tolerancia, según las experiencias realizadas.

Anexo B: Toolbox de Lógica Difusa

El toolbox de lógica difusa es una colección de funciones escritas en el entorno informático numérico MATLAB. Permite crear y editar sistemas de inferencia difusa utilizando MATLAB, o integrar sistemas difusos con el simulador Simulink y además, construir programas independientes en lenguaje C. El toolbox provee de una interfaz gráfica de usuario (GUI) que permite realizar el trabajo más fácilmente, pero también es posible trabajar completamente en línea de comandos.

El toolbox ofrece tres tipos de herramientas:

- Funciones en línea de comandos.
- Herramientas interactivas gráficas.
- Bloques de simulación Simulink y ejemplos.

La primer categoría está compuesta por funciones que pueden ser invocadas desde la línea de comandos o desde otras aplicaciones.

La segunda categoría proporciona la posibilidad de acceder a múltiples funciones a través de una interfaz gráfica de usuario utilizando herramientas interactivas.

La última categoría del toolbox consiste en un juego de bloques para ser utilizado con un software de simulación de MATLAB llamado Simulink

Además, es posible modificar y crear funciones propias que modifiquen el comportamiento del toolbox adaptándolo a cada problema.

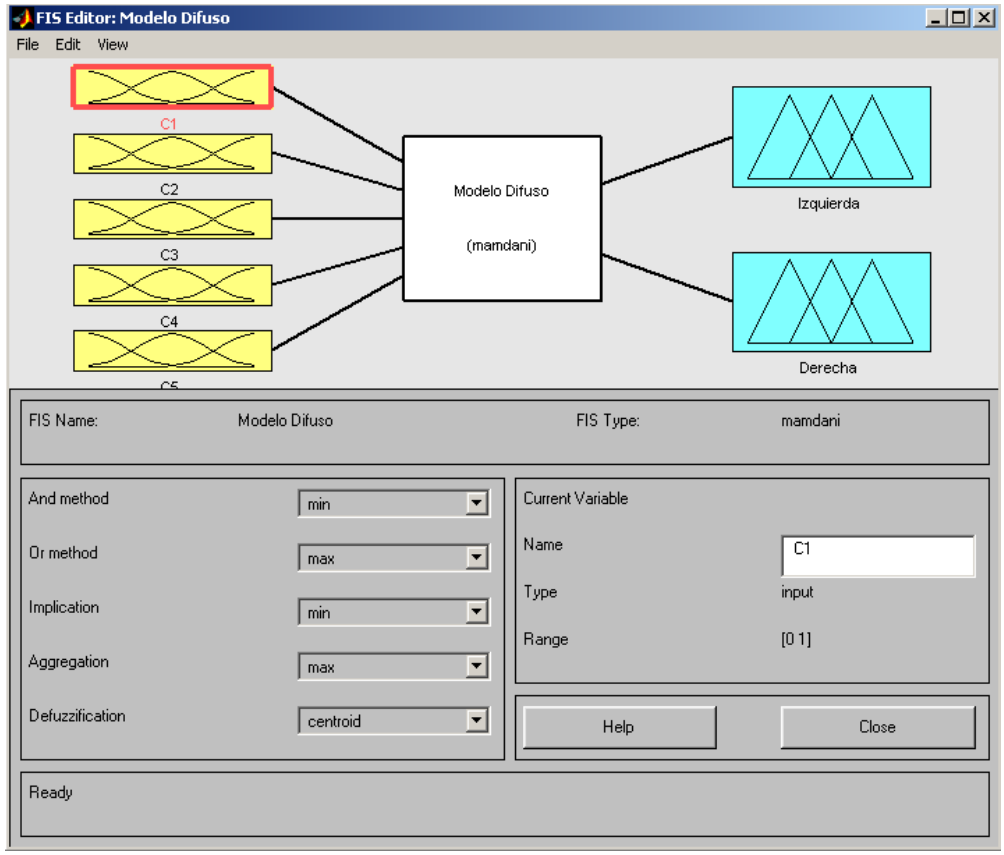
Para poder trabajar en un sistema de inferencia difuso a través de la interfaz gráfica de usuario (GUI), el toolbox ofrece cinco herramientas:

- Editor FIS (Fuzzy Inference System)
- Editor de Funciones Membresía
- Editor de Reglas Difusas
- Visor de Reglas Difusas
- Visor de Superficie

Estas herramientas están conectadas dinámicamente, de manera que los cambios que se producen a través de alguna de ellas, influye en las otras.

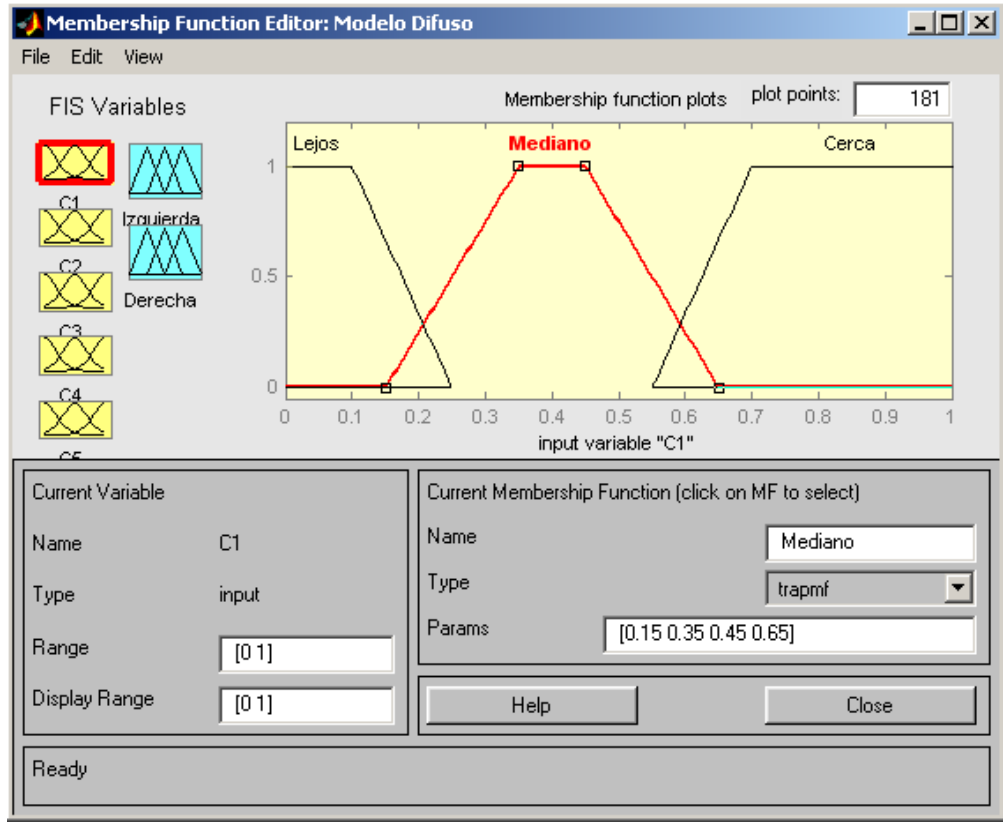
B.1 Editor FIS (Fuzzy Inference System)

El editor del sistema de inferencia difuso permite manejar las características generales o de más alto nivel del sistema. A través de un gráfico simple muestra la cantidad de variables de entrada y de salida, sus nombres o etiquetas lingüísticas, los métodos de inferencia, agregación y defusificación utilizados así como también los operadores difusos.



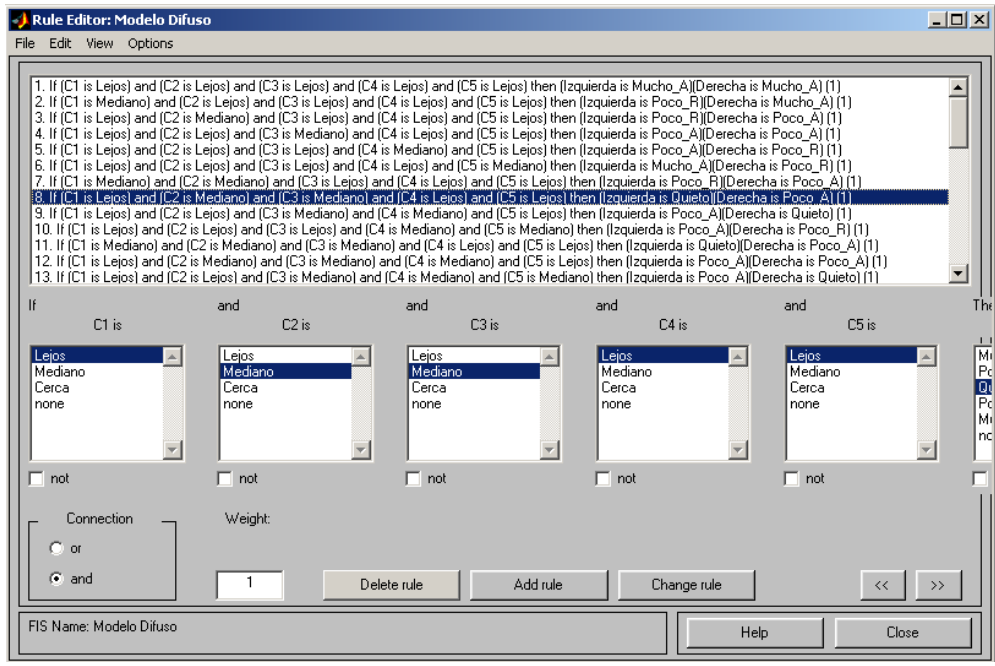
B.2 Editor de Funciones Membresía

El editor de las funciones membresía permite ver y editar todas las funciones de membresía, ya sea que correspondan a variables de entrada o de salida. Los valores a definir en este GUI se refieren al nombre, tipo y parámetros de cada función.



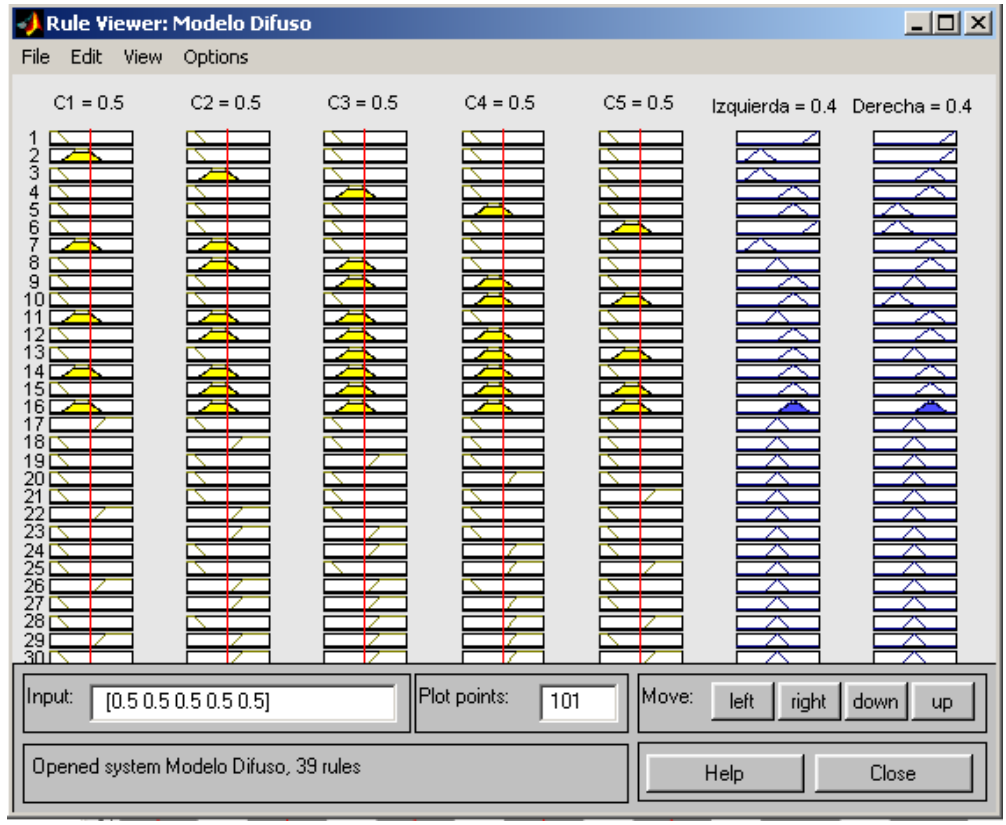
B.3 Editor de Reglas Difusas

Con base en la declaración de las variables de entrada y salida en el editor FIS, el editor de reglas difusas permite crear fácilmente las reglas a utilizar en el proceso de inferencia, seleccionando un valor para cada variable y un operador lógico de conexión. Pueden crease nuevas reglas, eliminarse reglas existentes o modificarlas.



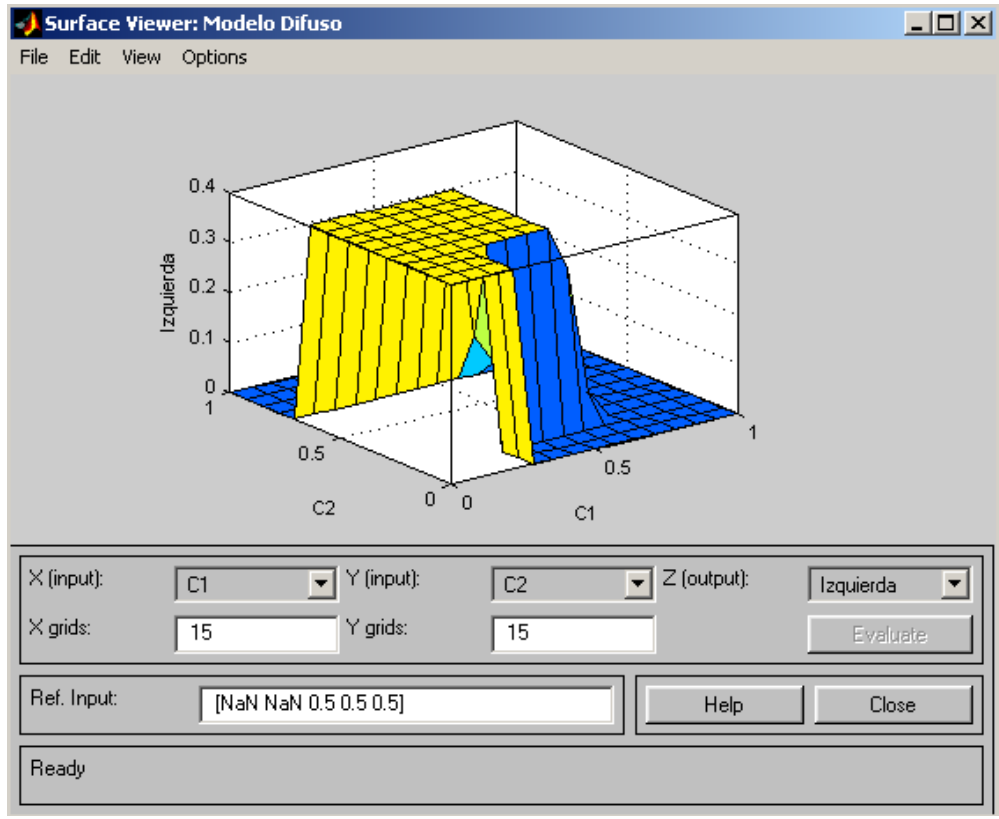
B.4 Visor de Reglas Difusas

El visor de reglas difusas muestra un mapa completo del proceso de inferencia difusa. En el campo "input" en el inferior de la pantalla se introducen los valores de las variables de entrada con lo cual se reajusta el sistema mostrando todo el cálculo del PID.



B.5 Visor de Superficie

El visor de superficie muestra la superficie de control de los Sistemas de Inferencia Difusos. Puede verse en forma tridimensional completa sistemas de dos entradas y una salida, pero si los sistemas tienen mayor cantidad de variables de entrada o salida la gráfica no puede representar la superficie de control total, solo mostrará la superficie para un par de variables de entrada y una de salida seleccionadas.



Bibliografía

[Pajares02] Gonzalo Pajares, Jesús M. De la Cruz, 2002, Visión por Computador, Imágenes digitales y aplicaciones, Dpto. de Arquitectura de computadores y Automática, Universidad Complutense de Madrid, 764 pp.

[Lopez] A. López, F. Pla, Visión estereoscópica basada en regiones mediante una técnica de minimización, Dept. Informática, Campus Rui Sec, Universitat Jaume I, 9 pp.

[Gonzalez93] Rafael C. Gonzalez y Richard E. Wood, 1993, Digital Image Processing, Addison Wesley, 716 pp.

[Marqués04] Luis Ángel Marqués Calvo, 2004, La técnica estereoscópica, www.estereoscopica.com

[Longstaff02] Damián Longstaff, 2002, Modelo Conexionista-Difuso para la navegación de agentes móviles, Universidad Nacional de Luján, 110 pp.

[AlcaláCasado] Fco. Javier Alcalá Casado, Introducción a la Visión Estéreo Calibrada, 30 pp.

[Kulkarni01] Arun D. Kulkarni, 2001, Computer Vision an Fuzzy-Neural Systems, 1/e, Prentice Hall, University of Texas, 518 pp.

[Romero00] Luis Alonso Romero, 2000, Diseño de Sistemas Borrosos, Dpto. de Informática y Automática, Universidad de Salamanca, 64 pp.

[MoralesLuna99] Guillermo Morales Luna, 1999, Elementos de Lógica Difusa, Centro de Investigación y Estudios Avanzados del IPN, 32 pp.

[Nauck98] Detlef Nauck, Frank Klawonn y Rudolf Kruse, 1998, Fuzzy Sets, Fuzzy Controllers and Neural Networks, Department of Computer Science, Technical University of Braunschweig, Germany, 22 pp.

[Zadeh65] Lotfi A. Zadeh, 1965, Fuzzy Sets, Information and Control, vol 8, 353 pp.

[Saffiotti01] A. Saffiotti y D. Driankov, 2001, Fuzzy Logic Techniques for Autonomous Vehicle Navigation, Physica-Verlag, Springer, 399 pp.

[Russell96] Stuart Russell, Peter Norvig, 1196, Inteligencia Artificial, Un

enfoque moderno, 1/e., Prentice may, 979 pp.

[Brooks86] Rodney A. Brooks, 1986, A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation, Vol RA-2, NO 1, 10 pp.

[Saffiotti97] Alessandro Saffiotti, 1997, Fuzzy Logic in Autonomous Robot Navigation – a case study -, Dept. of Technology, University of Örebro, Sweden

Bibliografía complementaria

[MathWorks01] MathWorks, Matlab, 2001, Fuzzy Logic Toolbox, User´s Guide Version 2, 295 pp.

[RoweVS03] Anthony Rowe y Carnegie Mellon University, 2003, CMUcam2 Vision Sensor, User Guide, 68 pp.

[RoweGUI03] Anthony Rowe y Carnegie Mellon University, 2003, CMUcam2 Graphical User Interface Overview, 13 pp.