

**Seguimiento de múltiples robots móviles mediante  
un sistema multicámara basado en una arquitectura  
pizarrón**



**T E S I S**

**Maestro en Ciencias**

especialidad en

**Sistemas Inteligentes**

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Por**

**Axel Castañeda González**

Mayo 2006

**Seguimiento de múltiples robots móviles mediante  
un sistema multicámara basado en una arquitectura  
pizarrón**

TESIS

**Maestro en Ciencias**

especialidad en

**Sistemas Inteligentes**

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Por**

**Axel Castañeda González**

Mayo 2006

# Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Monterrey

División de Tecnologías de Información y Electrónica

Programa de Graduados

Los miembros del comité de tesis recomendamos que la presente tesis de Axel Castañeda González sea aceptada como requisito parcial para obtener el grado académico de **Maestro en Ciencias** especialidad en:

**Sistemas Inteligentes**

**Comité de tesis:**

---

Dr. Santiago E. Conant P.

Asesor de la tesis

---

Dr. Rogelio Soto R.

Sinodal

---

Dra. Olivia M. Barrón C.

Sinodal

---

Dr. David A. Garza Salazar

Director del Programa de Graduados  
en Tecnologías de Información y  
Electrónica

Mayo de 2006

**Seguimiento de múltiples robots móviles mediante  
un sistema multicámara basado en una arquitectura  
pizarrón**

por

**Axel Castañeda González**

**Tesis**

Presentada al Programa de Graduados en Tecnologías de Información y Electrónica  
como requisito parcial para obtener el grado académico de

**Maestro en Ciencias**

especialidad en

**Sistemas Inteligentes**

**Instituto Tecnológico y de Estudios Superiores de Monterrey**

**Campus Monterrey**

Mayo de 2006

*A mi madre*  
*Adela González Castillo*

*A la memoria de mi abuela*  
*Saturnina Castillo Gil*

*A mi abuelo*  
*Antonio González Velázquez*

## Reconocimientos

Quiero externar mi más profundo agradecimiento a las personas que fueron parte fundamental para llegar a este instante dentro de mi vida.

A mi madre, por brindarme todo su ser, por creer en mí y por alentarme en cada decisión que he tomado a lo largo de mi vida.

A mis abuelos, por haberme cobijado con su cariño y enseñanzas.

A Itzel Janik, por su amistad y cariño en cada instante que hemos compartido.

A mi asesor Dr. Santiago E. Conant Pablos por haberme brindado su apoyo, orientación y sugerencias a lo largo del desarrollo de esta investigación.

A mis sinodales, Dra. Olivia M. Barrón y Dr. Rogelio Soto R. por sus observaciones y sugerencias en el transcurso de este proyecto de investigación.

AXEL CASTAÑEDA GONZÁLEZ

*Instituto Tecnológico y de Estudios Superiores de Monterrey*  
*Mayo 2006*

# Seguimiento de múltiples robots móviles mediante un sistema multicámara basado en una arquitectura pizarrón

Axel Castañeda González, M.C.  
Instituto Tecnológico y de Estudios Superiores de Monterrey, 2006

Asesor de la tesis: Dr. Santiago E. Conant P.

Esta tesis corresponde al programa de la Maestría en Ciencias con Especialidad en Sistemas Inteligentes del Programa de Graduados en Tecnología de Información y Comunicaciones.

En la actualidad, los Sistemas Multiagentes (*Multiple Agent Systems - MAS*) son motivo de diversos esfuerzos de investigación. A través de su implementación se pueden resolver problemas complejos mediante el funcionamiento en paralelo de entidades especializadas que perciben el mundo y lo modifican tratando de maximizar su medida de desempeño. Similarmente, el interés por realizar sistemas que puedan seguir múltiples objetos se ha venido manifestando en los últimos años en aplicaciones militares y de índole comercial.

En este proyecto de investigación se plantea implementar un enfoque multiagente para resolver el problema de coordinación de múltiples cámaras estáticas para el seguimiento de Múltiples Robots Móviles (*Wheel Mobile Robots - WMR*) que se desplazan a lo largo de los campos de visión de las cámaras. Se usará una arquitectura de comunicación entre agentes de tipo pizarrón en la que los agentes puedan contribuir con sus visiones parciales del ambiente para crear un ambiente global.

En este esquema cada agente tendrá la tarea de analizar cierta área de interés. Una vez hecho esto, el agente incorporará sus percepciones dentro del espacio de trabajo compartido pizarrón. En este espacio se identificarán las posiciones de los robots móviles a lo largo de su desplazamiento dentro de las regiones analizadas por los agentes. De esta forma, mediante el análisis de las percepciones plasmadas en el pizarrón, se formará una representación condensada del entorno analizado de forma segmentada.

# Índice general

<b>Reconocimientos</b>	<b>VI</b>
<b>Resumen</b>	<b>VII</b>
<b>Índice de cuadros</b>	<b>x</b>
<b>Índice de figuras</b>	<b>xI</b>
<b>Capítulo 1. Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	3
1.2. Hipótesis . . . . .	3
1.3. Contribuciones . . . . .	4
1.4. Resumen . . . . .	5
<b>Capítulo 2. Algoritmo de Seguimiento</b>	<b>6</b>
2.1. Introducción . . . . .	6
2.2. Localización del objetivo . . . . .	7
2.3. Análisis del movimiento . . . . .	8
2.3.1. Filtro de Kalman . . . . .	9
2.4. Algoritmo de Seguimiento <i>STT</i> utilizado en este trabajo. . . . .	12
2.5. Algoritmo de Seguimiento de Múltiples Objetos <i>MTT</i> . . . . .	18
2.5.1. Modelo <i>MTT</i> . . . . .	21
2.6. Resumen . . . . .	22
<b>Capítulo 3. Arquitectura Multiagente para el seguimiento de objetos</b>	<b>23</b>
3.1. Agentes inteligentes . . . . .	23
3.1.1. Tipos de Arquitecturas de Agentes . . . . .	24
3.1.2. Arquitectura de pizarrón . . . . .	25
3.2. Fusión de Datos en Sistemas Multisensor . . . . .	26
3.3. Arquitectura Multiagente propuesta . . . . .	28
3.3.1. Descripción . . . . .	28
3.4. Interacción entre Agente Cámara y Agente de Control . . . . .	31



3.5.	Fusión de datos en el sistema multiagente propuesto . . . . .	35
3.6.	Resumen . . . . .	37
<b>Capítulo 4. Implementación, experimentos y resultados</b>		<b>38</b>
4.1.	Caracterización del Filtro de Kalman . . . . .	38
4.2.	Correspondencia de coordenadas . . . . .	41
4.3.	Medición de error de posición . . . . .	43
4.4.	Experimentos . . . . .	43
4.4.1.	Una cámara, Agente Cámara 1 . . . . .	44
4.4.2.	Una cámara, Agente Cámara 2 . . . . .	46
4.4.3.	Dos cámaras, Agentes Cámara 1 y 2 . . . . .	47
4.5.	Experimentos de Fusión de datos mediante la arquitectura multiagente propuesta . . . . .	51
4.6.	Seguimiento intercámara . . . . .	59
4.7.	Resumen . . . . .	62
<b>Capítulo 5. Conclusiones y Trabajo Futuro</b>		<b>63</b>
5.1.	Los sistemas de seguimiento STT/MTT . . . . .	63
5.1.1.	Localización . . . . .	63
5.1.2.	Análisis de movimiento . . . . .	64
5.2.	Sistema Multiagente/Multicámara . . . . .	64
5.3.	Arquitectura de pizarrón para el seguimiento de objetos . . . . .	64
5.3.1.	Fusión de datos . . . . .	65
5.3.2.	Seguimiento Intercámara . . . . .	65
5.4.	Conclusiones Generales . . . . .	65
5.5.	Trabajo Futuro . . . . .	66
<b>Apéndice A. Calibración de las cámaras</b>		<b>67</b>
A.1.	Introducción . . . . .	67
A.2.	Modelo de cámara Tsai . . . . .	67
A.3.	Mapeo Inverso . . . . .	70
A.4.	Obtención de los parámetros intrínsecos y extrínsecos . . . . .	71
A.5.	Resumen . . . . .	75
<b>Apéndice B. Interfase computacional y archivos de configuración</b>		<b>76</b>
B.1.	Interfase computacional . . . . .	76
B.2.	Archivos de configuración . . . . .	78
<b>Bibliografía</b>		<b>79</b>
<b>Vita</b>		<b>82</b>

## Índice de cuadros

2.1. Ecuaciones de pronóstico del estado del Filtro de Kalman Discreto . . .	11
2.2. Ecuaciones de corrección del Filtro de Kalman Discreto . . . . .	12
4.1. Datos en una ventana de 100ms. . . . .	53
4.2. Fusión de datos para una ventana de 100ms. . . . .	54
4.3. Datos en una ventana de 150ms. . . . .	55
4.4. Fusión de datos en una ventana de 150ms. . . . .	56
4.5. Datos en una ventana de 200ms. . . . .	57
4.6. Fusión de datos para una ventana de 200ms. . . . .	58

## Índice de figuras

1.1. Arquitectura Multiagente. . . . .	4
2.1. <b>Ciclo del Filtro de Kalman Discreto.</b> La actualización del estado hace la proyección hacia adelante en el tiempo y la actualización de la medición ajusta el estimado proyectado con respecto a la medición en ese tiempo determinado. . . . .	11
2.2. Ciclo completo de funcionamiento del Filtro de Kalman Discreto. Se muestran las ecuaciones de predicción-corrección del Filtro de Kalman .	13
2.3. Algoritmo de Seguimiento. . . . .	14
2.4. Ejemplo del coeficiente de Bhattacharyya. . . . .	16
2.5. Vecindad de Objetos Candidatos. . . . .	18
2.6. Comparación de Objeto interés vs. Objetos candidatos. . . . .	19
2.7. <b>Elementos básicos de un sistema MTT convencional</b> [4]. . . . .	20
2.8. Algoritmo MTT. . . . .	22
3.1. Componentes de un <i>Sistema pizarrón</i> . . . . .	26
3.2. Sistema de seguimiento/fusión de múltiples sensores [4] . . . . .	27
3.3. Esquema ilustrativo del sistema. . . . .	28
3.4. Arquitectura del sistema multiagente: Agentes Cámara y Control. El primero tiene como objetivo primordial realizar el seguimiento de los objetos dentro de su área de interés, el segundo administra la información que se desea escribir en el <i>pizarrón</i> . . . . .	29
3.5. Estados del Agente Control. . . . .	30
3.6. Estados del Agente Cámara. . . . .	32
3.7. Registro de Agente Cámara. . . . .	33
3.8. Actualización/Solicitud de Información de Agente Cámara. . . . .	33
3.9. Ejemplo de intercambio de información. . . . .	34
3.10. Esquema de interacción entre los agentes. . . . .	35
3.11. Algoritmo de Fusión de Datos. . . . .	36
4.1. Sistema de Referencia de Agente Cámara 1. . . . .	42
4.2. Sistema de Referencia de Agente Cámara 2. . . . .	42

4.3. Se presentan dos percepciones que los agentes cámara tomaron del ambiente durante los experimentos realizados. . . . .	43
4.4. STT-Agente1. . . . .	44
4.5. MTT-Agente1. . . . .	45
4.6. STT-Agente 2. . . . .	46
4.7. MTT-Agente2. . . . .	47
4.8. Agente 1 y 2-STT. . . . .	48
4.9. Agente 1 y 2-MTT. . . . .	49
4.10. Agente 1 y 2-MTT. . . . .	50
4.11. Se muestra el proceso de seguimiento de un objeto por dos agentes cámara hasta los límites del área de seguimiento de uno de los dos agentes cámara . . . . .	59
4.12. Se muestra el proceso de seguimiento de un objeto de interés por dos agentes cámara hasta los límites del área de seguimiento de uno de los dos agentes cámara, a partir de ahí sólo el agente cámara B, continua siguiendo al objeto dentro de su trayectoria. . . . .	60
4.13. Se muestra el proceso de seguimiento de un objeto de interés por dos agentes. Un agente deja de seguir al objeto porque sale de su área de cobertura y retoma el seguimiento al buscarlo en una zona determinada indirectamente por el otro agente cámara. . . . .	61
A.1. Geometría de la cámara . . . . .	68
A.2. Transformación de coordenadas . . . . .	69
A.3. Toolbox de Calibración. . . . .	72
A.4. Ejemplo de un conjunto de imágenes de calibración. . . . .	72
A.5. Selección de esquinas de calibración. . . . .	73
A.6. Marco de referencia . . . . .	74
B.1. Mapa de Entorno. . . . .	76
B.2. Pizarrón. . . . .	77
B.3. Ventana de Percepción. . . . .	77
B.4. Archivo de configuración. . . . .	78

## Capítulo 1

# Introducción

La tecnología de Agentes, formalizada en el artículo de Shoham [32], es utilizada en la actualidad en un gran número de disciplinas: Interfases inteligentes de usuarios, comercio electrónico y sistemas de comunicación [18]. La visión computacional se ha valido de este paradigma debido a las ventajas que este esquema aporta, tales como la flexibilidad y la cooperación[18]. Los sistemas de análisis visual representan una aplicación ideal para los sistemas multiagente, debido a que las cámaras puedan ser modeladas como agentes independientes y realizar cooperación entre ellos con el objetivo de examinar mejor el entorno.

En el campo de la visión computacional, el seguimiento de objetos ha ido tomando mayor auge a medida que se desea ir automatizando diversos procesos en la industria. Los Sistemas de Seguimiento (*Tracking Systems*) han sido utilizados a lo largo del tiempo en diversas aplicaciones, tales como el monitoreo de camiones en aplicaciones de minería, el monitoreo de tráfico aéreo, y los sistemas de vigilancia. El uso de sistemas de vigilancia se ha incrementado en los últimos años a raíz de los hechos acontecidos el 11 de Septiembre del 2001, los cuales pusieron de manifiesto la necesidad de tener sistemas de vigilancia confiables que puedan cubrir extensiones grandes de monitoreo a través del uso de múltiples sensores y de esta forma se pueda reducir el riesgo de ataques terroristas [20].

Los sistemas de seguimiento se pueden dividir con respecto a tres características principales: conocimiento del tipo de objetos a seguir, movimiento de la cámara y rigidez de los objetos a seguir [13]. Entre los métodos clásicos de seguimiento destaca el Filtro de Kalman [19], el cual se convirtió en una herramienta básica de los sistemas de seguimiento desde su aparición en 1960. Este filtro ha sido ampliamente usado para realizar predicción de movimiento y de igual forma es utilizado para la fusión de datos en sistemas multisensor [39, p. 21]. Además, el incremento de los recursos computacionales a lo largo de los últimos años ha hecho factible la implementación de métodos más complejos para realizar el seguimiento de múltiples objetos [4]. Entre ellos se puede mencionar el filtro de partículas [1] (*Particle Filter*) y el algoritmo de desplazamiento medio [6] (*Mean-Shift*).

El seguimiento de objetos se puede realizar mediante sistemas monocámara y

multicámara. Un sistema monocámara es insuficiente para cubrir de forma eficiente una área determinada de interés de grandes dimensiones. Una alternativa para lograrlo es utilizar una cámara móvil o incluir un mayor número de cámaras estáticas para poder extender la región de cobertura. La desventaja de emplear una cámara móvil es que no puede analizar todos los puntos de su área de cobertura al mismo tiempo. Es por esto que el empleo de múltiples cámaras para realizar el seguimiento ha venido incrementándose en las últimas décadas, a medida que se abaratan los costos de estos dispositivos [8].

Los sistemas multicámara se han utilizado en diversas aplicaciones, entre ellas destaca su uso en sistemas de vigilancia (*Surveillance Systems*). Este tipo de sistemas se ha desarrollado bajo esquemas de traslape [3], no-traslape [31] y modelaje automático de correspondencia [35].

En base a las características inherentes del problema de seguimiento de múltiples objetos con múltiples cámaras estáticas, tales como el análisis de una gran cantidad de información y su naturaleza distribuida, presenta una arquitectura multiagente para implementar los mecanismos de cooperación e interacción entre las cámaras estáticas que estarán distribuidas para cubrir diferentes regiones de interés con lo cual se permitirá extender efectivamente la escena a analizar. Mediante este tipo de arquitectura se espera desarrollar un sistema con alto grado de autonomía y escalabilidad; de tal forma que se puedan incorporar nuevas cámaras al sistema sin afectar la capacidad de seguimiento de los objetos a lo largo de las escenas analizadas.

Los sistemas multiagente han demostrado su efectividad en diversos problemas que pueden abordarse desde enfoques distribuidos [10]. Características tales como autonomía, interacción, y cooperación son sumamente importantes para afrontar problemas complejos a través del trabajo en conjunto de agentes en un ambiente determinado. Las técnicas multiagente han servido para resolver una amplia variedad de problemas, donde han sido utilizados en la modelación e implementación de sistemas robóticos cooperativos y que recientemente se han incorporado en esquemas que involucran la coordinación distribuida de múltiples cámaras [21]. Los sistemas multiagente se han empleado en diferentes tareas dentro de algunos modelos de sistemas multicámara, entre algunos esfuerzos de investigación por incorporarlos en estos sistemas podemos destacar: la coordinación de cámaras activas para el seguimiento de múltiples objetos en tiempo real [36], coordinación de cámaras en sistemas de vigilancia [17], comunicación y coordinación de cámaras en sistemas de vigilancia de tráfico [2], la fusión de datos mediante diversos sensores modelados como agentes para la cooperación de robots móviles [9], y la coordinación de sociedades de cámaras inalámbricas móviles [22].

El paradigma multiagente ha sido utilizado en el desarrollo de sistemas de vigilancia como en Orwell[18] y Remagnino [24]. El uso de este esquema en visión computacional se ha incrementado en la última década [25, 34], debido a que el paradigma multiagente tiende en sí a un sistema de inteligencia distribuida capaz de permitir la in-

interpretación automática de escenas dinámicas con las técnicas de visión computacional [24].

## 1.1. Objetivos

El objetivo fundamental de este trabajo es implementar un sistema basado en una arquitectura multiagente tipo pizarrón que lleve a cabo el seguimiento en tiempo real de múltiples robots móviles (*MTT*) mediante un sistema constituido por múltiples cámaras estáticas (pasivas). El esquema considera a las cámaras estáticas como los sensores de los agentes, los cuales mediante mecanismos de cooperación e interacción extenderán efectivamente el área de cobertura de sensado.

De forma particular, la arquitectura que se propone para integrar las capacidades de múltiples cámaras tiene el objetivo de poder realizar el seguimiento de robots móviles a lo largo de trayectorias arbitrarias en ambientes donde una sola cámara es incapaz de cubrir el desplazamiento de todos los robots en el entorno deseado. En la figura 1.1 se muestra un esquema de los elementos de la arquitectura propuesta.

Los objetivos específicos de esta tesis son los siguientes:

- Seguir a múltiples WMRs. Se extenderá el método del filtro de Kalman para realizar el seguimiento de múltiples robots móviles, los cuales serán considerados como objetos rígidos con marcas de color en su parte superior.
- Establecer una arquitectura multiagente que permita utilizar múltiples cámaras. Se coordinará a las cámaras modeladas como agentes en su intercambio de información a través de un esquema de pizarrón.
- Definir un modelo de correspondencia que integre las vistas parciales de las cámaras. Se establecerá la forma en que se va a integrar la información de las cámaras en el pizarrón donde se plasmará el análisis de cada agente.
- Realizar fusión de datos cuando se tenga redundancia de información de un WMR. La fusión de datos se realizará en los traslapes de las vistas de los agentes para tener una mayor certeza de la posición de los robots móviles en esas áreas específicas de análisis.

## 1.2. Hipótesis

Se desea demostrar que un sistema basado en una arquitectura multiagente de tipo pizarrón es una alternativa funcional y eficaz para realizar la coordinación y cooperación

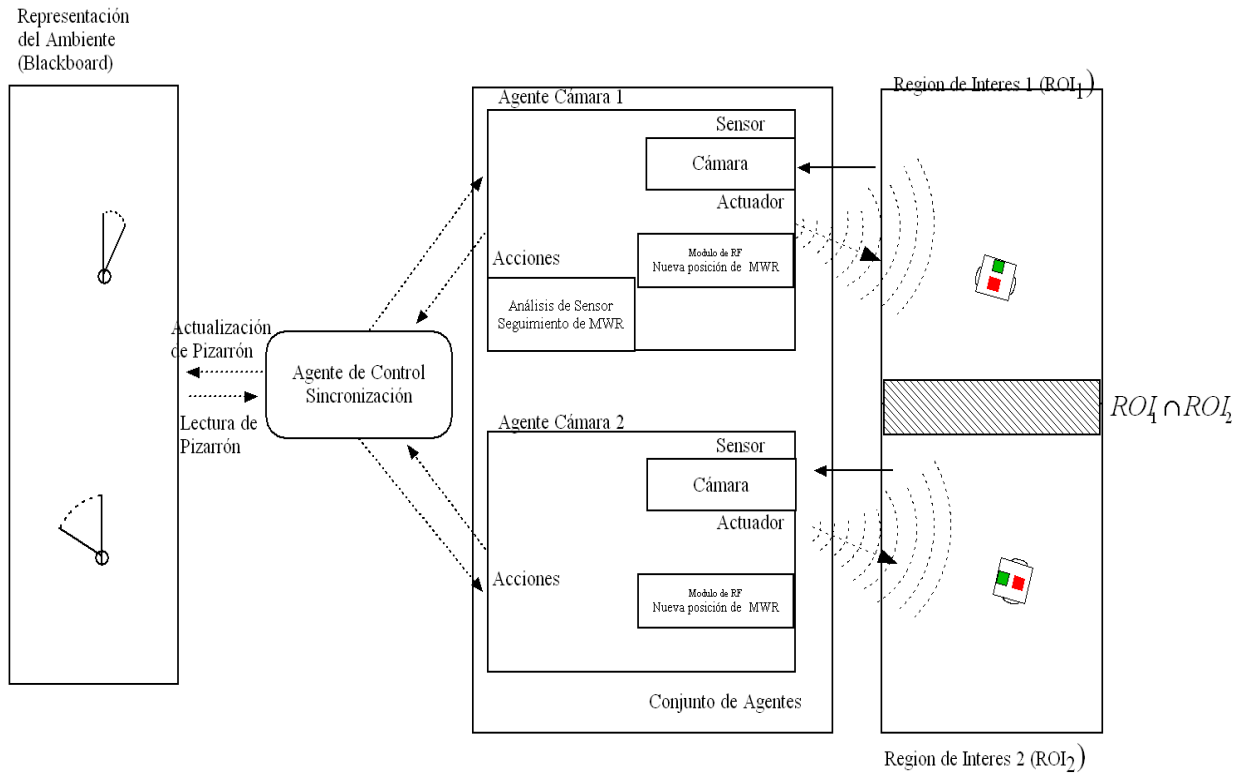


Figura 1.1: Arquitectura Multiagente.

Arquitectura utilizada en esta investigación para realizar seguimiento de múltiples robots móviles mediante múltiples cámaras.

de varias cámaras estáticas en una implementación de seguimiento de múltiples robots móviles en tiempo real.

Algunas preguntas que se esperan resolver durante el transcurso de la investigación son las siguientes:

- ¿Cómo incorporar un esquema de fusión de datos para darle mayor robustez al sistema de seguimiento?
- ¿Cómo incorporar nuevas cámaras para extender la cobertura de sensado?
- ¿Cómo administrar un sistema tipo pizarrón para apoyar la coordinación y cooperación de múltiples agentes de sensado?

### 1.3. Contribuciones

El desarrollo de este proyecto de investigación sienta las bases para trabajos futuros donde se desee realizar el seguimiento de objetos mediante un sistema multicámara,



la arquitectura propuesta permite incorporar nuevos seguidores que permitan la implementación de otras técnicas de análisis de movimiento. Específicamente, las principales aportaciones de este trabajo de investigación son:

- Sistema de seguimiento. Se diseñó, desarrolló y experimentó un esquema de seguimiento de un sólo robot (STT) y de múltiples (MTT) robots móviles mediante la caracterización y empleo de uno y de múltiples filtros de Kalman, respectivamente.
- Sistema multicámara. Se presentó un esquema multiagente para el seguimiento de uno y de múltiples robots a través de una arquitectura de pizarrón, donde los robots pueden ser seguidos a través de las áreas combinadas de las cámaras.
- Fusión de datos. Se presentó un esquema de fusión de datos que realiza la combinación de estimaciones en zonas donde existe traslape de las áreas de interés de los agentes cámara.

## 1.4. Resumen

En este capítulo introductorio se presentaron los objetivos, la hipótesis y las preguntas de investigación que se desean concretar en este trabajo de tesis. En el Capítulo 2, ofrece un panorama general de los sistemas de seguimiento tanto de uno (*STT*, *Single Target Tracking*) como de múltiples objetos (*MTT*, *Multiple Target Tracking*), de este modo se muestran los algoritmos de seguimiento utilizados en esta investigación que implementan el seguimiento en base a estos esquemas. Primero se presenta los sistemas de *STT*, para posteriormente extender este enfoque a los sistemas de seguimiento de múltiples objetos. En el Capítulo 3 se describirán brevemente las diferentes arquitecturas multiagente y se mostrará con mayor detalle la arquitectura de pizarrón, la cual se empleará como base para desarrollar nuestro sistema multicámara. En el Capítulo 4 se caracteriza el filtro de Kalman empleado en esta investigación como herramienta para realizar el análisis del movimiento de los robots, así como el esquema de interacción entre los agentes y los experimentos realizados al seguir a uno y dos robots con uno y dos agentes. En el Capítulo 5 se muestran las conclusiones obtenidas en esta investigación y trabajo futuro sobre este mismo proyecto. El apéndice A ofrece una descripción del proceso de calibración de las cámaras empleadas por los agentes cámara y finalmente en el apéndice B, se presenta una descripción de la interfase computacional y de los archivos de configuración que el sistema desarrollado necesita para su funcionamiento.

## Capítulo 2

# Algoritmo de Seguimiento

En este capítulo se describe el esquema de seguimiento utilizado. Primero se presenta el algoritmo para el seguimiento de un sólo objeto (*STT*), posteriormente se extiende este esquema hacia un sistema de seguimiento de múltiples objetos (*MTT*). En el capítulo siguiente se muestra cómo un esquema que utiliza un enfoque multi-agente permite ampliar aún más la capacidad de seguimiento, al examinar una área de mayor dimensión mediante el uso de dos cámaras, las cuales analizan dos áreas de interés con cierto traslape.

## 2.1. Introducción

El seguimiento de objetos se refiere al trazado del progreso de un objeto a lo largo de su desenvolvimiento dentro de una escena visual. Se refiere al problema de seguir a objetivos en movimiento a través de una secuencia de imágenes.

Hafed [13] ilustra tres características que describen a los algoritmos de seguimiento de objetos:

- *Conocimiento disponible.*

Algunos algoritmos requieren de la incorporación de conocimiento previo con respecto al tipo de objetos que se desea seguir. Por ejemplo, se han desarrollado sistemas en los que el seguimiento de rostros lo ajustan a elipses de distintos tamaños a lo largo de cada cuadro de video. Si la finalidad es seguir rostros, el conocimiento previo es que los objetos que se seguirán son sólo de este tipo, con lo cual, se implementa un método especializado donde se trata de maximizar el desempeño para ese problema en particular; esa es la ventaja de conocer de forma previa el tipo entidades que se analizarán. Sin embargo, existen otras técnicas donde no se requiere tal información. No obstante, las ventajas o desventajas dependen del tipo de dominio del problema que se desea resolver. En este proyecto de investigación los robots móviles serán identificados mediante marcas de color, lo cual permitirá identificarlos y seguirlos a lo largo de su desplazamiento.

- *Movimiento de Cámara.* La movilidad de la cámara influye en los esquemas para afrontar el seguimiento de objetos, debido a que si la cámara no se mueve el objetivo es trazar las ubicaciones de los objetos dentro de un campo visual estacionario; en cambio, si la cámara se mueve la meta es típica consiste en centrar el objeto que se está siguiendo en la imagen. En el caso particular de este proyecto de investigación, se utiliza un conjunto de cámaras estáticas.
- *Cuerpos Rígidos vs NoRígidos.*

La capacidad de los algoritmos para seguir cuerpos no rígidos es importante para su incorporación en aplicaciones de la vida real. Sin embargo, hay veces que no es necesario seguir objetos no rígidos; por ejemplo, en una línea de producción donde los robots sólo manipulan piezas de ensamblaje perfectamente bien definidas, el sistema de seguimiento visual se desempeñaría con objetos rígidos. La determinación del algoritmo a utilizar depende de cada problema particular. En este proyecto, los objetos con los cuales se trabaja son rígidos.

Los sistemas de seguimiento se enfrentan a dos problemas fundamentales: el movimiento y la localización del objetivo.

## 2.2. Localización del objetivo

Consiste en identificar un elemento de la imagen en una área de búsqueda determinada. Encontrar correspondencias (cuadro a cuadro).

La localización sobre el tiempo típicamente envuelve la comparación de objetos en imágenes consecutivas usando atributos tales como puntos y líneas[37].

Hu *et al.* [37], presenta una clasificación de los métodos de localización divididos en tres categorías principales: seguimiento basado en regiones, seguimiento basado en contornos activos, seguimiento basado en atributos.

- *Seguimiento basado en regiones.*

Estos algoritmos realizan el seguimiento de los objetos de acuerdo a las variaciones de las regiones de las imágenes correspondientes a los objetos en movimiento. Para estos algoritmos, el fondo de la imagen es mantenido dinámicamente y las regiones en movimiento son usualmente sustraída del fondo de la imagen actual [37].

- *Seguimiento basado en contornos activos.*

Siguen a los objetos representando el perímetro de estos como límites de contorno y actualizando estos contornos dinámicamente en cuadros consecutivos . Estos algoritmos extraen directamente las formas de los objetos y proveen descriptores más efectivos de los objetos que los algoritmos basados en regiones [37].

- *Seguimiento basado en atributos.*

Estos algoritmos realizan el reconocimiento y el seguimiento de los objetos por medio de la extracción de elementos dentro de la imagen condensándolos como atributos de más alto nivel y comparándolos con atributos entre imágenes. Estos algoritmos pueden ser clasificados tres subcategorías [37]:

- Algoritmos basados en atributos globales. Incluyen centroides, perímetros, áreas y colores.
- Algoritmos basados en atributos locales. Incluyen segmentos de líneas y vértices de esquinas.
- Algoritmos basados en dependencias gráficas. Incluyen una gran variedad de distancias y relaciones geométricas entre atributos.

Existen diversas clasificaciones de los algoritmos de localización debido a la variedad de criterios que pueden ser utilizados para agruparlos.

Las características que describen al algoritmo básico de seguimiento que se utiliza en este trabajo, así como clasificación del método de localización seleccionado se muestran a continuación:

- Basado en atributos globales, el algoritmo de seguimiento se vale de marcas de color en los robots para poder realizar comparaciones entre las secuencias de las imágenes.
- Utiliza conocimiento previo. Se utiliza un conjunto de información de color para cada objeto de interés(*OI*) que se desea seguir. Este conocimiento se tiene de antemano para poder encontrar y seguir a los objetos.
- Se utilizan cámaras estáticas.
- Cuerpos rígidos. Se sabe de antemano que los cuerpos que se están siguiendo son rígidos.

## 2.3. Análisis del movimiento

El análisis del movimiento, comúnmente denominado *análisis dinámico de imágenes*, se realiza al buscar correspondencias entre pares de puntos de interés en las imágenes secuenciales [26].

El análisis del movimiento consiste en predecir la localización de un elemento en una imagen del objeto que se sigue en la imagen mediante el uso de posiciones previas. De esta forma se identifica una región de búsqueda en la imagen en la que el elemento se espera encontrar con una alta probabilidad.

El movimiento del  $OI$  es considerado continuo dentro de una escena examinada. Para esto, el intervalo de muestreo debe ser suficientemente pequeño, para suponer que el movimiento de características entre dos imágenes de una secuencia tiene un comportamiento de tipo lineal [26].

Se han desarrollado herramientas matemáticas muy útiles para realizar el análisis del movimiento, en las que se incluyen el filtro de Kalman y el algoritmo de condensación, entre otros. En las siguientes subsecciones se describirá el Filtro de Kalman, el cual se utilizará como herramienta básica para realizar el análisis de movimiento en nuestro sistema debido a su simplicidad y efectividad.

### 2.3.1. Filtro de Kalman

Una herramienta importante en el análisis del movimiento es el filtro de Kalman. Hafed [13] remarca la importancia de este filtro, debido a que involucra fusión recursiva de mediciones ruidosas para producir un estimado mejorado del estado del sistema analizado. El trabajo de Kalman ha tenido un impacto dramático en el área de seguimiento de objetos. A mediados de los 60s, el filtro de Kalman se convirtió en una metodología estándar para el seguimiento de sistemas de un solo elemento. Este filtro también es parte esencial en aplicaciones que requieren el seguimiento de múltiples objetos.

Para problemas particulares, si las consideraciones que asume el filtro de Kalman son acertadas, entonces, ningún otro algoritmo puede superarlo. Sin embargo, en problemas reales no siempre se acierta en estas consideraciones, por lo que se deben implementar métodos aproximados [1].

En 1960, Rudolph E. Kalman publicó su famoso artículo “*A new approach to linear filtering and prediction problems*”, en donde se describe una herramienta matemática que permite estimar estados de un sistema lineal. El criterio que él propuso es minimizar el vector de estados de la norma de covarianza, mediante el uso de la recursión clásica [29].

El filtro de Kalman es esencialmente un conjunto de ecuaciones matemáticas que implementan un estimador de tipo predicción-corrección, el cual es óptimo en el sentido de minimizar la covarianza estimada del error cuando ciertas condiciones son satisfechas [38]. Este filtro es muy poderoso por diversas razones: soporta estimaciones de estados pasados, presentes e incluso futuros, y lo hace aún cuando la naturaleza precisa del modelo es desconocido [39].

#### **Funcionamiento:**

Dan Simon hace énfasis en el empleo del filtro de Kalman en procesos que pueden ser descritos como sistemas lineales. Muchos procesos físicos, tales como el manejo de

un vehículo a lo largo de una carretera, un satélite orbitando la tierra, o una señal de carga sinusoidal de frecuencia de radio pueden ser aproximados en este tipo de sistemas [33].

Welch y Bishop presentan una descripción detallada del funcionamiento del Filtro de Kalman en [38], algunos puntos importantes necesarios para comprender este *estimador óptimo recursivo* se muestran a continuación:

El algoritmo trata de estimar el estado  $x \in \mathfrak{R}^n$  de un proceso controlado en tiempo discreto, el cual es definido por una ecuación lineal estocástica de diferencias.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (2.1)$$

con una medición  $z \in \mathfrak{R}^m$

$$z_k = Hx_k + v_k \quad (2.2)$$

donde las variables aleatorias  $w_k$  y  $v_k$  representan la incertidumbre asociada al proceso y la medición, respectivamente. Se asume que estas variables son independientes entre sí, blancas y con distribuciones de probabilidad normales, como se denota en las ecuaciones 2.3 y 2.4.

$$p(w) \sim N(0, Q) \quad (2.3)$$

$$p(v) \sim N(0, R) \quad (2.4)$$

La matriz  $A$  de dimensiones  $n \times n$  en la ecuación de diferencias (2.1), relaciona el estado anterior  $k - 1$  con el actual. La matriz  $B$  de dimensiones  $n \times l$  relaciona una entrada de control opcional al modelo del sistema  $u \in \mathfrak{R}^l$  para el estado  $x$ . La matriz  $H$  en la ecuación (2.2) asocia el estado del modelo con la medición  $z_k$ .

**Algoritmo del Filtro de Kalman.** La idea básica del Filtro de Kalman es estimar un proceso utilizando la forma de control con retroalimentación; es decir, el filtro estima el estado del proceso en un determinado tiempo y después obtiene una retroalimentación en forma de mediciones con cierto grado de incertidumbre (ruido). De tal forma que el filtro se puede dividir en dos etapas principales: *actualización del estado (predicción del estado)* y *actualización de la medición (corrección)*. La actualización del estado incluye las ecuaciones correspondientes para proyectar hacia adelante (en el tiempo) el estado actual y la estimación de la covarianza de error. Por otra parte, las actualizaciones de la medición son responsables por la retroalimentación – incorporando una nueva medición en el estimado *a priori* para obtener el estimado *a posteriori*

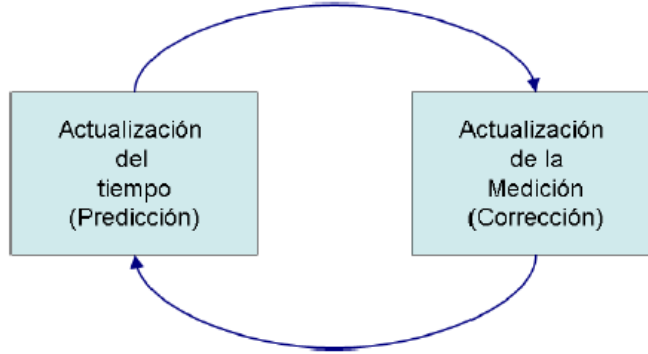


Figura 2.1: **Ciclo del Filtro de Kalman Discreto.** La actualización del estado hace la proyección hacia adelante en el tiempo y la actualización de la medición ajusta el estimado proyectado con respecto a la medición en ese tiempo determinado.

mejorado. De tal forma que el ciclo de funcionamiento del FK trabaja como el proceso de predicción-corrección. Ilustrado en la figura 2.1.

Las ecuaciones que modelan el proceso de pronóstico-corrección se muestran en los cuadros 2.1 y 2.2.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (2.5)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (2.6)$$

---

Cuadro 2.1: Ecuaciones de pronóstico del estado del Filtro de Kalman Discreto

En el Cuadro 2.1 se presentan las ecuaciones de proyección hacia adelante de los estimados del estado y la covarianza del error de un tiempo  $k - 1$  a uno  $k$ . La matriz  $A$  relaciona el estado en un tiempo  $k - 1$  con el estado actual en un tiempo  $k$  y  $Q$  representa la covarianza de la perturbación aleatoria del proceso que trata de estimar el estado.

La primer tarea, que se muestra en la Ecuación 2.7, durante la corrección de la proyección del estado es el cálculo de la ganancia de Kalman ( $K_k$ ). Posteriormente se realiza una medición del proceso para obtener  $z_k$ , y se obtiene un estimado de estado *a posteriori* mediante la Ecuación 2.8. Finalmente, se calcula un estimado de la covarianza del error *a posteriori* a través de la Ecuación 2.9.

Después de cada par de actualizaciones, tanto del tiempo como de la medida, el proceso es repetido tomando como punto de partida las nuevas estimaciones del estado y de la covarianza del error [27]. En la Figura 2.2 se muestra en conjunto como interactúan las ecuaciones de estimación y corrección.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (2.7)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-) \quad (2.8)$$

$$P_k = (I - K_k H) P_k^- \quad (2.9)$$

Cuadro 2.2: Ecuaciones de corrección del Filtro de Kalman Discreto

### Ventajas:

Evita la influencia de posibles cambios estructurales en la estimación. El filtro de Kalman utiliza toda la historia de la serie de datos, pero intenta estimar una trayectoria estocástica de los coeficientes en lugar de una determinística, con lo cual soluciona el posible sesgo de la estimación ante la presencia de cambios estructurales [27].

El filtro se distingue por su habilidad para predecir el estado de un modelo en el pasado, presente y futuro, aún cuando la naturaleza precisa del sistema modelado es desconocida. La modelación dinámica de un sistema es una de las características claves que distingue el método de Kalman. Los modelos lineales dinámicos son modelos con una transición lineal desde un periodo al próximo, los cuales pueden describir la mayoría de los modelos comúnmente utilizados en trabajos de series de tiempo [27].

## 2.4. Algoritmo de Seguimiento *STT* utilizado en este trabajo.

El algoritmo de seguimiento de un objeto utiliza el Filtro de Kalman como predictor de posición, el cual le permite adelantarse al movimiento del *OI* a lo largo de una área determinada de interés. Se escogió este predictor por su probada eficacia para el seguimiento de objetos desde su aparición, debido a esto se consideró que es un muy buen punto de partida para realizar el seguimiento de uno y varios objetos con una cámara y con varias cámaras mediante la arquitectura multiagente.

Este algoritmo de forma general define un objeto de interés (estos son los robots móviles que se desplazan en el ambiente) con una distribución  $\hat{\mathbf{q}}$ , el cual cuenta con un estado  $x_k$  en el tiempo  $t_k$ , el estado futuro ( $x_{k+1}$ ) es estimado por el filtro de Kalman. Una vez estimada su posición siguiente se realiza la medición y una comparación con objetos candidatos dentro de una vecindad 8, es decir, con ocho posibles objetos candidatos más el objeto candidato de la predicción. Esto es similar a lo realizado por el algoritmo de desplazamiento medio [6], sin embargo se optó por limitar la comparación



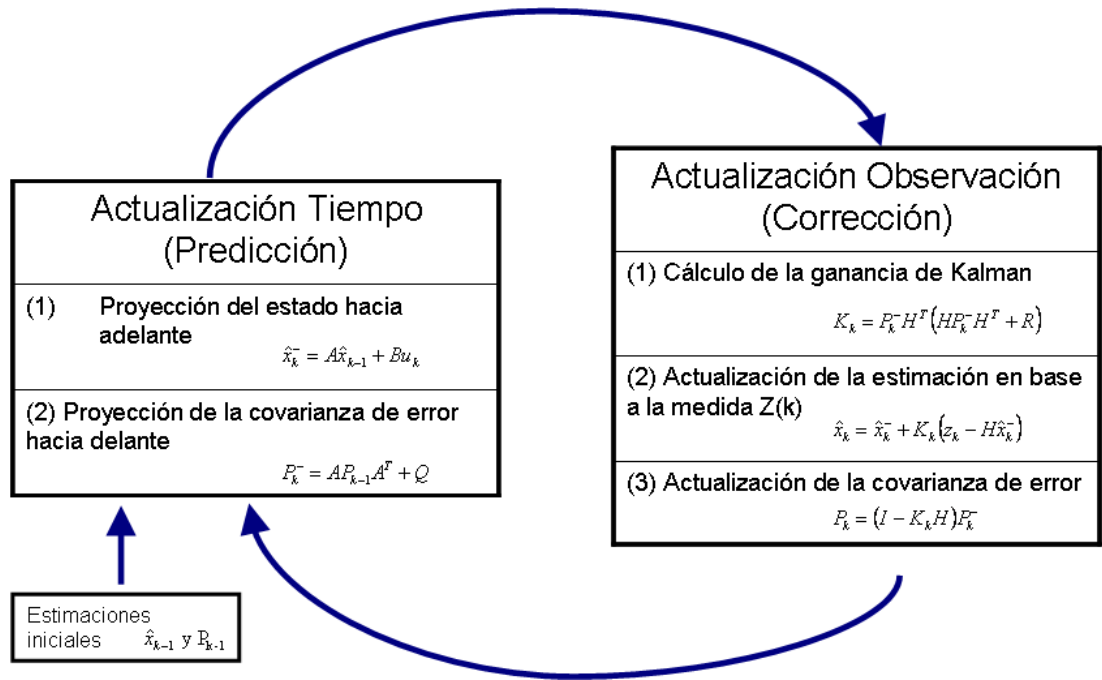


Figura 2.2: Ciclo completo de funcionamiento del Filtro de Kalman Discreto. Se muestran las ecuaciones de predicción-corrección del Filtro de Kalman

de objetos candidatos a sólo aquellos que se encuentran dentro de una vecindad definida con la finalidad de disminuir el número de objetos candidatos evaluados y simplificar la selección de los mismos. Un factor importante necesario para darle funcionalidad al esquema de seguimiento mencionado es contar con un método de discernimiento adecuado que permita identificar con una alta confiabilidad entre distribuciones de objetos de interés diferentes. La medición que retroalimenta al filtro de Kalman es aquella que maximiza el coeficiente de similitud de Bhattacharyya al comparar a los objetos candidatos con el objeto de interés. El algoritmo se muestra en la figura 2.3.

### Objeto de Interés

El objeto de interés ( $OI$ ) está definido por la estimación de densidad  $\hat{q}_u$  calculada por la ecuación 2.10.

$$\hat{q}_u(y) = \frac{1}{(2h)^2} \sum_{i=-h}^h \sum_{j=-h}^h \delta [b(\mathbf{y} + \mathbf{x}_{ij}) - u] \quad (2.10)$$

donde:

- El subíndice  $u$  indica que la distribución  $\hat{q}_u$  se obtiene con base en características de color  $u=1\dots m$ ,  $m$  es el número de colores en la imagen

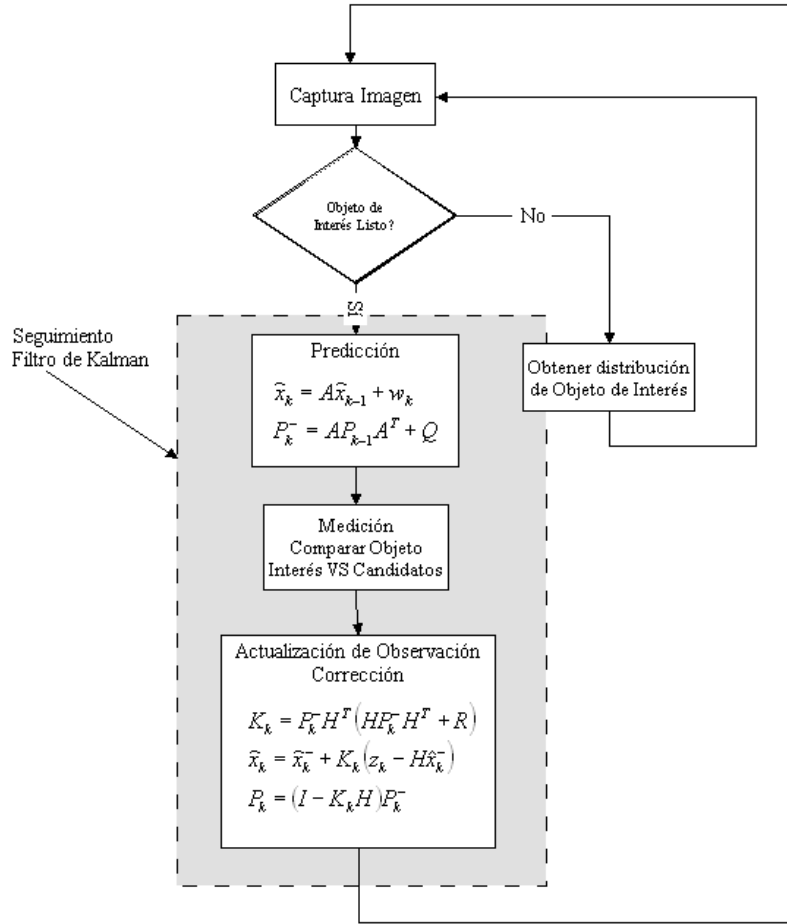


Figura 2.3: Algoritmo de Seguimiento.

Se muestran los pasos que el algoritmo de seguimiento realiza sobre las observaciones

- $h$  es el radio de la ventana de seguimiento.
- $(2h)^2$  es el número total de píxeles en la ventana de interés.
- $\mathbf{x}_{ij} = (x_{ij}, y_{ij})$  son las posiciones de los píxeles que conforman la ventana tomando como origen el centro de la región de interés.
- $\mathbf{y}$  es la posición donde se localizó al  $OI$ .
- $b(\mathbf{y} + \mathbf{x}_{ij})$  es una función que asigna a cada píxel el índice de la distribución correspondiente al color en dicho píxel en el modelo de color  $L_1L_2L_3$ .
- $\delta$  es la función delta de Kronecker definida por:

$$\delta = \begin{cases} 1 & \text{si } b(\mathbf{y} + \mathbf{x}_{ij}) = u \\ 0 & \text{si } b(\mathbf{y} + \mathbf{x}_{ij}) \neq u \end{cases} \quad (2.11)$$

En esta investigación se utiliza un histograma como función de densidad no paramétrica, el modelo de color escogido es el  $L_1L_2L_3$ , el cual fue definido por Gevers y Smeulders en 1999 [11], este modelo de color es invariante al punto de vista, la geometría del objeto y la iluminación. Estas características no las posee el modelo  $RGB$ , lo cual reduce considerablemente el desempeño del reconocimiento de objetos utilizando este modelo. En sus experimentos Gevers y Smeulder mostraron que el modelo de color  $L_1L_2L_3$  alcanzaba la mayor tasa de reconocimiento. Por otra parte, el modelo de color HSV, comunmente usado comparte estas características, pero es más costoso computacionalmente hablando el cambio de RGB a HSV que de RGB a  $L_1L_2L_3$ . Las ecuaciones [2.12-2.14] describen las dimensiones del modelo de color  $L_1L_2L_3$ .

$$l_1 = \frac{|R - G|}{|R - G| + |R - B| + |G - B|} \quad (2.12)$$

$$l_2 = \frac{|R - B|}{|R - G| + |R - B| + |G - B|} \quad (2.13)$$

$$l_3 = \frac{|G - B|}{|R - G| + |R - B| + |G - B|} \quad (2.14)$$

La distribución se discretiza en 8 niveles para cada dimensión del modelo  $L_1L_2L_3$ . Por definición cada dimensión del modelo de color se encuentra entre  $[0, 1]$ .

### Coefficiente de Similitud

La medida de similitud utilizada para comparar las distribuciones de los objetos candidatos es el coeficiente de Bhattacharyya (Ecuación 2.15, Figura 2.4), el cual ha demostrado mejor desempeño que la distancia Jeffrey, la distancia Minkowski y la estadística- $\chi^2$  [28].

$$\rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^m \sqrt{\hat{p}_u(y)\hat{q}_u} \quad (2.15)$$

donde:

- $\hat{p}_u(y)$ .  
Es la distribución estadística del objeto candidato que se está comparando con el  $OI$   $\hat{q}_u$ .
- $\hat{q}_u$ . Es la distribución estadística del objeto de interés  $OI$ .
- $\rho = 1$ . Cuando se comparan dos distribuciones idénticas.

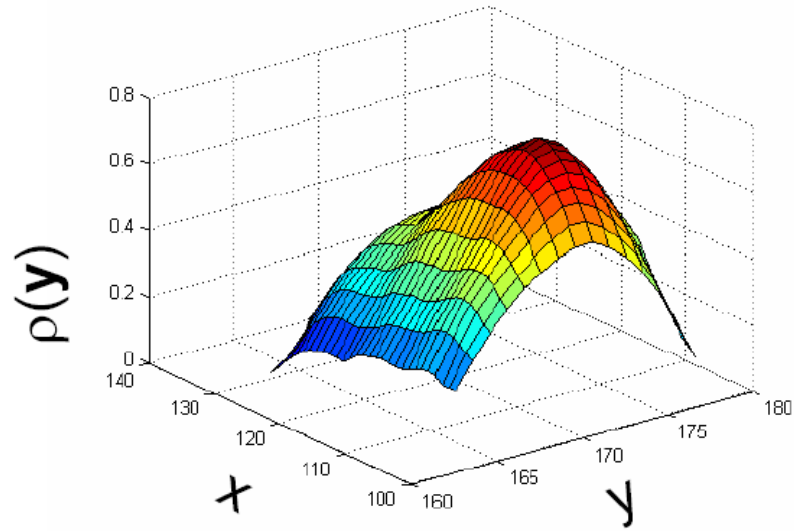


Figura 2.4: Ejemplo del coeficiente de Bhattacharyya.  
Medida de similitud evaluada en diferentes lugares dentro de la imagen.

### Actualización del modelo del objeto de interés

La distribución original del objeto de interés varía a lo largo de su desenvolvimiento en el área de cobertura de la cámara, ya sea por cambios de iluminación o por factores inherentes al movimiento en si del objeto. Por esto se necesita actualizar la distribución del modelo.

Para establecer cuando se debe ajustar el modelo de distribución del objeto modelo se emplea como criterio el coeficiente de Bhattacharyya. Este coeficiente indica cuando las dos distribuciones analizadas son lo suficientemente similares como para no actualizar la distribución del modelo del objeto de interés. Un valor del coeficiente igual a uno indica que existe una correspondencia biunívoca entre las dos distribuciones, en cambio un valor menor a un umbral determinado estipula que debe modificarse el modelo original para que la distribución del objeto de interés se adapte mejor a los cambios.

El modelo actualizado se obtiene mediante un promedio ponderado entre la distribución original y la distribución en la imagen actual utilizando Ecuación 2.16:

$$\hat{q}_u = \beta \hat{q}_u + (1 - \beta) \hat{p}_u(y) \quad (2.16)$$

donde  $y = (x, y)$  es el centro de la ventana de seguimiento actual y  $\beta$  es una constante que ajusta el porcentaje de contribución de las dos distribuciones.  $\beta$  se obtiene empíricamente.

### Algoritmo básico de seguimiento.

1. Captura de imagen patrón.
2. Extracción de la distribución del objeto de interés  $\hat{q}_u$ .
3. Captura de la siguiente imagen de la secuencia.
4. Búsqueda de la posición de la distribución del objeto de interés en la primer imagen obtenida  $(x, y) = (posX, posY)$  .
5. Predicción. Realización de la predicción del estado siguiente usando el Filtro de Kalman ( $FK$ ) como estimador. Ecuaciones 2.5 y 2.6.
6. Medición. Comparación de objetos de interés ( $OI$ ) con objetos candidatos ( $OC$ ). El algoritmo se muestra en la Figura 2.6.
7. Actualización de la distribución del objeto de interés  $\hat{q}_u$ .
8. Corrección. Corresponde a la etapa de corrección del Filtro de Kalman ( $FK$ ). Ecuaciones 2.8 y 2.9. Regresa al paso 5.

El algoritmo básico de seguimiento de un objeto (Figura 2.3) realiza una serie de comparaciones del  $OI$  con el  $OC$ , Figura 2.6. Este algoritmo de comparaciones se describe a continuación.

### Algoritmo de comparacion $OI-OC$ .

Algunos métodos de seguimiento como el algoritmo de desplazamiento medio utilizan el gradiente descendiente con la finalidad de determinar que objetos de interés van a comparar, y lo hacen hasta alcanzar un umbral de certeza deseado [6]. Sin embargo, el cálculo del gradiente descendiente para determinar a que objeto de interés se va a considerar impone un costo computacional adicional considerable. En este trabajo de investigación se optó por utilizar una técnica más sencilla, la cual consiste discretizar la vecindad de la posición estimada en ocho secciones, lo cual permite disminuir el costo de comparación, considerando que se tiene un método confiable de discernimiento entre los objetos de interés. El objetivo de comparar la distribución  $OI$  con los objetos candidatos dentro de una vecindad a la posición predicha por el filtro de Kalman del  $OI$  es dar mayor robustez al sistema de seguimiento al poder verificar dentro de un radio una posible evidencia de un desplazamiento diferente al estimado. La comparación de las distribuciones de los modelos de los  $OC$  se hace del modelo del  $OC$  correspondiente  $p_0$  hasta  $p_8$  utilizando el esquema de vecindad mostrado en la figura 2.5. La separación  $\Delta$  entre las posiciones de los objetos candidatos se hace de forma arbitraria, aunque con una  $\Delta$  mayor se permite que el  $OI$  pueda ser seguido aún con cambios importantes en

su velocidad, en contraparte ante una  $\Delta$  más grande se cuenta con una mayor oscilación en la determinación de la posición del objeto.

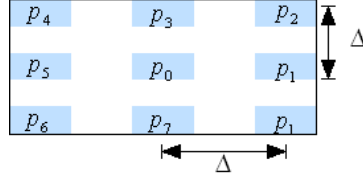


Figura 2.5: Vecindad de Objetos Candidatos.

Esquema de distribución de los  $OC$  con los que el  $OI$  se comparará con el fin de encontrar una posible distribución con más semejanza al  $OI$  que  $p_0$ .

1. Inicializar  $numOC = 0$ , el cual determina que número de objeto candidato ( $p_{numOC}$ ) se está comparando con el objeto de interés  $q_u$ . Se inicializa también la  $p_{max} = p_{numOC}$ , esta variable indica qué vecino ( $OC$ ) se asemeja más al  $OI(\hat{q}_u)$ . La posición  $(X, Y)$  del  $OI$  corresponde a la posición del  $OC$  más semejante al aplicar el coeficiente de Bhattacharyya(2.15) como medida de semejanza.  $E_{max} = 0$  similitud máxima analizada.
2. Verificar si  $numOC < MAXOCs$ , se verifica si ya se compararon todos los posibles  $OCs$  con el  $OI$ . Si ya se examinaron todos regresa la posición  $(posX, posY)$  del  $OC$  candidato que se asemeja más al  $OI$ . Si no se han examinado todos, se prosigue analizando todos los  $OC$ .
3. Obtener la distribución modelo del objeto candidato  $p_{numOC}$  en  $(posX, posY)$
4. Compara  $OI(\hat{q}_u)$  con  $p_{numOC}$ . Si la similitud de  $p_{numOC}$  con  $q_u$  es mayor que la de  $p_{max}$  con  $q_u$  entonces  $p_{max} = p_{numOC}$ .
5. Se establece el siguiente valor para  $p_{numOC}$ . Se regresa a al paso 2.

## 2.5. Algoritmo de Seguimiento de Múltiples Objetos *MTT*

El seguimiento de múltiples objetos (MTT- Multiple Target Tracking) es una tarea que requiere el empleo de gran cantidad de recursos; debido a que para llevar a cabo el seguimiento de  $n$  objetos, se necesita un esfuerzo computacional de  $n^2$  veces la situación particular donde sólo se sigue a un objetivo [13].

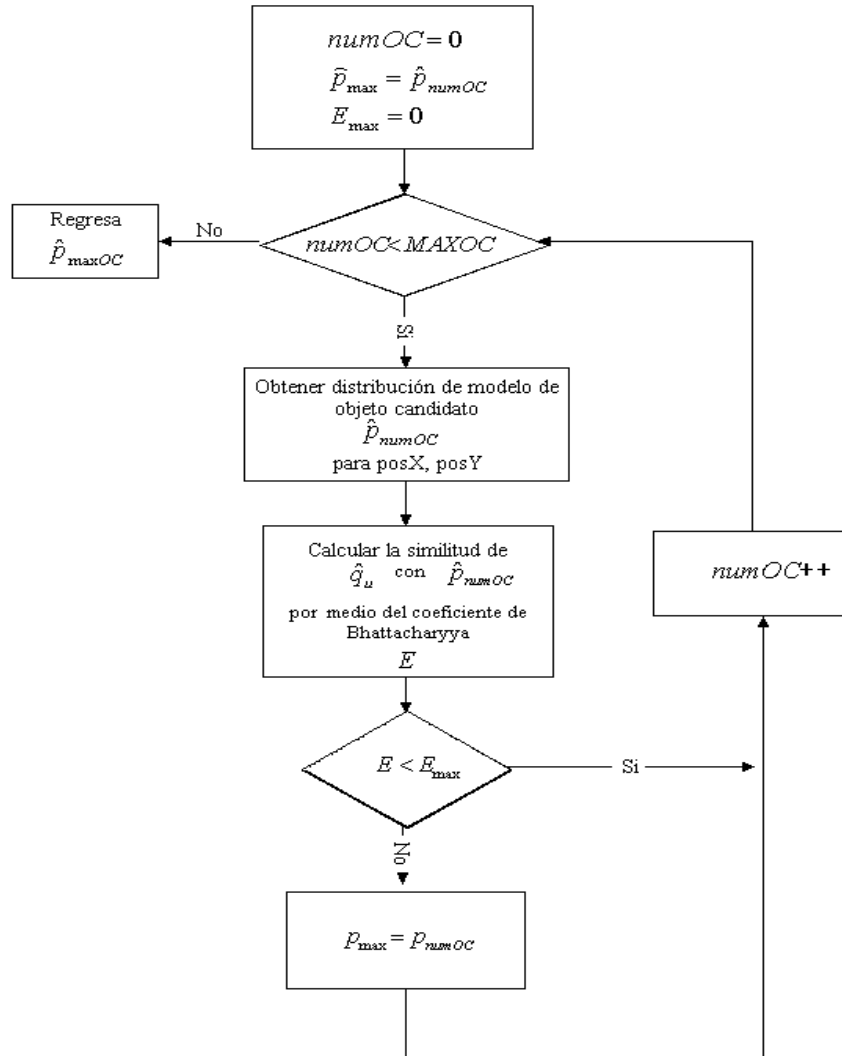


Figura 2.6: Comparación de Objeto interés vs. Objetos candidatos. Diagrama de operación de la comparación entre el objeto de interés (OI) y las distribuciones vecinas a la predicción.

La extensión de un sistema STT(Single Target Tracking) a un sistema MTT requiere una lógica de asociación de datos compleja para clasificar la información sensada en categorías generales de interés y desechar la que es de fuentes recurrentes [4]. Las metodologías para realizar el seguimiento de múltiples objetos se pueden dividir en dos grupos básicos:

- Procesamiento por lotes de datos. Se establecen mediante una situación ideal en la que no se pierde información en la etapa de decisión, debido a que el total de las observaciones son procesadas en conjunto.
- Procesamiento recursivo o secuencial. El procesamiento en este tipo de técnicas se hace en cada muestreo del ambiente.

Los métodos que utilizan el enfoque de procesamiento por lotes de datos suponen una situación óptima, por lo que empiezan rápidamente a excederse en los requerimientos computacionales que necesitan para llevarse a cabo[4]; por esta razón el sistema de seguimiento que se ha implementado es de tipo secuencial.

## Elementos de un Sistema MTT

Blackman-Popoli [4] hacen referencia a una representación funcional de los elementos de un sistema de MTT recursivo; ésta se puede observar en la Figura 2.7.

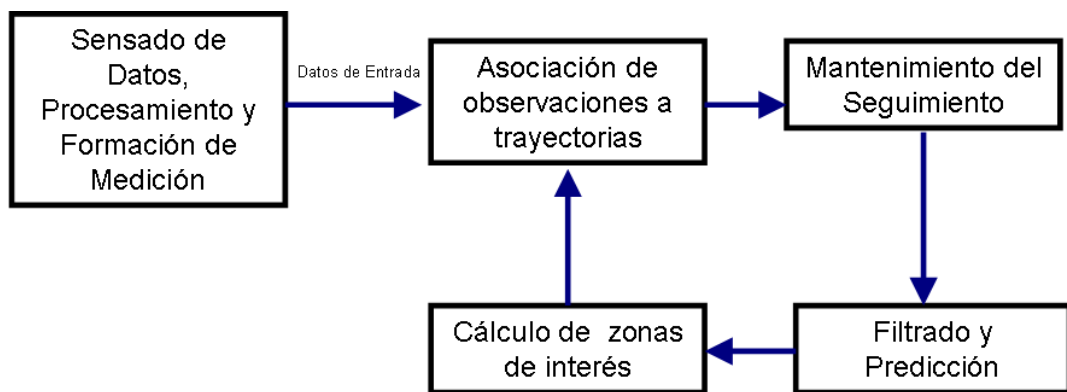


Figura 2.7: Elementos básicos de un sistema MTT convencional [4].

A continuación se hace una breve descripción de algunos elementos relevantes que forman parte de la estructura de un sistema MTT.

- **Asociación de datos.** Consiste en encontrar la posición verdadera de un objeto en movimiento en presencia de candidatos igualmente válidos. En nuestro sistema de seguimiento, el uso del color como atributo permite realizar una asociación de datos confiable, debido a que este atributo permite discriminar a los *WMRs* entre sí y del ambiente. La Figura 2.7 ilustra tres componentes que realizan la función de asociación:
  - *Cálculo de regiones de interés (Gating).* Reduce el cálculo innecesario de evaluaciones de observaciones para su inclusión en una trayectoria. Es una región en el espacio de mediciones. Las regiones de interés se limitan a la comparación de la vecindad de la posición estimada del objeto que se está siguiendo, estas regiones son relativamente pequeñas lo que permite reducir la posible identificación errónea de otro objeto con nuestros objetos de interés.
  - *Asociación de Observaciones a Trayectorias.* Toma las observaciones de los objetos de interés que satisfacen la región de interés y determina cuáles pares corresponden a la trayectoria que el objeto está describiendo a lo largo



de su desplazamiento[4]. La implementación de múltiples filtros de Kalman permite asociar las nuevas observaciones del desenvolvimiento de los objetos de forma transparente y mantener de esta forma las trayectorias examinadas de cada uno de ellos.

- *Mantenimiento del seguimiento.* En el sistema de seguimiento utilizado cada filtro de Kalman realiza el mantenimiento del seguimiento de forma independiente al realizar el análisis del movimiento del objeto de interés en desplazamiento. La implementación de múltiples filtros de Kalman permite asociar las nuevas observaciones del desenvolvimiento de los objetos de forma transparente y mantener de esta forma las trayectorias examinadas de cada uno de ellos.
- *Filtrado y predicción.* El filtrado incorpora la asignación de observaciones en los parámetros estimados de la trayectoria actualizada. Asimismo, las predicciones son hechas al tiempo en el que el siguiente muestreo es recibido [4]. Los sistemas modernos de tracking usan el filtrado de Kalman. Éste provee la predicción del estado así como la medida de la exactitud de la predicción, lo cual determina el tamaño de la región de interés(gate). La interacción de múltiples filtros de Kalman son ampliamente aceptados cuando existen condiciones que dificultan la manipulación de las entidades [15].

### 2.5.1. Modelo MTT

El algoritmo de seguimiento *MTT* utiliza un filtro de Kalman por cada objeto analizado de forma paralela en periodos de muestreo de  $tms$ , figura 2.8. Este procedimiento es esencialmente una extensión del algoritmo descrito para realizar el seguimiento de un único objeto descrito en la sección pasada, figura 2.8.

En la figura 2.8 se muestra el algoritmo implementado para realizar el seguimiento de múltiples objetos. El procedimiento básico que utiliza este algoritmo es asociar un filtro de Kalman por cada objeto que se desea seguir, para esto, los filtros de Kalman realizan el análisis de forma paralela en periodos determinados. Este esquema permite estimar las posiciones de los objetos de forma independiente en base a su desenvolvimiento dentro de su trayectoria. Las estimaciones de las posiciones permiten reducir el área de búsqueda del objeto en el siguiente periodo de tiempo y el método de localización incorporado mediante el uso del modelo de color  $L_1L_2L_3$  proporciona mayor eficiencia en la comparación y discernimiento de las distribuciones candidatas durante el seguimiento.

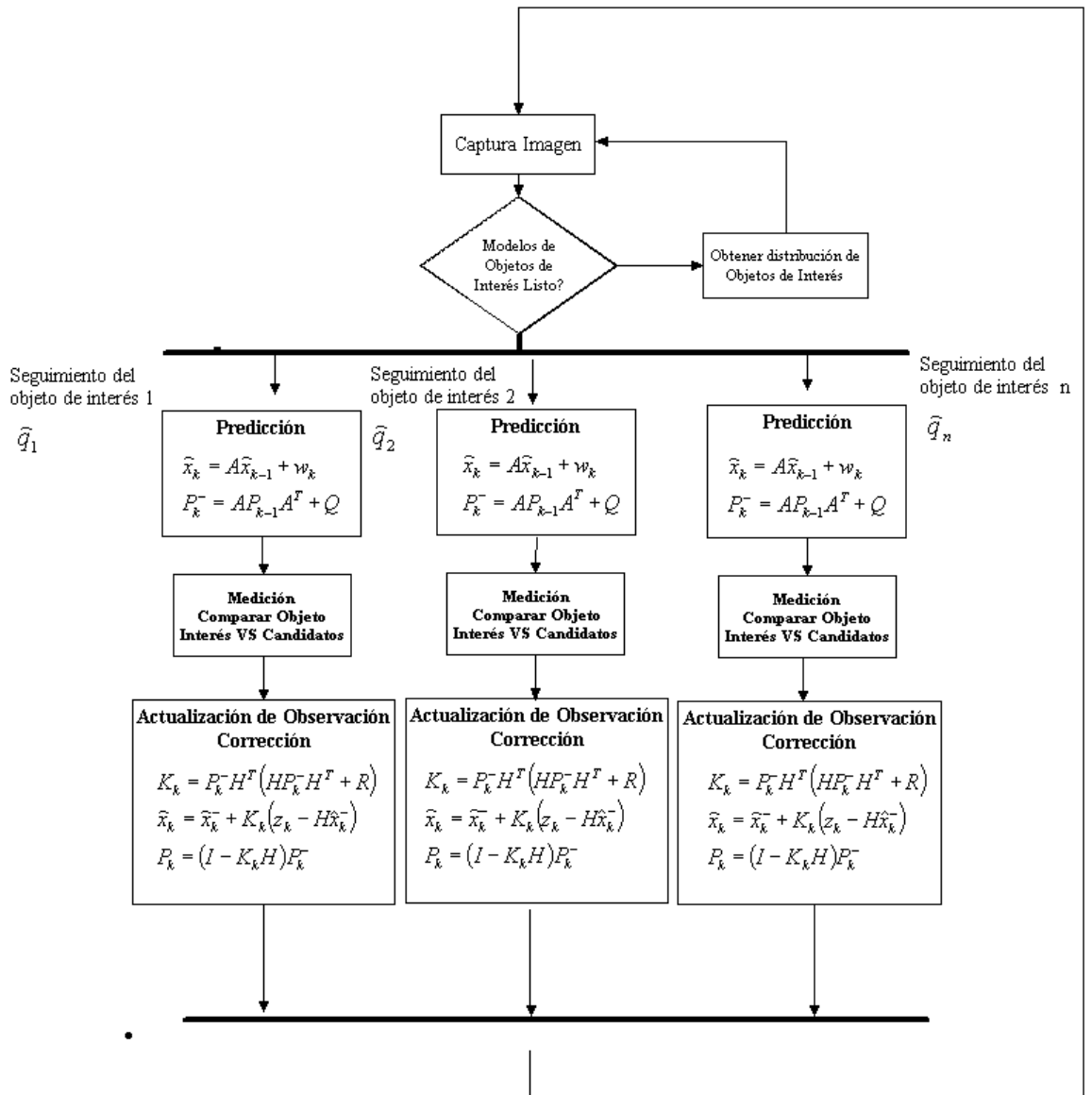


Figura 2.8: Algoritmo MTT.

Algoritmo de seguimiento de múltiples objetos mediante el uso de  $FK$  paralelos.

## 2.6. Resumen

En este capítulo se presentó, en primera instancia una introducción a los sistemas de seguimiento y las herramientas matemáticas empleadas en ellos, se hizo énfasis en la importancia del Filtro de Kalman y su uso en este trabajo de tesis. Además, se describieron los esquemas utilizados para realizar el seguimiento de uno y de múltiples objetos.

## Capítulo 3

# Arquitectura Multiagente para el seguimiento de objetos

El empleo de sistemas multiagente tiene como motivación a la naturaleza distribuida de muchos problemas dentro de diversas áreas del conocimiento, lo cual puede provocar un cuello de botella si es que se atacan de manera centralizada [23].

### 3.1. Agentes inteligentes

Russell y Norvig [30] definen a un *Agente Inteligente* como una entidad que mapea percepciones a acciones. Adquiere información de su ambiente (“Percibe” el ambiente), decide sus acciones y las realiza. Además, en [30] también se establece la siguiente clasificación de las propiedades de los ambientes mostrados:

- *Completamente observables vs parcialmente observables.* Un ambiente observable es aquel en el que el agente puede obtener información completa, exacta y actualizada acerca del estado del ambiente.
- *Episódico vs Secuencial.* En ambientes episódicos la acción que realizará el agente depende solamente del episodio actual, en cambio en un ambiente secuencial, la decisión actual puede afectar todas las posteriores.
- *Determinísticos vs no-determinísticos.* Un ambiente determinístico es el que cualquier acción tiene garantizado un efecto determinado, es decir, no existe incertidumbre acerca del estado que se alcanzará después de realizar esa acción.
- *Estáticos vs dinámicos.* Un ambiente estático es aquel que se asume que no cambia mientras se realiza el proceso de decisión.
- *Discretos vs continuos.* Un ambiente es discreto si cuenta con un número fijo y finito de acciones y percepciones.

- *Monoagente vs Multiagente.* Un sistema multiagente es aquél en el que el nivel de abstracción utilizado es el agente. Pretende proporcionar los principios de construcción de sistemas complejos que involucran múltiples agentes y mecanismos para la coordinación de los comportamientos de los agentes independientes.

El ambiente en el cual se desarrollará la presente investigación es de tipo multiagente, continuo, dinámico, no-determinístico, secuencial y parcialmente observable, el cual es considerado el tipo de ambiente más complejo.

### 3.1.1. Tipos de Arquitecturas de Agentes

Ferber [10] hace una clasificación de los principales tipos de arquitecturas de agentes tomando en cuenta algunos parámetros del análisis estructural de las organizaciones artificiales.

- *Arquitectura horizontal modular.* Se considera como un ensamblaje de módulos los cuales tienen una función horizontal específica. Los módulos más usados son:
  - Funciones de percepción.
  - Enviar e interpretar comunicaciones.
  - Modelación del entorno.
  - Administración de las obligaciones.
  - Manejo del dominio de habilidades.
  - Administración de metas y toma de decisiones.
  - Planeación de acciones.
- *Arquitectura basada en pizarrón (pizarrón).* Se basa en la división en módulos independientes los cuales no se comunican datos de forma directa, pero interactúan de forma indirecta mediante la compartición de un área de datos.
- *Arquitectura Subsumption.* En esta arquitectura el agente se divide en módulos verticales, donde cada uno de ellos es responsable de un comportamiento muy limitado.
- *Tareas Competitivas.* Un agente se compone de un ensamblaje de tareas, sólo una de ellas puede estar activa a la vez. Estas tareas entran en competencia para la selección mediante un mecanismo de decisión el cual toma en cuenta varios parámetros: el peso de determinada tarea en un momento determinado, el contexto de la aplicación, y la información proveniente del exterior.

- *Sistemas de producción.* Se define como la combinación de una base de datos, una base de reglas de producción y un intérprete.
- *Sistemas basados en clasificadores.* Estos sistemas toman la forma de una variación de los sistemas de producción, donde los hechos (Mensajes) son cadenas de caracteres con una longitud fija y las reglas (Clasificadores) son pares de cadenas de caracteres en los que los elementos son tomados del mismo alfabeto aumentando el uso de un carácter comodín. La información entrante es codificada en hechos y las reglas son aplicadas a estos hechos, lo cual puede producir otros hechos para activar otros clasificadores o para realizar acciones sobre el entorno de trabajo.
- *Arquitectura Conexionista.* Está basada en el esquema del cerebro, está realizada por elementos idénticos que son denominados neuronas.
- *Arquitectura basada en sistemas dinámicos.* El comportamiento del agente es implementado simplemente por la descripción directa de las ecuaciones las cuales ligán los valores de los sensores a los comandos de los actuadores.

### 3.1.2. Arquitectura de pizarrón

Esta arquitectura usa un depósito central para poder compartir información. La información del pizarrón representa hechos, consideraciones y deducciones realizadas por el sistema durante de la solución de un problema. Cada agente percibe información del pizarrón y trata de contribuir con la solución. Un facilitador realiza la negociación entre agentes que intentan escribir en el pizarrón.

Corkill hace un compendio de la relación entre las arquitecturas pizarrón y los Sistemas Multiagentes (*MAS*) [7]:

Un sistema de pizarrón consiste de tres componentes básicos:

- *Fuentes de Conocimiento (Knowledge sources-KSs).* Son módulos computacionales independientes que de forma conjunta contienen la información necesaria para resolver el problema.
- *Pizarrón .* Es un depósito central que recibe la información de entrada, soluciones parciales y otros datos en varios estados dentro de la solución del problema. Este depósito puede servir como:
  - Memoria compartida de datos, soluciones parciales, alternativas y soluciones finales; asimismo de información de control.
  - Un medio de comunicación y buffer.
  - Un mecanismo disparador de *KSs*

- *Componente de control.* Realiza las decisiones en tiempo de ejecución con respecto al curso que tomará el proceso de solución y también a la repartición de recursos de la solución. En un esquema de control en un sistema tradicional *pizarrón* la ejecución de una KS genera eventos a medida que va haciendo cambios en el *pizarrón*. Estos eventos son administrados hasta que la ejecución de la KSs es completada. Los eventos generados tienen la potencialidad de activar KSs. Las activaciones de las KSs son ordenadas, y la más apropiada es escogida para ejecución.

En la Figura 3.1 se muestra la estructura básica de un sistema *pizarrón*.

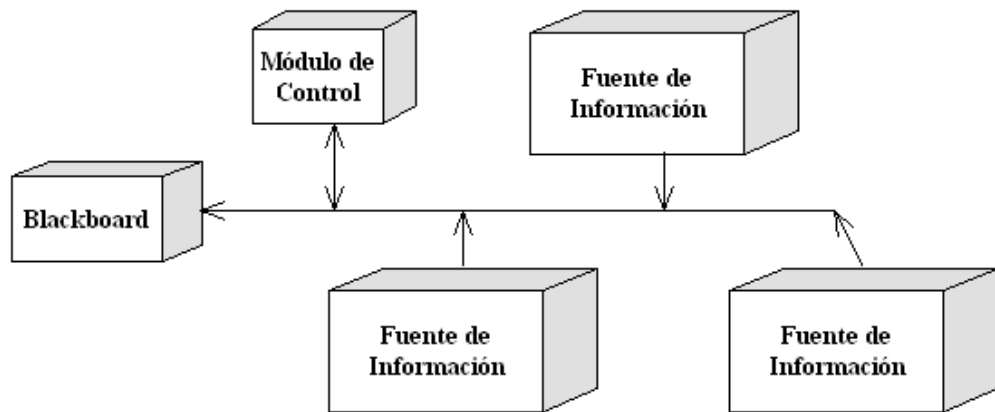


Figura 3.1: Componentes de un *Sistema pizarrón*

## 3.2. Fusión de Datos en Sistemas Multisensor

En esta investigación se necesitará emplear técnicas de fusión y asociación de datos para unificar las percepciones de los diferentes sensores(cámaras) que estarán distribuidos en el entorno que se analizará.

De acuerdo al *JDL Data Fusion Lexicon*[14], fusión de datos es definida como:

“Un proceso que trata con la asociación, correlación, y combinación de datos e información de una o varias fuentes para lograr estimados refinados de posición e identidad. Se caracteriza por mejoras continuas de sus estimados, y de la evaluación de la

necesidad de fuentes adicionales, o modificaciones del proceso en si ”

Los sensores múltiples pueden estar co-localizados, como el radar, el ESM (*Electronic Support Measures*) y elIRST(Infrared Search and Track) en un sistema de control de tráfico aéreo, o distribuidos, como en un sistema de radares a lo largo de un país. Además pueden surgir combinaciones entre los dos esquemas anteriores en un mismo sistema [4]. En la Figura 3.2 se muestra una arquitectura básica de un sistema de seguimiento/fusión.

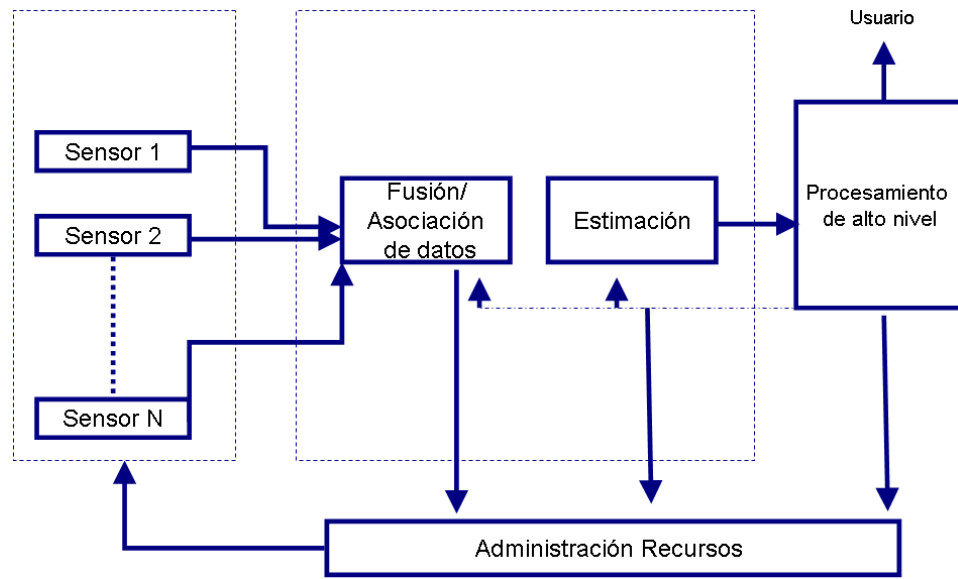


Figura 3.2: Sistema de seguimiento/fusión de múltiples sensores [4]

Algunos de los problemas prácticos que surgen al desarrollar sistemas de múltiples sensores incluyen la sincronización y alineación de las mediciones, las diferencias de precisión, y resolución de los sensores.

La sincronización de los datos es una de las tareas más importantes que se deben llevar a cabo en el transcurso del proyecto, los agentes analizarán el ambiente en instantes diferentes, lo que implica que la información reportada por los agentes cámara al pizarrón no corresponderá al mismo momento, por lo que se debe buscar alguna forma de encontrar una correspondencia temporal entre las mediciones de los agentes cámara con el fin de de realizar fusión de datos en regiones se encuentren traslapadas áreas de interés de varios agentes cámara. Se han utilizado diversas técnicas para resolver este problema, entre ellas se pueden mencionar *time-stamping* y mediciones que reflejan el retraso. La primera es óptima porque las mediciones se procesan en el orden adecuado, mientras que la segunda es subóptima pero de mayor simplicidad [4]. En este proyecto de investigación usaremos *time-stamping* para decidir si dos mediciones tienen suficiente correspondencia temporal entre ellas para poder fusionarlas.

### 3.3. Arquitectura Multiagente propuesta

#### 3.3.1. Descripción

En esta sección se presenta una arquitectura multiagente de tipo *pizarrón* para realizar el seguimiento de múltiples robots. Este tipo de arquitectura ha sido ampliamente usada para atacar los problemas con características que contengan incertidumbre [16]. Las aplicaciones que utilizan este tipo de esquema han sido creadas para resolver diversos tipos de problemas como sistemas de comunicación, movilidad y coordinación en tiempo real[16]. En la figura 3.3 se ilustra el diagrama del sistema de seguimiento multicámara que utiliza una arquitectura multiagente para su funcionamiento.

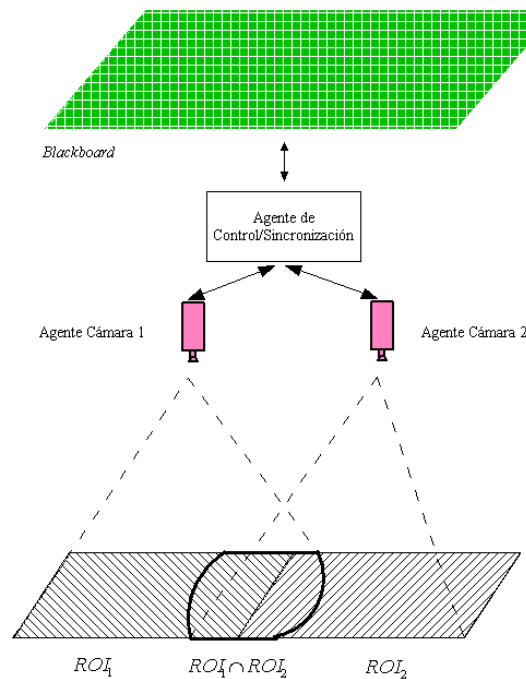


Figura 3.3: Esquema ilustrativo del sistema.

En esta arquitectura se cuenta con los siguientes elementos, Figura 3.4:

- *Pizarrón*. Es el repositorio de toda la información en la cual la fuente de conocimiento opera, y el destino de todas las conclusiones de la fuente de información. Existen dos tipos de conocimiento: estático y dinámico. El conocimiento estático no cambia, es información invariante en el largo plazo, mientras que el dinámico es el conocimiento generado durante la ejecución [16].
- *Conjunto de Agentes*. Se definen dos clases de agentes:



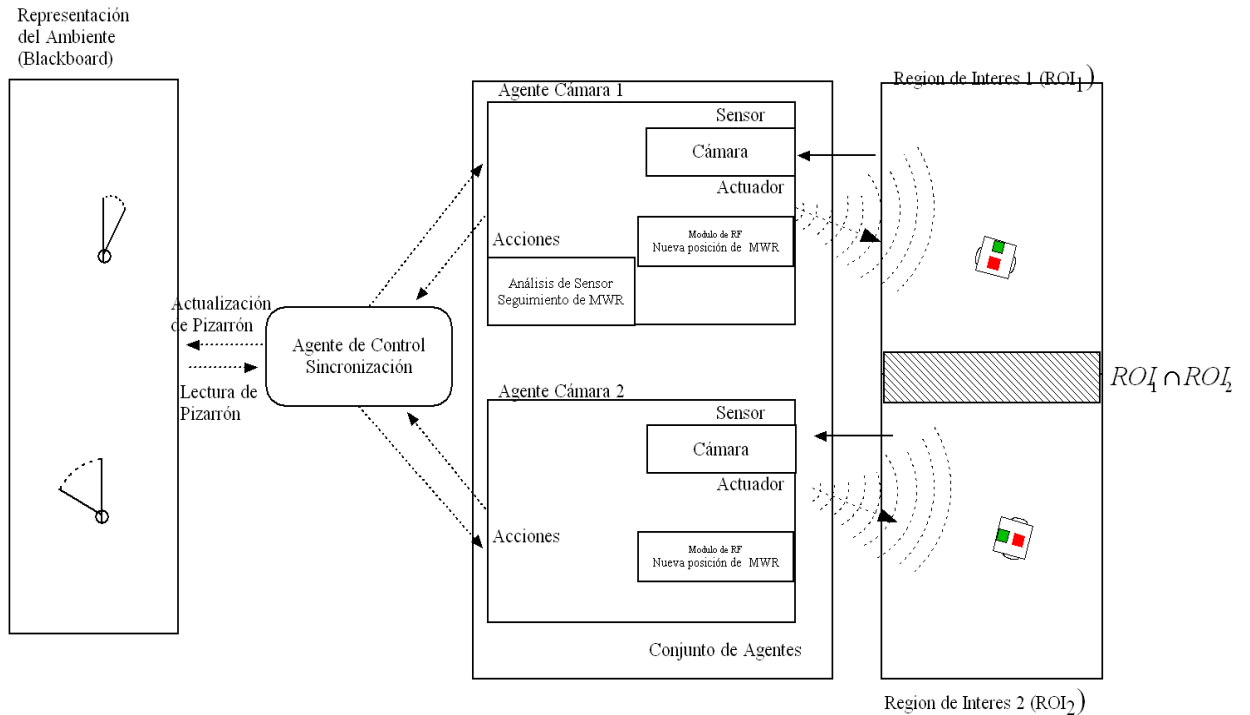


Figura 3.4: Arquitectura del sistema multiagente: Agentes Cámara y Control. El primero tiene como objetivo primordial realizar el seguimiento de los objetos dentro de su área de interés, el segundo administra la información que se desea escribir en el *pizarrón*

1. *Agente de Control*. Como en Dong-Chen [16], se define un agente de control, el cual cumple con la tarea de sincronizar y de servir de interfase entre los agentes cámara y el *pizarrón*. Este agente determina cuando se debe realizar la fusión de datos de agentes cámara que están analizando al mismo objeto en una región de interés (*ROI- Region of interest*) traslapada. Este agente consta básicamente de cuatro estados de funcionamiento, Figura 3.5.
  - *Escuchando*. En este estado se atienden los mensajes de los agentes cámara que desean registrarse y posteriormente actualizar el *BB*.
  - *Actualizando*. Renueva la información correspondiente de un agente cámara dentro del *BB*. Cuando un agente ya está registrado y comienza a realizar el seguimiento de los objetos dentro de su área de interés, envía un mensaje de solicitud de actualización al agente de control, de esta forma, si el agente de control se encuentra en su estado escuchando, pasa al estado *Actualizando*, donde se actualiza la información del agente cámara y se realiza la fusión de datos de los objetos de interés que están siendo analizados a su vez por otro agente cámara.
  - *Registrando*. Actualiza el repositorio de información con el registro de

un nuevo agente cámara. Este estado es alcanzado debido a la solicitud de registro de un nuevo agente cámara. Una vez hecho el registro del agente cámara solicitante se le envía como respuesta una confirmación de registro.

- *Buscando*. Busca información en el *BB* de los *OI* no encontrados por el agente cámara que solicitó la actualización y se le envía como respuesta los identificadores de los segmentos de imagen más cercanos a cada uno de los objetos no encontrados o perdidos. De esta forma los agentes se benefician del conocimiento de otros agentes, en caso de que algún otro esté siguiendo a un objeto que él no ha encontrado. Además se disminuye el procesamiento necesario para encontrarlos, debido a que no tiene sentido buscar constantemente a un objeto que ningún agente está siguiendo, a menos que aparezca por primera vez en el entorno analizado. Sin embargo, en este caso el procesamiento puede hacerse en intervalos mucho más grandes, sólo si no hay algún agente que este observando a ese *OI*.

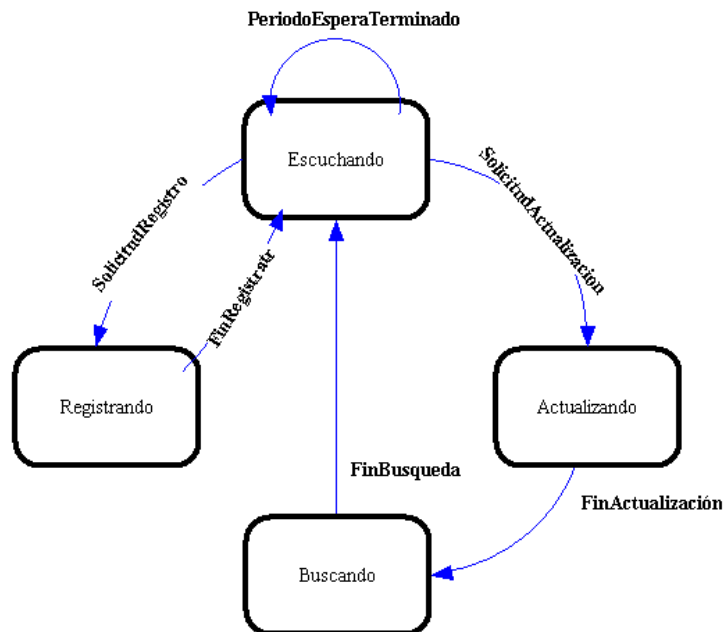


Figura 3.5: Estados del Agente Control. Los posibles estados en los que puede encontrarse el Agente de Control.

2. *Agente Cámara*. Así como en el trabajo desarrollado por Orwell [18], se establece un agente cámara el cual realiza el seguimiento de los objetos (*Wheel Mobile Robots, WMR*) dentro de su área de interés. El agente cámara

tiene la tarea de proveer información de los objetos que se desplazan por su área de cobertura. Para esto, notifica en cada periodo  $T$  al agente de control acerca de los cambios en la porción de ambiente percibida por él. Asimismo, solicita información correspondiente a los objetos de interés no percibidos en su segmento de ambiente; esta información le permite localizar a un  $OI$  que se está incorporando al ambiente percibido mediante la localización realizada por otro agente cámara. El agente cámara tiene cuatro posibles estados de funcionamiento, Figura 3.6:

- *Solicitando Registro.* Solicita el registro del agente cámara ante el agente de control y espera la confirmación de registro, para iniciar el seguimiento de los  $OI$  dentro de su área de cobertura.
- *Siguiendo.* El agente cámara localiza y realiza el seguimiento de los  $OI$  que se desplazan por su área de interés. Los  $OI$  que no son localizados, son considerados como perdidos por el agente cámara, por lo que son buscados periódicamente en el área de interés y también cuando se recibe información proporcionada por el agente de control con respecto a la posición de alguno de ellos.  
Además de esto, determina qué objetos no están dentro de su porción de ambiente analizada.
- *Notificando.* En esta fase el agente cámara solicita la actualización de la información al agente de control, de los datos correspondientes a los  $OI$  que están siendo analizados por él, con el objetivo de actualizar dentro del  $BB$  la información de los  $OI$  que son examinados por él.  
Asimismo, solicita información de la localización de los  $OI$  que no han sido encontrados en el área de interés, esto con la finalidad de localizarlos mediante información de otros agentes que pueden o no estar compartiendo cierto segmento de cobertura. La información recibida indica qué segmento de la imagen analizada se encuentra más cerca del  $OI$  no encontrado.
- *Verificando Información.* Analiza el segmento de la imagen indicado por el agente de control para verificar si hay movimiento y si se detecta éste, se busca al  $OI$  que no se había localizado o que se había perdido.

### 3.4. Interacción entre Agente Cámara y Agente de Control

La interacción entre los dos tipos de agentes se lleva a cabo en dos situaciones, en el registro de un agente cámara y durante el reporte de información del agente cámara

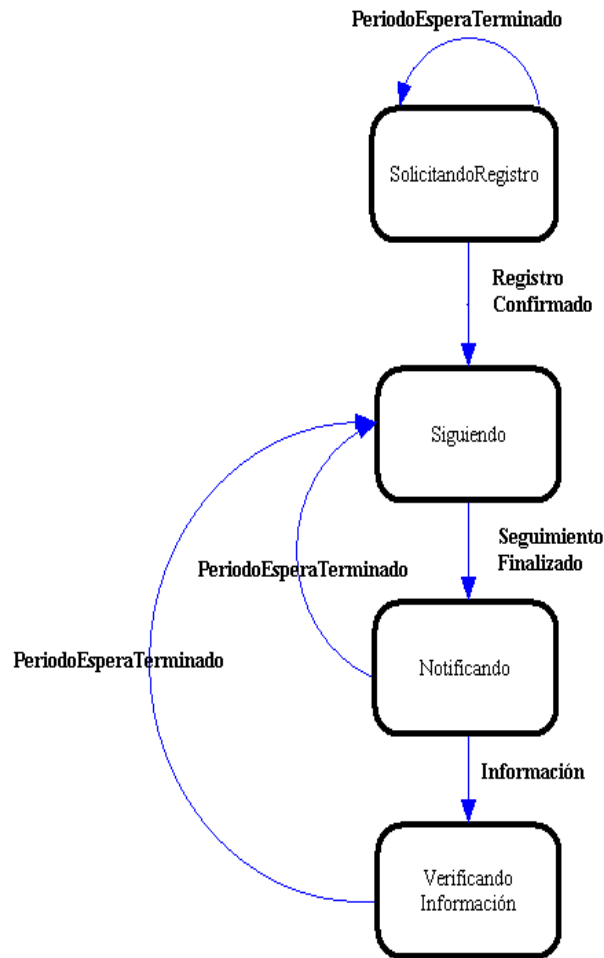


Figura 3.6: Estados del Agente Cámara.

Los posibles estados en los que puede encontrarse el Agente de Cámara. al agente de control.

- *Registro.* El agente cámara solicita su registro al agente de control, para poder realizar el análisis de su área de interés. En caso de no recibir una confirmación de registro, el agente cámara envía su solicitud de registro nuevamente en intervalos constantes de tiempo hasta obtener una confirmación. Una vez hecho esto el agente cámara inicia el seguimiento de los *OI* dentro de su área de cobertura.
- *Actualización/solicitud de información de seguimiento.* El agente cámara solicita la actualización de la información en el *pizarrón* correspondiente a los *OI* que se encuentran en su área de cobertura. Asimismo, el agente cámara pide información de los *OI* que no ha localizado dentro de su área de interés al agente de control cada vez que realiza una solicitud de actualización en periodos  $\Delta t$ .

El agente de control realiza la actualización del *pizarrón* y verifica si hay algún agente cámara que esté analizando el movimiento de los objetos que no han sido localizados por el agente cámara solicitante de actualización. Si es así, evalúa la proximidad de estos objetos a subzonas definidas dentro del área de interés del agente cámara y envía el identificador de la zona más cercana al objeto no localizado.

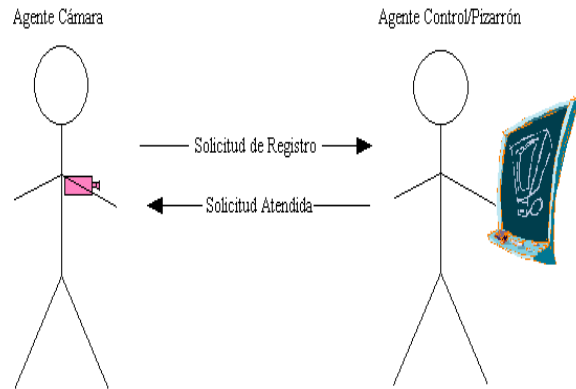


Figura 3.7: Registro de Agente Cámara.

Los agentes cámara solicitan su registro en el *pizarrón* para poder realizar el seguimiento de los objetos dentro de su área de cobertura.

El agente cámara espera la información de las zonas donde debe de buscar a los *OI* que no ha localizado. La información se espera durante un intervalo tiempo que no afecte el periodo constante en el que se debe realizar el análisis del movimiento para que sea válido el modelo de sistema empleado en el *FK*.

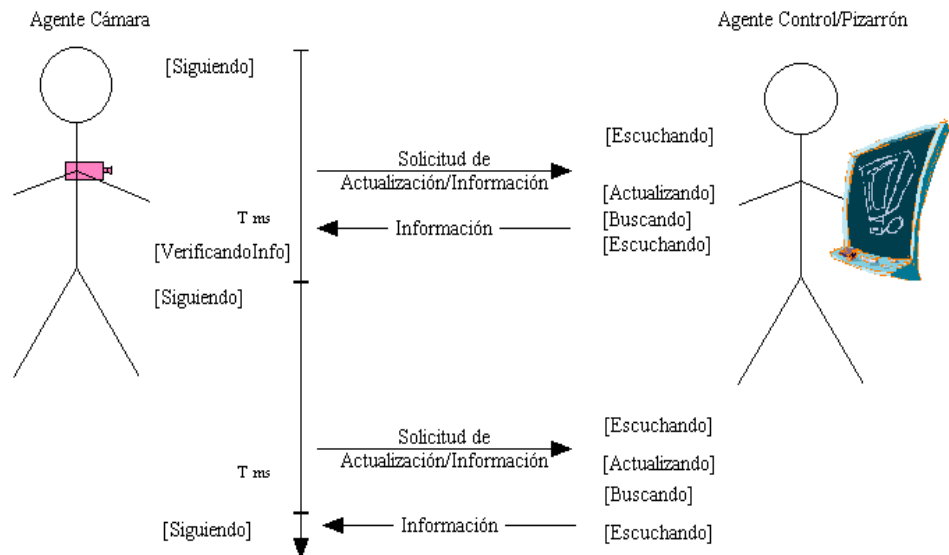
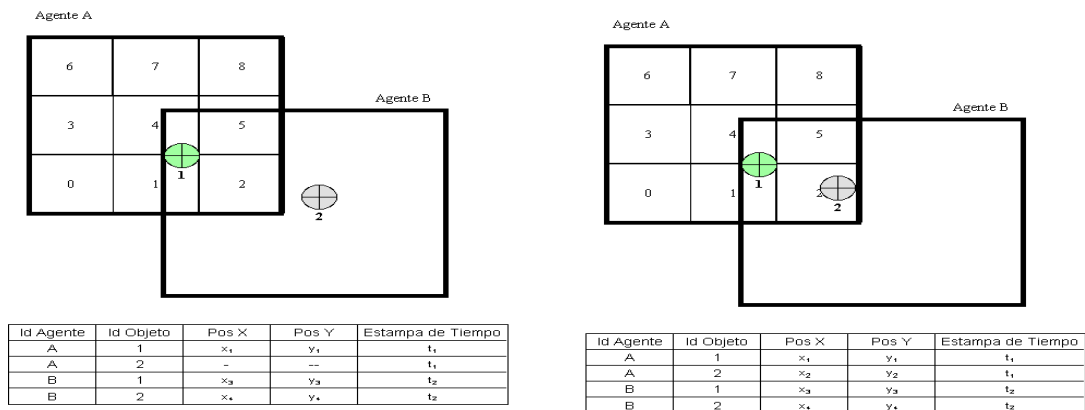


Figura 3.8: Actualización/Solicitud de Información de Agente Cámara.

Cada agente cámara realiza el análisis de seguimiento de sus *OI* en cada periodo  $\Delta t$  y realiza una solicitud de actualización/información al agente de pizarrón. El agente de pizarrón responde con la información de los *OI* que no han sido localizados por el agente cámara.

En la figura 3.9 se presenta un ejemplo donde se muestra la importancia del intercambio de información entre agentes, lo cual permite encontrar a un *OI* perdido o no localizado. En este caso el agente A sigue al objeto 1, pero el objeto 2 no es localizado.

El agente cámara A pide información de los objetos que no está siguiendo, objeto 2 en este ejemplo, cada vez que solicita la actualización del pizarrón al agente de control. Éste, a su vez busca información en el pizarrón correspondiente al objeto 2. El agente de control encuentra información de la posición del objeto 2 en el pizarrón debido a que el agente cámara B lo está siguiendo. El agente de control determina cuál es la zona, en este caso la número 2, donde el agente cámara A debe buscar al objeto 2 y se la envía como respuesta a su solicitud de actualización de información. El agente cámara A recibe esta información y busca al objeto 2 en esa zona, si lo encuentra comienza a seguirlo a partir de ese instante, de lo contrario lo sigue considerando como no localizado y por lo tanto continúa con la petición de información, correspondiente a este objeto, al agente de control en los periodos subsecuentes. La condensación en el pizarrón de la información de los agentes cámara permite la reducción de procesamiento, debido a que el agente A sólo buscará en una zona de la imagen en el peor de los casos, porque si no existe otro agente que haya localizado al objeto 2 no tiene caso buscarlo constantemente.



(a) Objeto 2 no localizado. El agente cámara A, no ha localizado al objeto 2 debido a que se encuentra fuera de su área de interés.

(b) Objeto 2 localizado. El agente de control indica al agente cámara A la subzona de interés 2, donde debe buscar al objeto de interés 2. El agente cámara A, verifica esta subzona de interés con la finalidad de localizar al objeto 2.

Figura 3.9: Ejemplo de intercambio de información.

### 3.5. Fusión de datos en el sistema multiagente propuesto

En el *pizarrón* se encuentra la información necesaria para realizar la fusión de datos, sin embargo existe un problema importante que impide fusionar los datos contenidos en él, la diferencia temporal al realizar las estimaciones por parte de los agentes cámara; es decir, que realizan el seguimiento y la notifica

Las estimaciones que los agentes cámara reportan al agente de control cuentan con una estampa de tiempo que sirve para evaluar la proximidad temporal entre cada medición. Si la diferencia entre las estampas de tiempo se encuentra dentro de un umbral que delimita una ventana de tiempo, y las estimaciones corresponden al mismo objeto, pero son analizadas por agentes cámara diferentes, se realiza el proceso de fusión. El esquema de interacción entre los reportes de dos agentes cámara se muestra en la Figura 3.10.

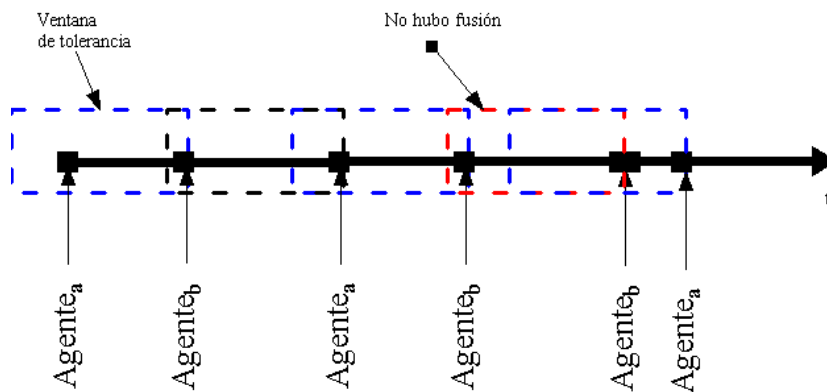


Figura 3.10: Esquema de interacción entre los agentes.

Se muestra cómo la ventana de tiempo es utilizada como criterio para realizar la fusión de los estimaciones entre los agentes.

El algoritmo utilizado para la realizar la fusión de datos de estimaciones realizadas por agentes cámara diferentes sobre un mismo objeto se muestra en la Figura 3.11. Este algoritmo consta de las siguientes etapas:

1. Inicializa el identificador  $i$  del objeto analizar con el  $id$  del primer objeto.
2. Actualiza en el pizarrón la información del objeto  $i$  para el agente cámara solicitante (A).
3. El agente de control (C) verifica si algún otro agente cámara (B) está siguiendo al objeto  $i$ , de lo contrario continúa al paso 11.

- Calcula la diferencia (*dif*) de tiempo entre las estimaciones realizadas por el agente cámara A y B.

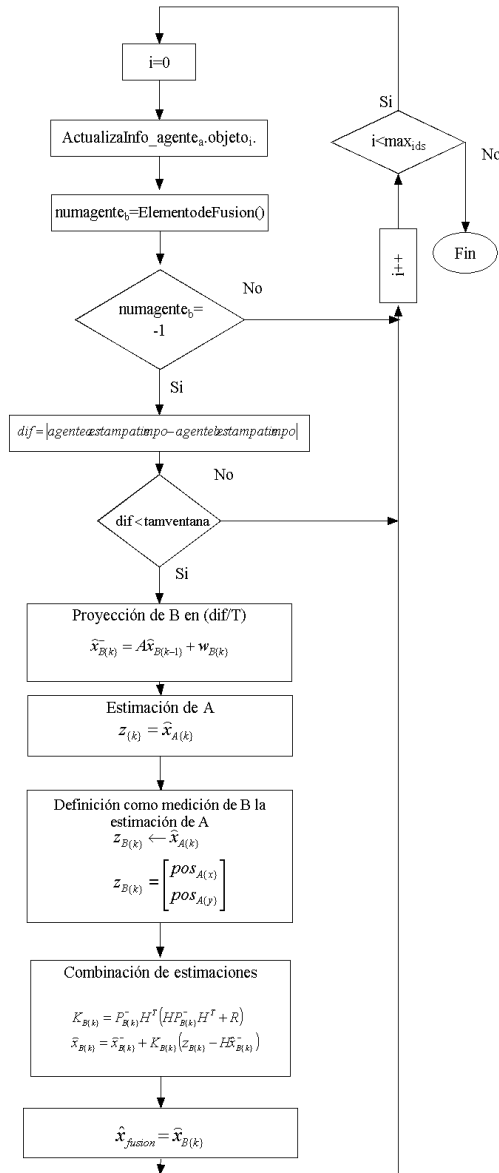


Figura 3.11: Algoritmo de Fusión de Datos.

Se muestra el procedimiento utilizado para realizar la fusión de estimaciones mediante el uso del Filtro de Kalman.

- Si *dif* es menor a un umbral predefinido por *tamventana*, entonces, el proceso de fusión se lleva a cabo ir al paso 6, de lo contrario se analiza el objeto de interés que tiene el siguiente *id*, ir al paso 11.



6. Proyecta la estimación de B en  $diff/\Delta T$ .
7. Se obtiene la estimación( $\hat{x}_k$ ) de A
8.  $\hat{x}_k$  se define como medición de B.
9. El filtro de Kalman fusiona las estimaciones de los agentes cámara a través de la ecuación 2.8, donde  $K_k$  es la ganancia de Kalman del agente cámara B, la estimación realizada por el agente cámara es incorporada como la medición por medio de  $Z_k$  y  $H$  es la matriz de medición que relaciona las mediciones con el sistema.
10. Se obtiene la estimación fusionada  $x_{fusion}^-$  de (A) con (B) proyectado.
11. Se actualiza  $i$  con el id del siguiente objeto.
12. Se verifica si ya se analizaron todos los posibles  $OI$   $i < max_{ids}$ , finaliza proceso de fusión; de caso contrario ir a paso 2.

### 3.6. Resumen

En este capítulo se presentaron las diferentes arquitecturas de los sistemas de agentes, se hizo énfasis en la arquitectura de pizarrón debido a que su funcionalidad y escalabilidad permiten definir entidades que interactúan de forma indirecta en el seguimiento de objetos. Se describió la arquitectura multiagente que se definió para resolver el problema del seguimiento de múltiples objetos en tiempo real con varias cámaras. Se describieron los tipos de agentes que interactúan en el sistema. El intercambio de información entre agentes proporciona la capacidad de realizar el seguimiento de un objeto de interés cuando pasa de una área de interés de un agente cámara a la de otro. Además se presentó un procedimiento para realizar la fusión de datos de las estimaciones de diferentes agentes cámara con respecto a un mismo objeto. En este esquema, el agente de control fusiona las estimaciones de agentes cámara diferentes, con la finalidad de obtener una estimación fusionada que permita mejorar la confiabilidad del sistema.

## Capítulo 4

# Implementación, experimentos y resultados

En el Capítulo 2 de este trabajo de tesis se presentaron varios métodos para realizar el seguimiento del movimiento de objetos. Asimismo, se mencionó que el algoritmo que se iba a implementar en esta investigación era el filtro de Kalman debido a su probada eficiencia y por considerarlo como un excelente punto de partida para el desarrollo de una arquitectura *pizarrón*. En el Capítulo 3 se describieron arquitecturas relevantes de sistemas multiagente y se presentó la arquitectura multiagente propuesta para realizar el seguimiento de los *WMR*, asimismo se expuso el algoritmo para realizar la fusión de datos y el seguimiento intercámara bajo esta arquitectura. En este capítulo se describe la caracterización del Filtro de Kalman y la interacción entre agentes cámara y agente de control con la finalidad de realizar el seguimiento de los *OIs* en una cámara y después extenderlo a múltiples cámaras.

### 4.1. Caracterización del Filtro de Kalman

A continuación se presenta la especificación de las ecuaciones del filtro de Kalman utilizadas en esta investigación.

El funcionamiento del filtro se basa en dos modelos esenciales: el modelo del sistema o proceso y el modelo de medida:

- *Modelo del sistema.* El sistema físico del movimiento se representa con un vector de estados  $\mathbf{x}$ , llamado simplemente estado, y un conjunto de ecuaciones denominado el modelo del sistema. El tiempo de observación se simboliza como  $t_k = t_0 + k\Delta T$ ,  $k = 0, 1 \dots$ ,  $\Delta T$  es el periodo de muestreo y  $\hat{\mathbf{x}}_k^-$  es el estado del sistema (Ecuación 4.1).

Se supone un periodo de muestreo  $\Delta T$  pequeño, por lo que se puede utilizar un modelo de sistema de tipo lineal.

*Ecuación de Estado:* Es la representación del estado actual del sistema.

$$\hat{\mathbf{x}}_{\mathbf{k}} = \begin{bmatrix} x_{P\{k\}} \\ y_{P\{k\}} \\ x_{V\{k\}} \\ y_{V\{k\}} \end{bmatrix} \quad (4.1)$$

Donde:

$x_{P\{k\}}$ : Es la posición sobre el eje  $x$  del objeto en el tiempo  $k$ .

$y_{P\{k\}}$ : Es la posición sobre el eje  $y$  del objeto en el tiempo  $k$ .

$x_{V\{k\}}$ : Es la velocidad instantanea sobre el eje  $x$  del objeto en un periodo  $\Delta T$ .

$y_{V\{k\}}$ : Es la velocidad instantanea sobre el eje  $y$  del objeto en un periodo  $\Delta T$ .

Las ecuaciones que relacionan el cambio entre los estados del sistema de un periodo a otro, es decir, que permiten hacer la predicción del siguiente estado, en base a las ecuaciones 2.5 y 4.1 se expresa el modelo del sistema en la ecuación 4.2.

$$\hat{\mathbf{x}}_{\mathbf{k}}^- = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{P\{k-1\}} \\ y_{P\{k-1\}} \\ x_{V\{k-1\}} \\ y_{V\{k-1\}} \end{bmatrix} + \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \end{bmatrix} \quad (4.2)$$

La interpretación de la ecuación 4.2 es la siguiente:

$$x_{P\{k\}} = x_{P\{k-1\}} + x_{V\{k-1\}} + \xi_1$$

$$y_{P\{k\}} = y_{P\{k-1\}} + y_{V\{k-1\}} + \xi_2$$

$$x_{V\{k\}} = x_{V\{k-1\}} + \xi_3$$

$$y_{V\{k\}} = y_{V\{k-1\}} + \xi_4$$

En el instante  $k$  la posición es determinada por la posición en el tiempo  $k - 1$  más un desplazamiento ( $vt, t = 1$ ). Las unidades utilizadas para establecer la posición son  $cm$  y para la velocidad son  $cm/\Delta T$ .

$\begin{bmatrix} \xi_1 & \xi_2 & \xi_3 & \xi_4 \end{bmatrix}^T$  representa la incertidumbre que se le atribuye al modelo del sistema. Este es un vector aleatorio, gaussiano blanco de media cero y matriz de covarianza  $\mathbf{Q}$  (Ecuación 4.3). La determinación la covarianza del error del proceso  $\mathbf{Q}$ , es más

complicada que el cálculo de  $R$ , porque por lo general no se tiene acceso al proceso que se está estimando. De hecho, un modelo de sistema sencillo puede producir aceptables resultados si se le inyecta suficiente incertidumbre al proceso a través de la selección de  $Q$ .

$$\mathbf{Q} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} \quad (4.3)$$

- *Modelo de medida.* Es el segundo elemento del sistema de estimación. Se supone que cada periodo  $\Delta T$  se obtiene una medición ruidosa del vector de estados  $\mathbf{x}$ . La medida en este caso es la posición observada por las cámaras.  $\mathbf{z}_k = (x_{P\{k\}}, y_{P\{k\}})$ . La ecuación 4.4 representa al modelo de medida empleado.

$$\mathbf{z}_k = Hx_k + \mu \quad (4.4)$$

Donde:

$H$  es la matriz que relaciona la medición realizada sobre el ambiente con el vector de estado del sistema, definida en la ecuación 4.5:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.5)$$

Con lo que finalmente se obtiene el modelo de medida, ecuación 4.6.

$$\begin{bmatrix} z_k^x \\ z_k^y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{P\{k\}} \\ y_{P\{k\}} \\ x_{V\{k\}} \\ y_{V\{k\}} \end{bmatrix} + \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad (4.6)$$

La interpretación del modelo de medida está dada de la siguiente forma:

$$\begin{aligned} z_k^x &= x_{P\{k\}} + \mu_1 \\ z_k^y &= y_{P\{k\}} + \mu_2 \end{aligned}$$

El vector  $\mu$  representa la incertidumbre de la medición. Éste es aleatorio, gaussiano, blanco de media cero y matriz de covarianza  $\mathbf{R}$ . La covarianza del error de la medición  $R$  se puede medir antes de la operación del filtro. La medición de la covarianza del error de la medida

es generalmente posible, por lo regular se toman medidas de prueba con la finalidad de determinar la varianza del error de medición. Para el cálculo de este error, se tomaron mediciones de localización del  $OI$  ( $\hat{\mathbf{q}}_{\mathbf{u}}$ ) en diversos puntos del espacio, estos puntos de control se encuentran ubicados en posiciones definidas dentro del tablero que se utiliza para calibrar a la cámara. Una vez hechas las mediciones, se calcula la varianza del error al colocar los identificadores de los robots sobre estos puntos. Obteniéndose dos matrices de covarianza, correspondientes a los agentes considerados (Ecuaciones 4.7 y 4.8).

$$\mathbf{R}_1 = \begin{bmatrix} 1.19 & 0 & 0 & 0 \\ 0 & 1.12 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.7)$$

$$\mathbf{R}_2 = \begin{bmatrix} 1.26 & 0 & 0 & 0 \\ 0 & 3.32 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.8)$$

Las matrices  $\mathbf{R}_1$  y  $\mathbf{R}_2$ , son utilizadas para modelar la incertidumbre de las mediciones de las cámaras 1 y 2 respectivamente.

- Para la incertidumbre del estado inicial  $P_k^-$  se puede escoger casi cualquier valor para  $P_k^-$ , mientras no sea 0, lo que provocaría que no se actualizara el estado porque el predictor asumiría que no existe error, en este caso se escoge un valor arbitrario grande para que la ganancia de Kalman tenga mayor peso sobre la innovación del sistema. Se fijan los valores iniciales en la ecuación 4.9, la elección no es crítica, porque eventualmente el filtro convergerá.

$$\mathbf{P}_0 = \begin{bmatrix} 1000 & 0 & 0 & 0 \\ 0 & 1000 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1000 \end{bmatrix} \quad (4.9)$$

## 4.2. Correspondencia de coordenadas

La correspondencia entre coordenadas de los agentes se logra mediante el uso de un mismo sistema de referencia para los agentes cámara. De esta forma, las proyecciones en el espacio son referenciadas a un mismo sistema. Las diferencias que puedan existir

entre una medición de una cámara y otra son causadas por errores de calibración y de localización pertenecientes a cada agente cámara.

En las figuras 4.1 y 4.2 se presenta la posición de la cámara del agente cámara 1 y del agente cámara 2 utilizadas, respectivamente, para los experimentos.

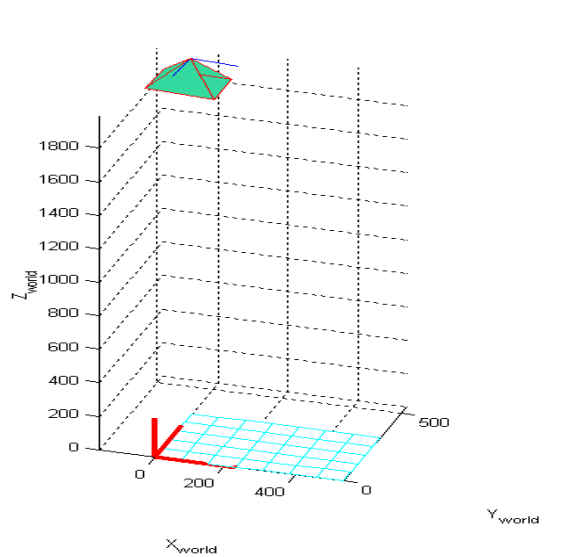


Figura 4.1: Sistema de Referencia de Agente Cámara 1.

Se observa la ubicación en el espacio de la cámara del agente cámara 1 con respecto al sistema referencial utilizado.

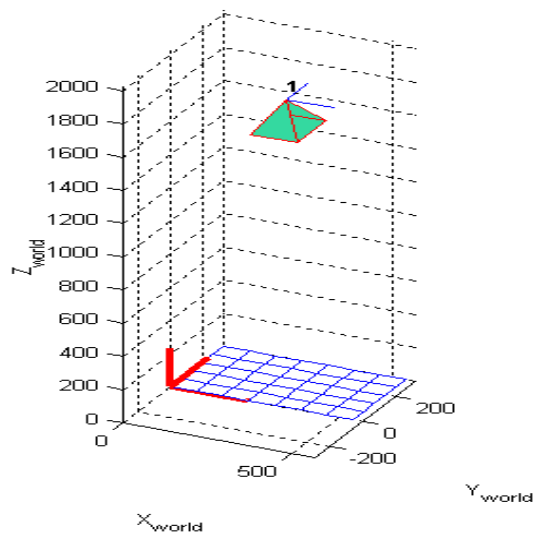


Figura 4.2: Sistema de Referencia de Agente Cámara 2.

Se observa la ubicación en el espacio de la cámara del agente cámara 2 con respecto al sistema referencial utilizado.

### 4.3. Medición de error de posición

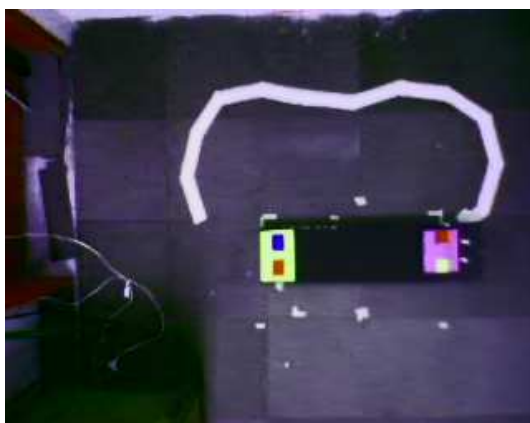
El error de posición se refiere a la distancia que existe entre la posición real del *OI* y la posición estimada por el seguidor. La posición real corresponde al punto central del *OI*. El cálculo del error se realiza mediante la distancia Euclidiana entre dos puntos, ecuación 4.10.

$$d = \sqrt{(x_p - x_r)^2 + (y_p - y_r)^2} \quad (4.10)$$

donde  $(x_p, y_p)$  representa la posición del *OI* considerada por el seguidor y  $(x_r, y_r)$  es la posición real.

### 4.4. Experimentos

Los experimentos se realizaron mediante el esquema de la arquitectura multiagente descrita, se utilizaron una y dos cámaras (agentes cámaras), para realizar el seguimiento de uno y dos objetos. Los objetos que se siguieron se montaron de tal forma que giraran alrededor de un punto base situado en  $(x, y) = (31cm, 27cm)$  con respecto al origen del sistema coordenado de referencia con radios de separación diferentes para cada objeto. Se utilizaron círculos con radios  $r_0 = 34cm$  y  $r_1 = 39cm$  para los objetos 0 y 1, respectivamente. El periodo de muestreo que se utilizó  $\Delta T$  fue de 200ms, lo cual equivale a poder hacer un análisis de 5 Frames por segundo, tanto para una como dos cámaras y en el seguimiento tanto de uno como de dos objetos. En la figura 4.3 se muestra un ejemplo del tipo de percepciones obtenidas por los agentes cámara durante los experimentos realizados.



(a) Percepción del Agente Cámara 1.



(b) Percepción del Agente Cámara 2.

Figura 4.3: Se presentan dos percepciones que los agentes cámara tomaron del ambiente durante los experimentos realizados.

Los experimentos que se realizaron se presentan a continuación, Figuras [4.4-4.10]. Primero se muestran para un agente cámara, tanto para el seguimiento de un objeto como de dos, y posteriormente se exponen para el caso de dos agentes cámara para analizar a uno y dos objetos. Con esto se trata de ejemplificar como funciona el sistema cuando se realiza el seguimiento de un objeto (*STT*) y para el caso de varios (*MTT*)

#### 4.4.1. Una cámara, Agente Cámara 1

##### Seguimiento de un objeto(*STT*)

Se muestra el comportamiento del sistema de seguimiento al utilizar un agente de tipo cámara, en este caso en particular, la cámara 1. Se puede observar en la figura 4.4 a la curva de color rojo, la cual indica el tipo de desplazamiento que el robot realiza. La curva negra indica la trayectoria que el agente cámara 1 percibe. Cabe mencionar que el desplazamiento del *OI* analizado no se desarrolla de forma continua ni a una velocidad constante. Donde se percibe una acumulación de puntos es porque el objeto se detuvo o redujo su velocidad, por otra parte donde se notan más espaciados estos puntos es porque el *OI* aumento su velocidad. El error promedio desarrollado por el

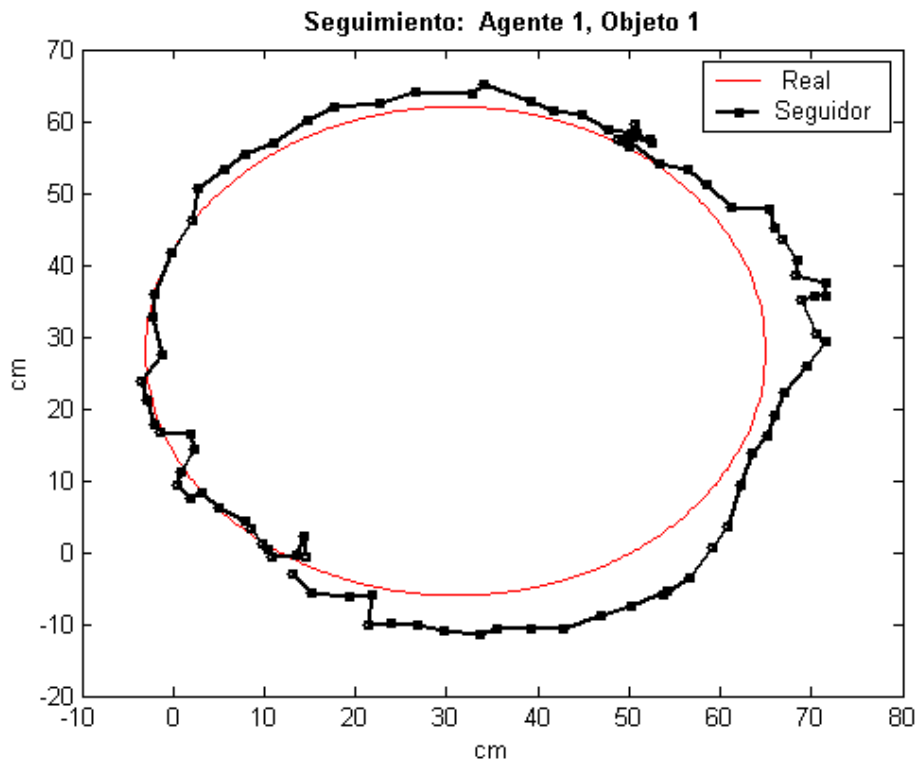


Figura 4.4: STT-Agente1.

Se presenta el seguimiento de un objeto a través del agente cámara 1. Se describe una trayectoria circular predefinida con radio  $r = 34\text{cm}$  centro de  $(x, y) = (31\text{cm}, 27\text{cm})$ .



### Seguimiento de dos objetos(MTT)

Una vez que se probó que el sistema de seguimiento podía realizar el análisis de un objeto en movimiento, el siguiente paso fue implementar experimentos con dos objetos, con el fin de verificar si el sistema tenía la capacidad de seguirlos y no perderlos. En la figura 4.5 se muestra el seguimiento de dos *OI* realizado por el agente cámara 1.

La relevancia de este experimento radica en mostrar que el sistema de seguimiento tiene la capacidad de analizar las trayectorias de dos objetos a la vez en el mismo periodo de tiempo.

Las características de las curvas presentes en la figura 4.5 son similares a las descritas en el experimento anterior. Se pueden observar intervalos donde existe una gran acumulación de puntos, esto indica que los *OIs* se detuvieron en ese lapso de la trayectoria. Los errores promedio que presentan los seguidores para los dos *OIs* a lo largo de su trayectoria son  $\bar{e}_1 = 2.96$  para el movimiento que describe el *OI1* y  $\bar{e}_2 = 3.08$  para el de *OI2*.

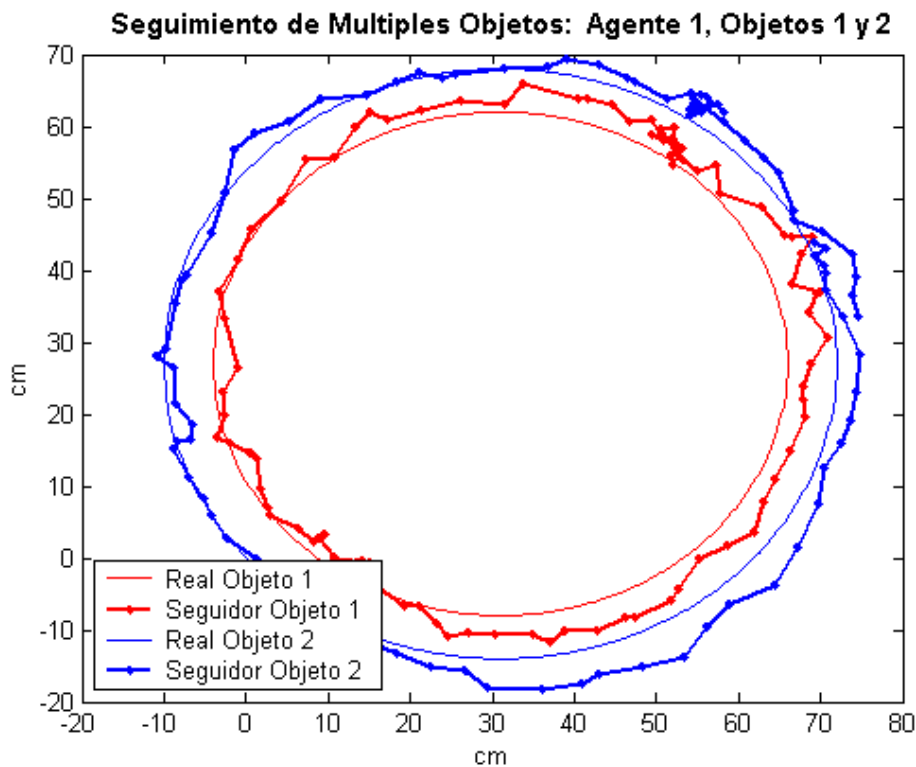


Figura 4.5: MTT-Agente1.

Se muestra el seguimiento de dos objetos por el agente cámara 1, ambos con trayectorias circulares centradas a  $(x, y) = (31\text{cm}, 27\text{cm})$ , sólo que uno con radio  $r_0 = 34\text{cm}$  y el otro con  $r_1 = 39\text{cm}$ .

#### 4.4.2. Una cámara, Agente Cámara 2

Se hicieron experimentos con el agente cámara 2 con la finalidad de verificar el comportamiento del sistema de seguimiento desde la ubicación de la cámara 2. Los objetos percibidos describieron el mismo tipo de trayectorias que en la subsección anterior, aunque con inicio del desplazamiento en distintos puntos y variaciones de la velocidad de de movimiento arbitrarias. En primera instancia se siguió a un sólo objeto(Figura 4.6) y posterior a eso se hicieron pruebas con el seguimiento de dos objetos 4.7.

##### Seguimiento de un objeto(*STT*)

Se puede observar que el seguimiento del objeto 1 realizado por el agente cámara 2 presenta mayor diferencia entre las trayectoria percibida y la real, lo que denota mayor error promedio ( $\bar{\epsilon}_1 = 3.45$ ) en el seguimiento que el desarrollado por el agente cámara

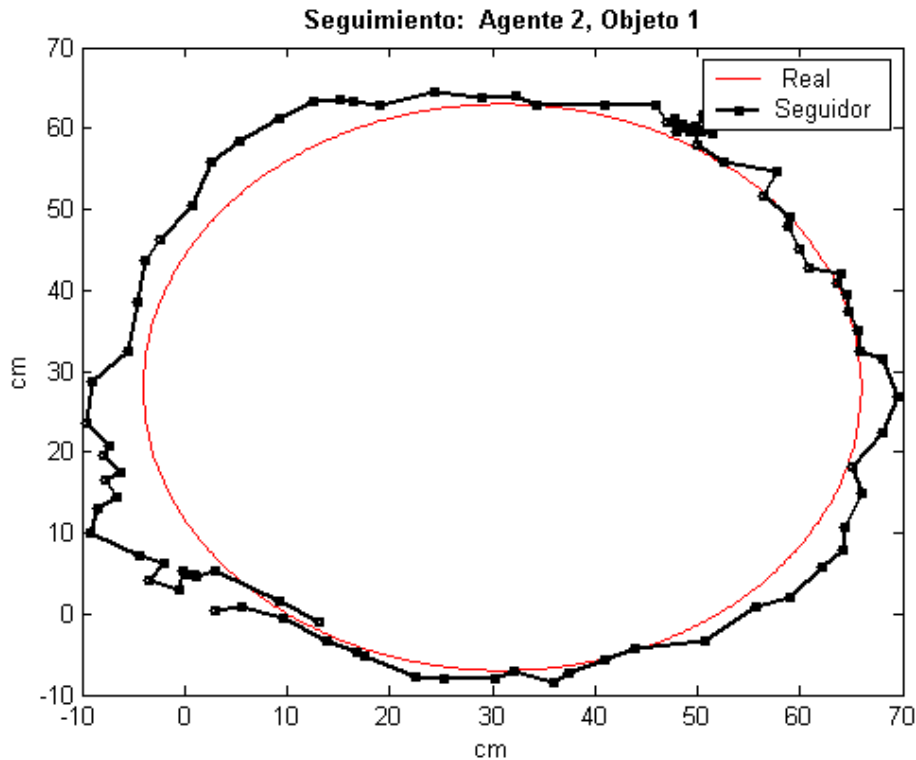


Figura 4.6: STT-Agente 2.

Se muestra el seguimiento de un objeto a través del agente cámara 2. Se describe una trayectoria circular predefinida con radio  $r = 34\text{cm}$  centro de  $(x, y) = (31\text{cm}, 27\text{cm})$  con respecto al sistema de referencia.

### Seguimiento de dos objetos(*MTT*)

El seguimiento de dos objetos de interés realizado por el agente cámara 2 presenta un error promedio de  $\bar{e}_1 = 3.31$  y  $\bar{e}_2 = 3.22$  para el objeto de interés 1 y 2 respectivamente, estos errores son un poco mayores a los mostrados por el agente cámara 1.

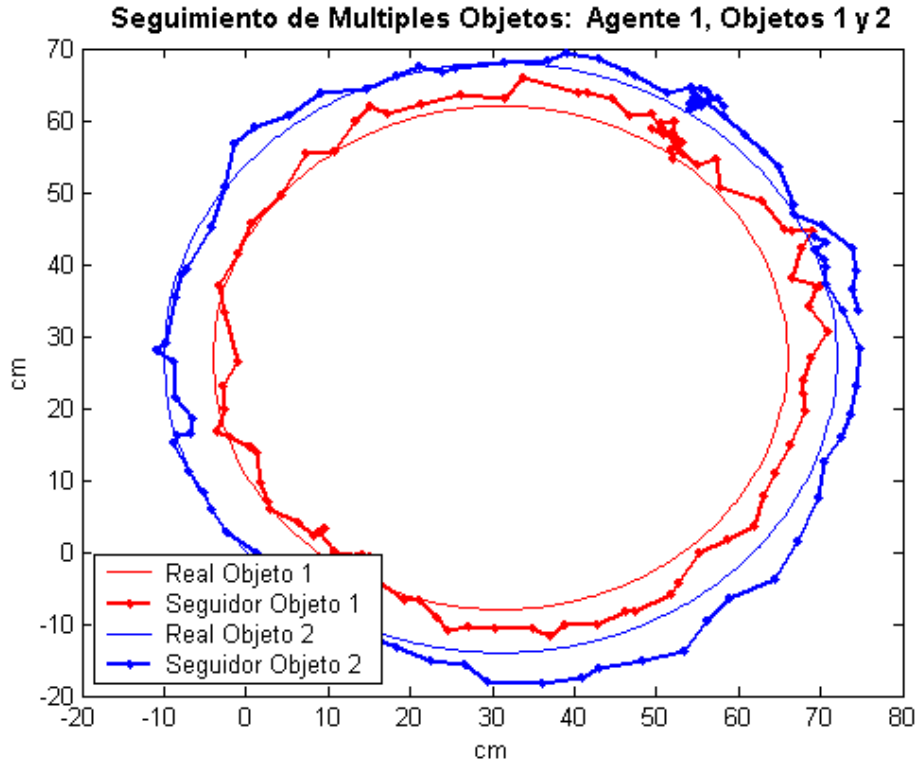


Figura 4.7: *MTT*-Agente2.

Se muestra la percepción del agente cámara 2 en el seguimiento de dos objetos, ambos con trayectorias circulares centradas a  $(x, y) = (31\text{cm}, 27\text{cm})$  con respecto al sistema de referencia, sólo que uno con radio  $r_0 = 34\text{cm}$  y el otro con  $r_1 = 39\text{cm}$ .

Hasta el momento se han realizado experimentos con los agentes cámara 1 y 2 de forma independiente, pero no de forma simultánea, para seguir a uno y dos objetos de interés. El desempeño de los seguidores para uno y varios objetos ha sido satisfactorio al mostrar errores promedio de 2-3.5 cm. Sin embargo, es necesario probar si se puede realizar el seguimiento del mismo número de objetos a través de dos cámaras, con la finalidad de extender de forma efectiva la superficie de sensado y proporcionar redundancia en el seguimiento de los objetos de interés.

#### 4.4.3. Dos cámaras, Agentes Cámara 1 y 2

Posteriormente, se hicieron pruebas con dos agentes cámara para obtener el análisis de forma simultánea del espacio. La ventaja de contar con dos agentes cámara es

que se puede observar una área de mayores dimensiones en conjunto, así como mejorar la localización de los *OI* en el espacio mediante la fusión de datos de mediciones correspondientes entre sí.

Enseguida se muestran los experimentos que se realizaron mediante el uso de dos agentes cámara en el sistema de seguimiento. Es necesario mencionar que en las figuras 4.9 y 4.10 en la sección de seguimiento *MTT* para dos agentes cámara se muestra por separado lo que cada agente cámara está analizando para fines de no saturación de las gráficas.

### Seguimiento de un objeto(*STT*)

Uno de los objetivos de contar con varios agentes cámara que analicen el espacio, es tener redundancia de mediciones de un mismo *OI*, lo cual nos da la posibilidad de mejorar la localización de éste a través de la fusión de datos de la información de varios sensores. En la figura 4.8 se muestran las trayectorias percibidas de los agentes cámaras 1 y 2 sobre el desplazamiento de un mismo objeto.

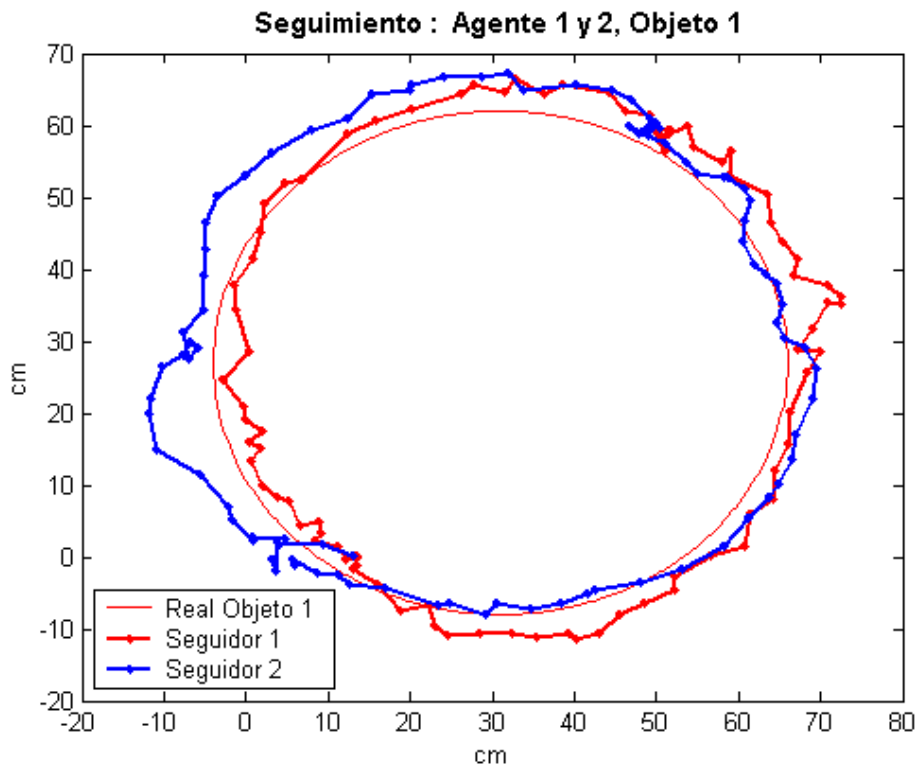


Figura 4.8: Agente 1 y 2-STT.

Se muestra la percepción realizada por el agente cámara 1 y el agente cámara 2 sobre un mismo objeto que se desplaza sobre una trayectoria circular centrada en  $(x, y) = (31\text{cm}, 27\text{cm})$  con respecto al sistema de referencia, con radio  $r_0 = 34\text{cm}$ .

Las áreas de interés de los agentes cámara 1 y 2 cuentan con cierto traslape, dentro de éste un objeto de interés se mueve a lo largo de su trayectoria circular. Los errores promedio que presentan los agentes cámara 1 y 2 en el seguimiento de los objetos son de  $\bar{e}_1 = 3.25$  y  $\bar{e}_2 = 3.71$  respectivamente. En la figura 4.8 se puede observar que hay mayor error en el seguidor correspondiente al agente 2, tal como queda evidenciado por el mayor error promedio presentado por este mismo. Se observa que las trayectorias son parecidas, la diferencia que existe entre ellas se debe al error que cada agente cámara tiene al localizar en el espacio a un *OI*. La incorporación de la fusión de datos en casos de redundancia de información concerniente a un mismo *OI* permitirá la mejora de su localización en el espacio.

### Seguimiento de dos objetos ( *MTT* )

Se extendió el experimento anterior con la finalidad de que dos agentes sigan a dos objetos figuras 4.9 y 4.10. Este experimento es importante porque la capacidad de computo necesaria es considerable debido a que se analizan dos cámaras y se siguen a dos objetos al mismo tiempo en una sola computadora.

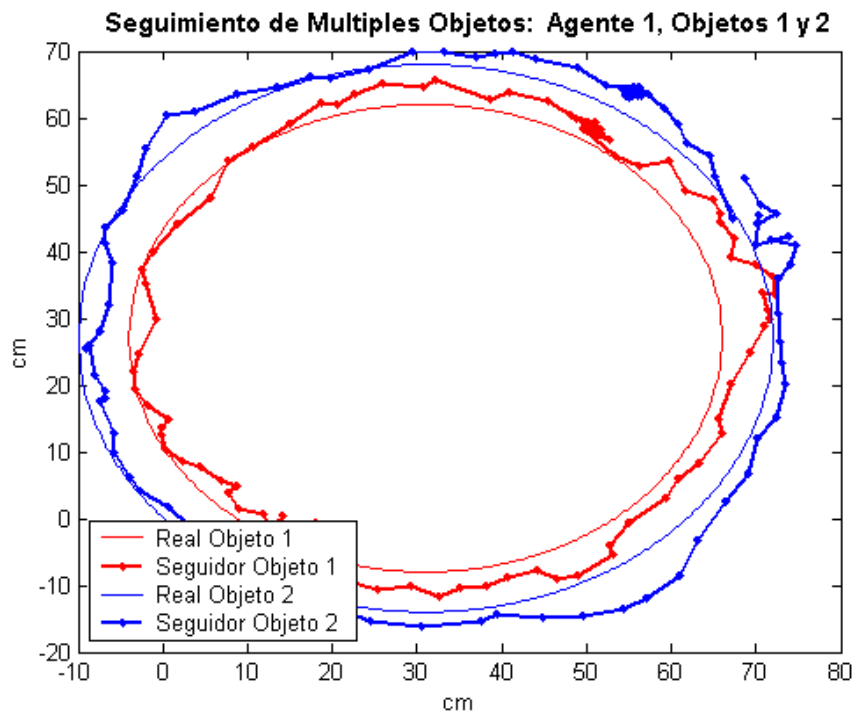


Figura 4.9: Agente 1 y 2-MTT.

Se observa el análisis realizado por el agente cámara 1 en el seguimiento de dos objetos que se desplazan en trayectorias circulares centradas en  $(x, y) = (31\text{cm}, 27\text{cm})$ , con radios de  $r_0 = 34\text{cm}$  y  $r_0 = 39\text{cm}$ .

El desempeño del seguimiento no se decrementa, el periodo de  $\Delta T$  sigue siendo

suficiente para evaluar las trayectorias de los dos objetos, que podrían ser considerados como si fueran cuatro en caso de que el área donde se estuviesen desarrollando no fuera de traslape. Los errores promedio presentados por los agentes cámara 1 y 2 para el seguimiento del objeto de interés 1 son  $\bar{e}_1 = 3.0$  y  $\bar{e}_1 = 3.29$  respectivamente. Con respecto al objeto de interés 2 son  $\bar{e}_1 = 3.13$  (Agente Cámara 1) y  $\bar{e}_1 = 3.27$  (Agente Cámara 2).

Los experimentos realizados en esta sección son relevantes porque se tenía que verificar si la arquitectura de múltiples seguidores se comporta de forma similar cuando se tiene más de una cámara. De esta forma, después de constatar que el error promedio no aumenta se comienza a verificar el desempeño de la fusión de datos y el esquema de seguimiento intercámara.

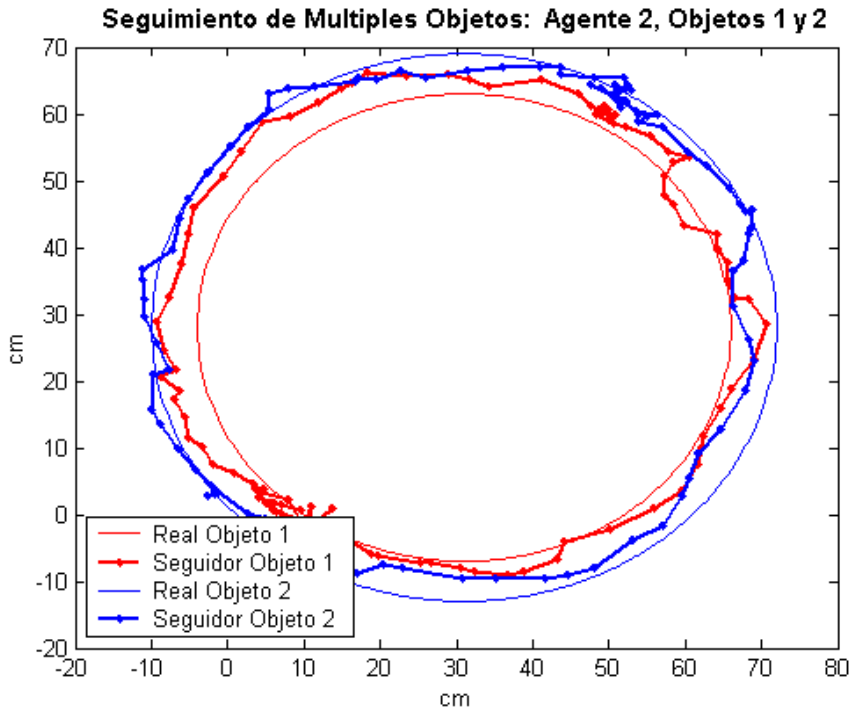


Figura 4.10: Agente 1 y 2-MTT.

Se observa el análisis realizado por el agente cámara 2 en el seguimiento de dos objetos que se desplazan en trayectorias circulares centradas en  $(x, y) = (31\text{cm}, 27\text{cm})$ , con radios de  $r_0 = 34\text{cm}$  y  $r_0 = 39\text{cm}$ . Este proceso de seguimiento es realizado por el agente cámara 1 y 2 de forma simultánea, asimismo como los dos agentes cámara proporcionan sus estimaciones, el agente de control las fusiona en orden de obtener una nueva estimación.

## 4.5. Experimentos de Fusión de datos mediante la arquitectura multiagente propuesta

Se efectuaron diversos experimentos para medir el error de seguimiento sin combinar información, y al fusionar datos. Se presentan tres series de tablas que son características de estos experimentos, cada serie está compuesta de dos tablas una contiene el desempeño de la información sin fusionar y la segunda muestra el desempeño de la fusión de datos mediante el Filtro de Kalman y mediante un promedio. Las tres series de tablas corresponden a los experimentos que se realizaron para diferentes magnitudes en las ventanas de tiempo; las ventanas de tiempo contempladas son de 100ms (Tablas [4.1 y 4.2]), 150ms (Tablas [4.3 y 4.4]) y 200ms (Tablas [4.5 y 4.6]). No se contemplaron otras magnitudes para las ventanas porque para magnitudes menores a 100ms los casos donde se realizaba la fusión de datos eran muy pocos. Por otra parte, si se aumenta la ventana por arriba de 200ms surge la posibilidad de fusionar estimaciones que tengan una gran diferencia entre el tiempo en que se realizaron una de la otra.

La información contenida en las tablas de datos sin combinar (Tablas [4.1, 4.3 y 4.5]) y las de fusión de datos es la siguiente (Tablas [4.2, 4.4 y 4.6]):

- *No*. Índice de dato, que sirve para referenciar y comparar registro a registro entre los datos fusionados y los no fusionados.
- *Agente B Proyectado*. Agrupa la información correspondiente al agente cámara(B) que está observando al mismo objeto que el agente cámara(A); este último es el que está proporcionando en ese instante la información al agente de control. Por lo tanto, el agente de control proyecta los datos de seguimiento del agente B, para ajustar en tiempo a las dos estimaciones.
- *Agente B*. Es el agente cámara considerado por el agente de control para realizar la fusión de datos. La información mostrada en este campo aún no ha sido proyectada.
- *Agente A*. Es el agente que solicita la actualización del *pizarrón*. Los experimentos fueron realizados considerando dos agentes cámara, por lo que estos agentes son diferenciados como Agente  $A_1$  y  $A_2$  que representan a los agentes cámara 1 y 2 respectivamente. Se presenta esta división en los conjuntos de datos de las tablas para poder apreciar los errores de los datos de forma separada al ser fusionados con los datos del agente B proyectado.
- *Filtro de Kalman*. Son los datos fusionados mediante el Filtro de Kalman.
- *Promedio*. Son los datos fusionados mediante un promedio entre las estimaciones del agente A y el agente B (sin proyectar)

El error es la distancia euclidiana de las estimaciones a la trayectoria real. De forma particular para los experimentos, es la distancia de la estimación realizada por el agente cámara a la circunferencia que describe el movimiento del objeto de interés a lo largo de su trayectoria. Al final de cada campo de estimaciones se muestra el error promedio desarrollado a lo largo del movimiento de los objetos de interés. Se observa que para todas las ventanas (100, 150 y 200ms) de fusión de datos, el Filtro de Kalman presenta mejor desempeño que la fusión mediante un promedio. Asimismo, se observa que el filtro de Kalman presenta una magnitud de error similar a la estimación del agente cámara con menor incertidumbre (Agente Cámara 1), debido a que el filtro de Kalman al momento de realizar la fusión de datos confía menos en la estimación realizada por el agente Cámara 2 por la mayor incertidumbre que presenta sobre sus estimaciones.

En la tabla 4.1 se observa que el agente cámara solicitante de actualización del pizarrón que es considerado en el proceso de fusión de datos siempre es el mismo, de forma particular para los experimentos realizados en esta investigación es el agente cámara 2; esto es consecuencia del tamaño de la ventana (100ms) el cual restringe posibles intercalados entre las actualizaciones al momento de fusionar los datos.

En las tablas 4.3 y 4.5 se observa que existe un mayor intercalado entre los agentes solicitantes de actualización del pizarrón que son considerados en el proceso de fusión de datos por el agente de control. Ésto es ocasionado porque las ventanas son más grandes, provocando un intercalado constante cuando es de 200ms.

La estimación proyectada del agente B presentó menor error que las estimaciones del agente B sin proyectar en todos los experimentos, a través de esta proyección se trata de disminuir el error de la fusión de datos resultante al mejorar una de las dos mediciones a combinar.

De forma general, el filtro de Kalman presentó mejor desempeño que la combinación de información mediante el promedio. Además se comportó mejor que el agente cámara con mayor incertidumbre (Cámara 2) en su estimación de posición y de forma similar al agente con menor incertidumbre en sus estimaciones (Cámara 1), lo cual era de esperarse porque el filtro de Kalman combina los datos mediante el uso de las incertidumbres de cada sensor dando mayor importancia a la fuente de información con menor incertidumbre.



Cuadro 4.1: Datos en una ventana de 100ms.

Se muestran las estimaciones de dos agentes (A y B), se observa que el agente cámara que solicita la actualización cuando se realiza la fusión de datos es el mismo durante todo el proceso. La estimación del agente B proyectado exhibe un menor error que la del agente B sin proyectar, esto es debido a que se estima la posición del objeto donde se debería encontrar en el instante que el agente a solicita la actualización. La estimación que presenta mayor error es la del agente cámara 2 (A).

No.	<i>Agente B</i>			<i>Agente B Proyectado</i>			<i>Agente A</i>		
	x	y	error	x	y	error	x	y	error
1	10.55	3.12	2.56	9.96	3.47	2.44	-0.14	6.10	3.50
2	8.89	2.67	1.12	9.07	4.91	2.87	0.54	6.11	2.94
3	7.79	3.99	1.32	7.62	6.22	2.72	-0.63	7.40	3.21
4	7.32	6.33	2.57	5.44	9.02	2.75	-2.62	8.37	4.44
5	6.10	10.25	3.99	4.63	11.09	3.20	-3.42	9.28	4.71
6	5.32	12.36	4.44	2.45	10.74	1.14	-4.81	12.20	4.75
7	1.60	10.38	0.23	0.93	12.12	0.45	-5.69	14.75	4.68
8	-0.51	11.18	1.26	0.48	13.15	0.48	-8.05	16.28	6.49
9	0.29	13.34	0.39	2.16	15.05	2.78	-8.33	19.09	6.12
10	2.49	15.59	3.29	3.03	16.62	4.17	-9.81	21.88	7.13
11	2.93	16.70	4.10	0.92	17.79	2.54	-7.89	23.88	5.01
12	0.82	18.05	2.52	-1.30	21.65	1.26	-8.34	26.97	5.34
13	-1.07	22.60	1.63	-2.26	24.77	0.67	-8.60	31.32	5.83
14	-1.72	26.06	1.27	-1.26	28.97	1.68	-7.33	35.10	5.18
15	-0.78	30.16	2.06	-1.51	34.51	0.63	-6.10	40.10	5.34
16	-2.44	34.41	0.25	-1.89	38.66	0.90	-5.29	44.81	6.42
17	-0.83	40.48	0.57	0.49	42.82	0.37	-5.41	48.53	8.30
18	0.51	43.48	0.66	1.53	47.22	1.74	-3.18	52.03	8.36
19	2.48	48.84	1.92	3.92	50.23	1.68	0.49	55.28	7.60
20	4.29	50.98	1.90	7.13	54.32	2.28	6.31	59.67	6.95
21	6.57	54.15	2.52	11.02	55.73	0.99	8.92	61.54	6.99
22	12.39	57.17	1.45	14.20	59.41	2.51	12.14	64.29	7.79
23	15.50	60.80	3.18	17.60	61.29	2.82	14.89	66.09	8.28
24	18.80	62.50	3.54	19.89	62.58	3.27	19.64	67.38	7.95
25	20.33	62.90	3.45	23.12	62.68	2.54	19.39	66.95	7.60
26	24.17	63.41	3.05	28.22	64.64	3.74	23.81	67.11	6.75
27	28.07	63.98	3.10	32.36	63.83	2.86	28.39	67.48	6.56
28	32.50	63.28	2.31	33.25	64.53	3.60	32.49	66.77	5.80
29	34.09	64.85	3.98	39.47	62.35	2.35	34.62	66.57	5.74
30	39.58	61.51	1.56	42.39	64.44	5.13	41.61	65.99	6.41
	<i>Promedio</i>		2.21	<i>Promedio</i>		2.10	<i>Promedio</i>		6.07

Cuadro 4.2: Fusión de datos para una ventana de 100ms.  
 Se observa que la fusión de datos realizada por el Filtro de Kalman presenta mejor desempeño que el promedio de las dos estimaciones de los agentes cámara.

Fusión						
	<i>Filtro de Kalman</i>			<i>Promedio</i>		
No.	x	y	error	x	y	error
1	4.85	4.24	0.67	5.21	4.61	0.16
2	4.97	4.12	0.66	4.72	4.39	0.67
3	3.32	5.12	1.28	3.58	5.70	0.72
4	2.28	7.18	0.90	2.35	7.35	0.74
5	0.96	9.99	0.52	1.34	9.77	0.30
6	0.47	12.69	0.28	0.26	12.28	0.09
7	-2.55	11.62	2.91	-2.05	12.57	2.06
8	-4.51	12.81	4.24	-4.28	13.73	3.69
9	-4.47	15.07	3.42	-4.02	16.22	2.64
10	-4.13	17.57	2.37	-3.66	18.74	1.63
11	-2.63	19.09	0.55	-2.48	20.29	0.15
12	-4.66	20.51	2.25	-3.76	22.51	1.05
13	-4.54	25.63	1.57	-4.84	26.96	1.84
14	-4.34	29.07	1.40	-4.53	30.58	1.70
15	-3.43	33.34	1.01	-3.44	35.13	1.39
16	-3.97	37.60	2.54	-3.87	39.61	3.08
17	-3.37	42.92	3.88	-3.12	44.51	4.35
18	-1.18	46.28	3.51	-1.34	47.76	4.42
19	0.74	50.44	4.28	1.49	52.06	4.72
20	5.61	53.76	2.89	5.30	55.33	4.25
21	7.56	56.25	3.48	7.75	57.85	4.63
22	11.91	59.16	3.40	12.27	60.73	4.58
23	15.24	62.45	4.80	15.20	63.45	5.72
24	19.17	63.95	4.80	19.22	64.94	5.73
25	20.00	64.24	4.83	19.86	64.93	5.53
26	23.53	64.29	4.03	23.99	65.26	4.90
27	28.84	65.38	4.44	28.23	65.73	4.83
28	32.67	64.45	3.49	32.50	65.03	4.05
29	34.55	65.48	4.64	34.36	65.71	4.86
30	40.34	62.68	2.88	40.60	63.75	3.98
	<i>Promedio</i>		2.73	<i>Promedio</i>		2.95

Cuadro 4.3: Datos en una ventana de 150ms.

Se observa intercalado entre los agentes solicitantes de actualización que son considerados en la fusión de datos, esto es debido a que la ventana de tiempo es de 150 ms, lo cual permite que datos más espaciados en el tiempo puedan ser fusionados. La estimación del agente B proyectado exhibe un menor error que la del agente B sin proyectar. Además el agente cámara 2 ( $A_2$ ) presenta mayor error que el agente cámara 1 ( $A_1$ ).

No.	<i>Agente B</i>			<i>Agente B Proyectado</i>			<i>Agente A<sub>1</sub></i>			<i>Agente A<sub>2</sub></i>		
	x	y	error	x	y	error	x	y	error	x	y	error
1	9.66	3.65	2.37	7.12	2.02	0.56				-3.11	3.08	7.66
2	6.04	1.09	1.98	6.51	4.65	0.84				-2.06	5.60	5.38
3	5.16	3.68	0.81	3.59	7.68	0.47				-3.04	7.66	5.15
4	3.13	7.95	0.24	2.29	8.60	0.10				-3.35	11.06	3.87
5	2.86	9.78	1.01	0.42	10.62	0.69				-3.73	13.16	3.39
6	0.91	11.88	0.32	0.57	11.85	0.01				-4.06	15.85	2.79
7	-0.12	11.58	0.73	2.32	13.62	2.35				-5.47	16.25	4.02
8	1.19	12.72	0.95	2.67	15.65	3.48				-5.68	19.22	3.50
9	2.66	15.89	3.56	3.61	17.32	4.95				-7.43	20.69	4.94
10	3.87	17.64	5.30	0.27	18.97	2.24				-7.08	22.38	4.36
11	0.01	18.93	1.98	-0.49	19.73	1.68				-9.49	24.67	6.56
12	-0.59	19.87	1.62	-1.81	23.22	0.97				-10.00	29.89	7.10
13	-1.50	24.02	1.36	-0.20	26.98	2.80				-7.98	32.21	5.33
14	0.57	27.90	3.56	-1.73	34.44	0.44				-5.28	36.26	3.44
15	-1.23	35.44	0.68	-2.00	39.10	1.15				-4.81	42.52	5.03
16	-5.70	41.96	5.63	-0.23	46.93	3.05	0.01	41.60	0.26			
17	1.10	43.06	0.06	2.78	45.03	0.51				1.15	50.76	4.15
18	1.17	51.05	4.32	1.75	54.72	6.30	6.25	48.76	1.04			
19	2.67	55.76	6.37	6.44	59.54	6.77	8.79	54.24	1.15			
20	6.65	59.83	6.87	9.04	62.91	8.09	10.93	54.91	0.38			
21	8.33	62.30	7.95	12.38	64.37	7.75	13.40	59.34	2.82			
22	13.52	65.50	8.28	16.15	65.50	7.26	15.61	61.16	3.47			
23	17.26	66.51	7.83	19.26	65.10	5.87	17.41	61.46	3.04			
24	20.06	65.82	6.33	19.37	65.00	5.74	20.86	63.62	4.00			
25	22.72	66.80	6.65	28.57	67.68	6.75	31.02	63.18	2.18			
26	32.79	67.52	6.56	33.55	65.67	4.75	38.69	64.27	4.06			
27	32.95	64.75	3.80	39.40	66.17	6.06	40.81	64.49	4.75			
28	39.25	65.61	5.48	44.20	65.62	6.81	46.27	63.54	5.60			
29	44.87	65.92	7.32	46.45	64.45	6.51	48.58	59.72	3.14			
30	46.18	63.67	5.69	48.08	61.95	4.90	50.45	59.63	3.99			
	<i>Promedio</i>		3.85	<i>Promedio</i>		3.66	<i>Promedio</i>		2.84	<i>Promedio</i>		4.79

Cuadro 4.4: Fusión de datos en una ventana de 150ms.

Se observa que la fusión de datos realizada por el Filtro de Kalman presenta mejor desempeño que el promedio de las dos estimaciones de los agentes cámara. El desempeño de la estimación fusionada mediante el filtro de Kalman presenta menor error que las demás estimaciones consideradas.

Fusión						
	<i>Filtro de Kalman</i>			<i>Promedio</i>		
No.	x	y	error	x	y	error
1	2.92	3.64	2.53	3.28	3.37	2.43
2	2.28	2.85	3.52	1.99	3.35	3.43
3	0.82	4.98	3.36	1.06	5.67	2.76
4	-0.02	9.12	1.80	-0.11	9.51	1.69
5	-0.68	10.84	1.56	-0.44	11.47	1.06
6	-1.12	13.48	0.85	-1.58	13.87	1.12
7	-3.21	12.91	3.00	-2.80	13.92	2.24
8	-2.44	14.78	1.60	-2.25	15.97	1.03
9	-2.90	17.33	1.25	-2.39	18.29	0.50
10	-2.01	19.15	0.07	-1.61	20.01	0.65
11	-4.83	20.88	2.35	-4.74	21.80	2.12
12	-6.21	22.66	3.46	-5.30	24.88	2.36
13	-4.40	26.88	1.40	-4.74	28.12	1.76
14	-2.19	30.70	0.60	-2.36	32.08	0.26
15	-2.93	37.75	1.59	-3.02	38.98	2.07
16	-2.43	41.73	2.53	-2.85	41.78	2.93
17	1.09	45.39	1.11	1.13	46.91	1.90
18	4.38	49.82	1.06	3.71	49.91	1.63
19	5.51	53.86	3.03	5.73	55.00	3.72
20	9.30	56.57	2.68	8.79	57.37	3.62
21	10.92	59.81	4.47	10.87	60.82	5.36
22	14.37	61.97	4.72	14.57	63.33	5.87
23	17.42	62.98	4.46	17.34	63.99	5.43
24	20.43	64.13	4.61	20.46	64.72	5.16
25	27.61	64.42	3.57	26.87	64.99	4.21
26	35.71	64.64	3.93	35.74	65.90	5.18
27	38.00	65.30	4.93	36.88	64.62	4.08
28	43.41	64.34	5.35	42.76	64.58	5.37
29	47.15	61.68	4.26	46.73	62.82	5.12
30	48.26	60.33	3.53	48.32	61.65	4.74
	<i>Promedio</i>		2.77	<i>Promedio</i>		2.99

Cuadro 4.5: Datos en una ventana de 200ms.

Se observa un intercalado completo entre los agentes solicitantes de actualización que son considerados en la fusión de datos, esto es debido a que la ventana de tiempo es de 200 ms, lo cual permite que la fusión de datos se realice cada vez que un agente cámara solicita la actualización del pizarrón. La estimación del agente B proyectado exhibe un menor error que la del agente B sin proyectar. Además el agente cámara 2 ( $A_2$ ) presenta mayor error que el agente cámara 1 ( $A_1$ ).

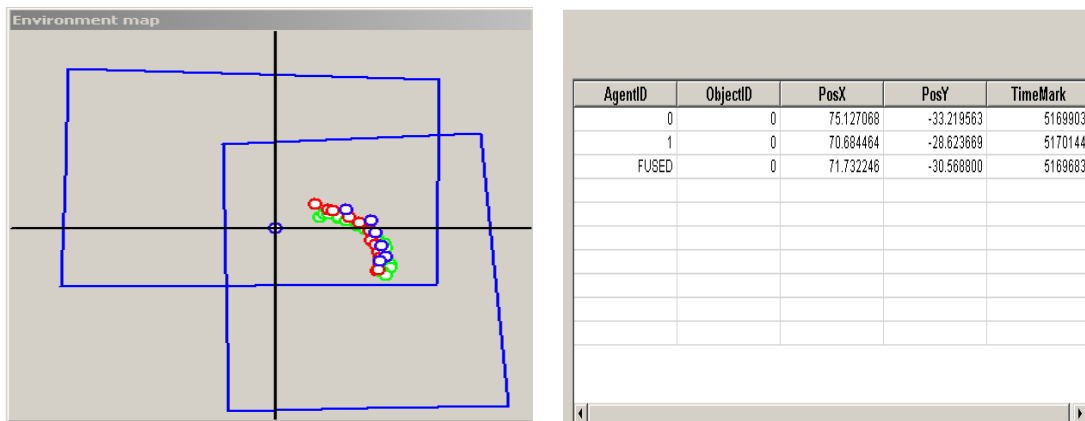
No.	Agente B			Agente B Proyectado			Agente $A_1$			Agente $A_2$		
	x	y	error	x	y	error	x	y	error	x	y	error
1	-1.62	17.23	17.31	-2.95	18.14	18.38				-7.47	23.75	24.90
2	-9.86	24.87	26.75	-7.47	23.75	24.90	-1.91	20.15	20.24			
3	-3.71	18.83	19.19	-1.91	20.15	20.24				-8.98	26.92	28.38
4	-7.02	25.70	26.64	-8.98	26.92	28.38	-2.65	24.85	24.99			
5	-3.11	19.96	20.20	-2.65	24.85	24.99				-8.12	32.40	33.40
6	-10.08	28.10	29.85	-8.12	32.40	33.40	-1.65	29.68	29.73			
7	-2.60	26.24	26.37	-1.65	29.68	29.73				-6.17	36.75	37.26
8	-8.84	33.95	35.08	-6.17	36.75	37.26	-0.84	35.57	35.58			
9	-1.34	31.61	31.64	-0.84	35.57	35.58				-4.58	41.60	41.85
10	-4.82	40.22	40.51	-4.58	41.60	41.85	0.12	38.37	38.37			
11	-0.32	37.51	37.51	0.12	38.37	38.37				-3.14	45.03	45.14
12	-2.96	45.79	45.89	-3.14	45.03	45.14	1.35	42.07	42.09			
13	0.60	39.88	39.88	1.35	42.07	42.09				-0.49	48.97	48.97
14	-1.84	48.54	48.57	-0.49	48.97	48.97	2.68	46.25	46.33			
15	0.92	42.90	42.91	2.68	46.25	46.33				-0.04	52.07	52.07
16	0.78	52.39	52.40	-0.04	52.07	52.07	5.55	49.06	49.37			
17	3.53	48.25	48.38	5.55	49.06	49.37				0.65	54.90	54.90
18	1.07	55.60	55.61	0.65	54.90	54.90	7.25	52.88	53.37			
19	7.02	51.44	51.92	7.25	52.88	53.37				1.95	57.68	57.71
20	2.24	59.03	59.07	1.95	57.68	57.71	11.08	55.71	56.80			
21	6.81	53.31	53.74	11.08	55.71	56.80				5.22	59.88	60.11
22	3.06	60.60	60.68	5.22	59.88	60.11	14.06	59.14	60.79			
23	11.31	56.46	57.58	14.06	59.14	60.79				11.08	62.61	63.58
24	6.79	62.06	62.43	11.08	62.61	63.58	16.44	60.59	62.78			
25	14.21	59.54	61.21	16.44	60.59	62.78				14.93	63.20	64.94
26	13.13	64.21	65.54	14.93	63.20	64.94	18.63	61.54	64.30			
27	18.05	62.40	64.96	18.63	61.54	64.30				19.17	64.58	67.37
28	17.64	64.89	67.24	19.17	64.58	67.37	24.48	64.21	68.72			
29	20.61	63.40	66.67	24.48	64.21	68.72				19.44	64.93	67.78
30	21.46	65.54	68.96	19.44	64.93	67.78	26.25	64.93	70.04			
31	26.26	65.65	70.71	26.25	64.93	70.04				23.48	67.27	71.25
32	21.87	66.37	69.88	23.48	67.27	71.25	30.94	62.88	70.08			
33	27.62	66.09	71.63	30.94	62.88	70.08				28.73	67.99	73.81
34	24.71	67.48	71.86	28.73	67.99	73.81	34.89	65.64	74.34			
35	32.57	63.18	71.08	34.89	65.64	74.34				31.08	68.04	74.80
36	32.22	69.39	76.51	31.08	68.04	74.80	42.08	64.25	76.80			
37	37.55	67.20	76.98	42.08	64.25	76.80				33.08	67.12	74.83
38	32.85	68.24	75.74	33.08	67.12	74.83	45.81	63.30	78.14			
39	43.44	63.65	77.06	45.81	63.30	78.14				39.93	66.81	77.83
40	34.42	66.38	74.77	39.93	66.81	77.83	48.78	62.32	79.14			
	<i>Promedio</i>		4.37	<i>Promedio</i>		3.96	<i>Promedio</i>		2.05	<i>Promedio</i>		6.08

Cuadro 4.6: Fusión de datos para una ventana de 200ms.  
 Se observa que la fusión de datos realizada por el Filtro de Kalman presenta mejor desempeño que el promedio de las dos estimaciones de los agentes cámara. El desempeño de la estimación fusionada mediante el filtro de Kalman presenta menor error que la estimación proyectada del agente B.

Fusión						
	<i>Filtro de Kalman</i>			<i>Promedio</i>		
No.	x	y	error	x	y	error
1	-4.74	19.26	2.57	-4.55	20.49	2.14
2	-5.43	21.38	2.86	-5.89	22.51	3.16
3	-6.41	21.40	3.83	-6.35	22.88	3.57
4	-4.97	24.54	2.05	-4.84	25.28	1.88
5	-5.25	24.10	2.37	-5.62	26.18	2.62
6	-5.12	29.45	2.20	-5.87	28.89	2.91
7	-4.38	29.55	1.47	-4.39	31.50	1.67
8	-4.64	34.67	2.46	-4.84	34.76	2.67
9	-2.78	34.85	0.68	-2.96	36.61	1.29
10	-1.46	39.60	0.82	-2.35	39.30	1.54
11	-2.13	39.66	1.47	-1.73	41.27	1.71
12	-1.08	42.41	1.59	-0.81	43.93	2.03
13	-0.04	42.64	0.76	0.06	44.43	1.51
14	1.23	47.53	2.16	0.42	47.40	2.76
15	0.55	45.79	1.78	0.44	47.49	2.79
16	3.12	49.54	1.85	3.17	50.73	2.57
17	1.95	50.28	3.23	2.09	51.58	3.94
18	3.91	52.81	3.42	4.16	54.24	4.24
19	4.08	53.26	3.61	4.49	54.56	4.24
20	6.77	56.02	3.81	6.66	57.37	4.92
21	5.43	55.04	3.95	6.02	56.60	4.73
22	9.60	59.91	5.26	8.56	59.87	5.80
23	11.01	58.24	3.09	11.20	59.54	4.09
24	11.80	60.40	4.53	11.62	61.33	5.42
25	14.70	60.72	3.45	14.57	61.37	4.10
26	15.96	61.93	4.03	15.88	62.88	4.93
27	18.29	62.86	4.05	18.61	63.49	4.54
28	21.56	64.45	4.62	21.06	64.55	4.84
29	20.57	64.20	4.63	20.03	64.17	4.75
30	24.79	65.86	5.35	23.86	65.24	4.90
31	24.78	66.17	5.66	24.87	66.46	5.93
32	26.94	63.76	2.98	26.41	64.63	3.90
33	27.85	66.46	5.59	28.18	67.04	6.14
34	30.92	66.69	5.69	29.80	66.56	5.58
35	32.07	64.84	3.86	31.83	65.61	4.62
36	38.43	66.47	6.16	37.15	66.82	6.29
37	34.93	67.08	6.27	35.32	67.16	6.39
38	40.76	65.39	5.61	39.33	65.77	5.65
39	41.13	64.39	4.74	41.69	65.23	5.70
40	41.96	62.68	3.33	41.60	64.35	4.83
	<i>Promedio</i>		3.49	<i>Promedio</i>		3.90

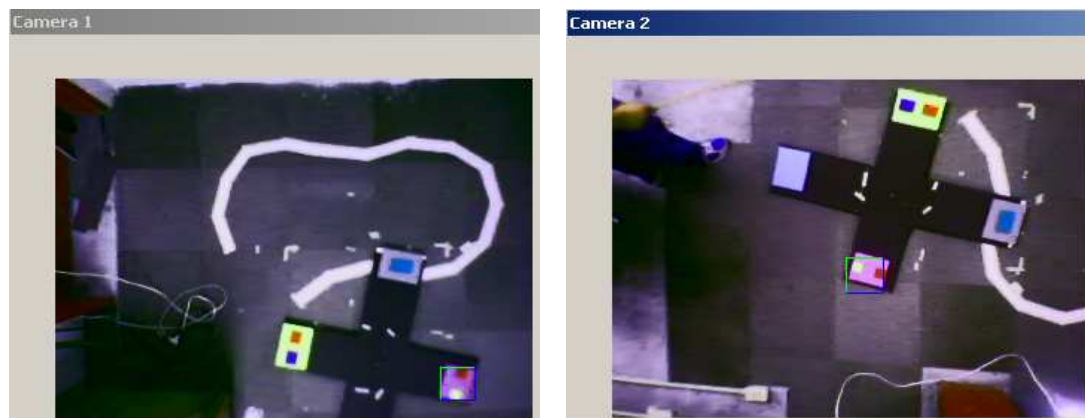
## 4.6. Seguimiento intercámara

El seguimiento intercámara es una funcionalidad muy importante con la que debe contar un sistema multicámara para seguir a un objeto a través del espacio de análisis combinado y así cubrir áreas de mayor extensión en forma simultanea. En las Figuras [4.11 - 4.13] se muestra un ejemplo de las ventajas que presenta la interacción entre los agentes cámara y control, para realizar el seguimiento entre cámaras de los WMR.



(a) Mapa de entorno. En azul se muestra el seguimiento realizado por el agente cámara B y en rojo el del agente cámara A. En verde se muestra la estimación fusionada al combinar los datos de los dos agentes. Se muestra la trayectoria desde un punto inicial hasta los límites del área de interés del agente cámara A.

(b) Pizarrón. Se muestra la información proporcionada por los dos agentes

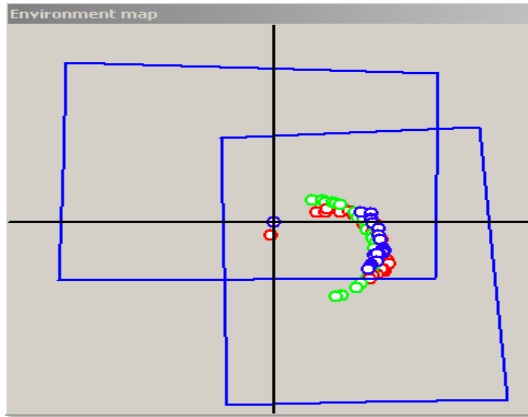


(c) Percepción del agente cámara A. El recuadro de seguimiento se encuentra sobre el objeto de interés.

(d) Percepción del agente cámara B. El recuadro de seguimiento se encuentra sobre el objeto de interés.

Figura 4.11: Se muestra el proceso de seguimiento de un objeto por dos agentes cámara hasta los límites del área de seguimiento de uno de los dos agentes cámara

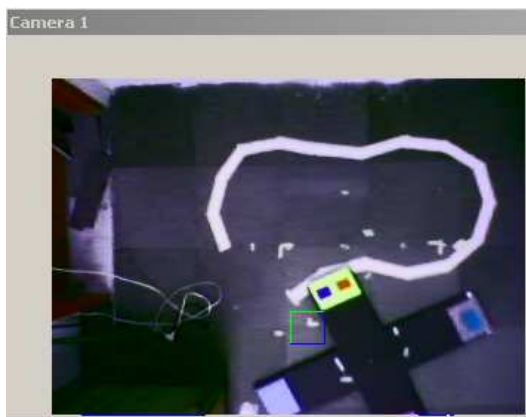
En estas figuras se presenta la situación característica donde un objeto(X) deja de ser visto por un agente cámara (A) y posteriormente se vuelve a incorporar a su área de análisis. En este caso, A solicita información de X al agente de control (C).



AgentID	ObjectID	PosX	PosY	TimeMark
MISSED	MISSED	MISSED	MISSED	MISSED
1	0	42.620106	-58.581901	9608416
FUSED	0	64.312469	-43.293224	9607484

(a) Mapa de entorno. Seguimiento del agente cámara A y B en color azul y rojo respectivamente; se presenta la estimación fusionada de los datos de los dos agentes en color verde. Se muestra la trayectoria del objeto hasta salir del área de interés del agente cámara A, a partir de ese punto el agente cámara B, es el único que continúa siguiendo al objeto.

(b) Pizarrón. En el registro correspondiente al agente cámara A, se indica que se ha dejado de seguir al objeto de interés.



(c) Percepción del agente cámara A. El recuadro de seguimiento se encuentra sobre el objeto de interés.

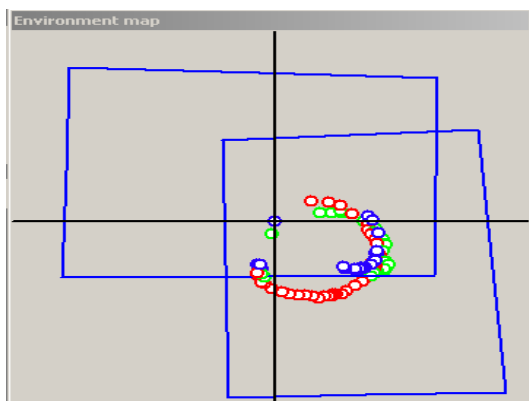
(d) Percepción del agente cámara B. El recuadro de seguimiento no se encuentra sobre el objeto de interés.

Figura 4.12: Se muestra el proceso de seguimiento de un objeto de interés por dos agentes cámara hasta los límites del área de seguimiento de uno de los dos agentes cámara, a partir de ahí sólo el agente cámara B, continúa siguiendo al objeto dentro de su trayectoria.

Existe otro agente cámara(B) que está percibiendo a X, B envía la información C



y este último actualiza el *pizarrón*. Entonces C al recibir la solicitud de información de A, verifica en el *pizarrón* si existe información de X, si es así, divide el área de análisis de A en  $K$  zonas y busca la más cercana ( $K_1$ ) a X. Una vez hecho esto, C notifica  $K_1$  a A, por lo que A busca en  $K_1$  al objeto X, si lo encuentra, se reinicia el seguimiento de X.



AgentID	ObjectID	PosX	PosY	TimeMark
0	0	-8.411428	-40.345055	10300281
1	0	-11.826325	-39.347271	10300381
FUSED	0	-12.592598	-40.140934	10300381

(a) Mapa de entorno. Se muestra el seguimiento realizado por el agente cámara A (Azul), el agente cámara B (Rojo) y la fusión de datos de las estimaciones de los agentes cámara (Verde). Se observa la trayectoria del objeto de interés al entrar nuevamente al área de interés del agente cámara A.

(b) Pizarrón. En los registros correspondientes a los dos agentes, se indica la posición del objeto de interés.



(c) Percepción del agente cámara A. El recuadro de seguimiento se repositiona sobre el objeto de interés.

(d) Percepción del agente cámara B. El recuadro de seguimiento se encuentra sobre el objeto de interés.

Figura 4.13: Se muestra el proceso de seguimiento de un objeto de interés por dos agentes. Un agente deja de seguir al objeto porque sale de su área de cobertura y retoma el seguimiento al buscarlo en una zona determinada indirectamente por el otro agente cámara.

La figura 4.11 muestra el seguimiento de un objeto (X) mediante dos cámaras.

En este caso los dos agentes cámara analizan el movimiento del objeto como se puede percibir en la figura 4.11(a). Se observa que X se dirige hacia los límites del agente cámara A, por lo que presumiblemente saldrá del área de análisis.

Instantes después en la figura 4.12, el agente cámara A reporta como perdido a X, figura 4.12(b), por lo que solicita información de él al agente de control. En la figura 4.12(c) se observa que el recuadro indicador de seguimiento se encuentra en su posición por omisión. Se muestra en la figura 4.12(a) que X es seguido, pero únicamente por el agente cámara B, éste le envía la información al agente de control. La información de X le es enviada al agente cámara A, sin embargo este aún no localiza a X porque aún se encuentra fuera de su área de cobertura.

Posteriormente, X reingresa al área de análisis del agente cámara A, Figura 4.13. Por lo que finalmente es localizado en el área indicada por el agente de control. Se muestra en la Figura 4.13(a) que el análisis del movimiento de X es retomado en el instante que entra nuevamente al área de interés del agente cámara A.

La interacción entre agentes permite disminuir el procesamiento a  $1/K$  en el mejor de los casos y facilita la localización de los objetos que entran al área de interés de un agente cámara, siempre y cuando exista otro agente cámara que los esté analizando.

El seguimiento intercámara es un elemento que permite realizar el análisis del movimiento de los objetos de interés a lo largo de las áreas de interés combinadas de las cámaras. El esquema propuesto proporciona la capacidad de comunicación entre agentes cámara de información de localización de los objetos de interés, lo cual permite reducir el tiempo de búsqueda de los objetos al pasar de una área de cobertura de un agente cámara a otra. En los experimentos mostrados en esta subsección se presentó una situación característica donde la comunicación entre agentes cámara permite localizar a un objeto de interés que sale de una área de cobertura y que posteriormente vuelve a entrar a ella.

## 4.7. Resumen

En este capítulo se presentó la caracterización de los Filtros de Kalman implementados para realizar el seguimiento de los objetos de interés a través de las dos cámaras utilizadas para los experimentos. Asimismo, se mostró la localización de estas cámaras con respecto al plano de referencia utilizado. Se presentaron los experimentos de seguimiento bajo un esquema de STT y MTT monocámara, posteriormente se extendieron estos experimentos bajo un esquema multicámara a través de la arquitectura multiagente propuesta. Además, se presentaron los resultados obtenidos por la fusión de datos en las áreas traslapadas de los agentes cámara y finalmente se mostró un caso característico donde la comunicación indirecta entre los agentes permite el seguimiento y relocalización a través de las áreas combinadas de interés de los agentes cámara.

## Capítulo 5

# Conclusiones y Trabajo Futuro

Esta tesis propone una arquitectura multicámara particular, con el objetivo de proporcionar mayor flexibilidad, robustez y escalabilidad al seguimiento de robots móviles.

Durante el transcurso de esta investigación, se desarrolló un sistema de seguimiento STT mediante el uso de un Filtro de Kalman, posteriormente, este sistema se extendió para realizar el seguimiento de múltiples objetos (MTT) a través de la implementación de filtros de Kalman en paralelo, finalmente se incorporó un esquema multi-agente basado en una arquitectura de pizarrón para dar soporte al sistema multicámara presentado en este trabajo.

### 5.1. Los sistemas de seguimiento STT/MTT

El abaratamiento de la tecnología ha permitido la proliferación de sistemas de seguimiento basados en esquemas STT y MTT de bajo costo. La necesidad de automatizar procesos ha promovido la investigación de este campo en los últimos años. En el proceso de seguimiento de objetos se deben resolver fundamentalmente dos problemas, la localización de estos objetos y el análisis del movimiento de los mismos. En nuestro sistema de seguimiento el atributo que utilizamos para localizar a los objetos es el color y para analizar el movimiento de los objetos se utiliza el filtro de Kalman.

#### 5.1.1. Localización

En nuestro sistema de seguimiento se utilizaron cámaras web para realizar la captura de imágenes del ambiente. Este tipo de cámaras son de bajo costo, lo cual permite disminuir el costo necesario para montar un sistema de seguimiento como el desarrollado; sin embargo, la mala calidad en las imágenes capturadas implicó la necesidad de contar con un sistema de localización efectivo, por lo que se utilizó un modelo de color que permitiera discernir de una mejor forma entre los objetos de interés. La localización de los objetos de interés se realizó mediante el uso del modelo de color  $L_1L_2L_3$ , el cual es invariante a la iluminación y a la rotación de los objetos al igual que el modelo *HSI*. La gran ventaja del uso de este modelo de color es que proporciona

una mayor robustez que el modelo de color por omisión RGB que dan como salida las cámaras Web empleadas en el desarrollo de este trabajo de investigación y la conversión de  $RGB-L_1L_2L_3$  es mucho más barata en términos computacionales que la de  $RGB-HSI$ .

### 5.1.2. Análisis de movimiento

El filtro de Kalman es usado frecuentemente como herramienta básica en los sistemas de seguimiento. Este filtro permite estimar en base a un modelo de movimiento (Modelo de Sistema) y uno de medición (Modelo de Medición), la posición de un objeto, a través de las posiciones anteriores. Se decidió implementar esta técnica debido a su probado desempeño en este tipo de sistemas, de esta forma, se tiene un parámetro de comparación para trabajos posteriores que tengan por objetivo probar la arquitectura multiagente desarrollada pero con otras herramientas matemáticas de seguimiento.

En este trabajo de tesis el análisis de movimiento se realizó mediante un filtro de Kalman para seguir a un objeto y a través de varios filtros para seguir a múltiples objetos en esquemas STT y MTT respectivamente. Este enfoque permitió estimar en tiempo real las posiciones futuras de los objetos de interés, realizando un análisis individual del movimiento de cada objeto. Este modelo presentó un error bajo en el seguimiento de uno y varios objetos en los experimentos realizados, lo cual es indispensable si se quiere retroalimentar a los robots móviles.

## 5.2. Sistema Multiagente/Multicámara

El sistema multiagente propuesto proporciona la capacidad de implementar un sistema multicámara que cuenta con flexibilidad de incorporar a varias cámaras y que permite extender y mejorar la localización y seguimiento de los objetos de interés definidos. Esto es logrado mediante el uso de la fusión de datos y la interacción entre cámaras bajo esta arquitectura.

## 5.3. Arquitectura de pizarrón para el seguimiento de objetos

La arquitectura de pizarrón ha permitido que las cámaras interactúen de forma indirecta con la finalidad de extender el área de cobertura de sensado. Utilizando este modelo, se definieron dos tipos de agentes:

- *Agente Cámara.* Este tipo de agente localiza a los objetos de interés dentro de su área de cobertura, posteriormente, su objetivo principal es el seguimiento de estos

dentro del ambiente y la notificación de las estimaciones de posición al agente de pizarrón. Asimismo, el agente cámara tiene como meta localizar y seguir al mayor número de objetos de interés en su ambiente de sensado.

- *Agente de Pizarrón/Control.* El objetivo de este agente es la administración del área de memoria compartida denominado Pizarrón. Por medio de este, se puede condensar el seguimiento de los objetos de interés con la finalidad de tener un ambiente combinado de todas las vistas parciales de los agentes cámara. Se definió que este agente realizara la fusión de datos de los objetos en áreas de traslape porque tiene acceso directo al pizarrón y por ende a la información de los objetos de interés analizados en el entorno global. pizarrón.

### 5.3.1. Fusión de datos

El método utilizado para fusionar datos en esta investigación es una adaptación del Filtro de Kalman (descrita en la sección 3.5), que aprovecha el modelo de sistema empleado por esta herramienta matemática para acoplar las estimaciones de los agentes cámara en el espacio temporal y considerarlas de esta forma como si se hubieran realizado en el mismo instante. La fusión de datos se realiza en las áreas de cobertura donde hay traslape entre el análisis de los agentes. La estimación obtenida en los experimentos al fusionar los datos con ventanas de umbrales de tiempo de 100, 150 y 200ms de los agentes cámara presenta un mejor desempeño que la estimación realizada por el agente cámara que presenta mayor error en su seguimiento.

### 5.3.2. Seguimiento Intercámara

El seguimiento intercámara se logró mediante la interacción indirecta entre agentes cámara vía el agente de control y el pizarrón, el proceso básico para lograrlo consistió en hacer uso de la información condensada en el pizarrón con la finalidad de comunicar a los agentes cámara las zonas donde deben de buscar a los objetos de interés que han dejado de analizar. El objetivo principal de este esquema es permitir que los objetos puedan ser seguidos entre las áreas de interés cubiertas por los agentes cámara, de esta forma, el seguimiento puede realizarse en el área total correspondiente a las áreas de interés combinadas de los agentes cámara.

## 5.4. Conclusiones Generales

El sistema de seguimiento multicámara propuesto basado en una arquitectura multiagente de tipo pizarrón proporciona una mayor robustez y flexibilidad al permitir el seguimiento de los objetos de interés dentro de las áreas combinadas de los agentes

cámara y realizar la fusión de datos dentro de las áreas traslapadas de sus percepciones. Asimismo, la arquitectura propuesta presentó un buen desempeño en el seguimiento de objetos bajo esquemas STT y MTT al mostrar errores de seguimiento de baja magnitud en los experimentos realizados.

## 5.5. Trabajo Futuro

Como trabajo futuro se presentan diversas áreas donde se puede continuar con esta investigación:

- Incorporación de diversas herramientas matemáticas para realizar el análisis de movimiento en el sistema de seguimiento. Técnicas tales como el filtro de partículas o el algoritmo de desplazamiento medio podrían ser comparadas con el sistema actual de seguimiento e incluso podrían ser combinadas en la etapa fusión de datos, con lo que podría realizarse interesantes análisis y mejoras de desempeño de las estimaciones de las posiciones de los WMR.
- Uso de cámaras móviles y cámaras activas para realizar el seguimiento mediante áreas de interés variables, lo cual proporcionaría mayor flexibilidad al sistema.
- Extensión del sistema actual para realizar el seguimiento e identificación de personas en un esquema de vigilancia.

## Apéndice A

### Calibración de las cámaras

En este apéndice primero se describe el modelo de geometría de cámara utilizado (modelo de cámara Tsai), posteriormente se muestra el procedimiento empleado para obtener los parámetros intrínsecos y extrínsecos de las cámaras a través de un *toolbox* de *Matlab* creado por Jean-Yves Bouguet [5].

#### A.1. Introducción

La calibración de las cámaras es un procedimiento muy importante en el análisis cuantitativo de imágenes. Esta área se ocupa de la identificación de información (primitivas de imágenes, atributos, objetos) en el espacio dentro de la imagen y de su transferencia y representación en el espacio de los objetos [12].

La calibración de las cámaras es un proceso necesario para extraer información 3-D a partir de las imágenes que están en 2-D. Las imágenes analizadas se diferencian una a otra a través de ciertos parámetros. Las características internas se conocen como *parámetros intrínsecos* (longitud focal y distorsión de la lente), mientras los parámetros que describen la geometría se denominan *parámetros extrínsecos* (posición y orientación con respecto a algún sistema de coordenadas)[26, p. 307].

#### A.2. Modelo de cámara Tsai

Uno de los métodos de calibración más populares es el método Tsai, el cual es utilizado en una gran cantidad de aplicaciones debido a que puede lidiar con puntos coplanares y nocoplanares. Éste método también ofrece la posibilidad de calibrar los parámetros internos y externos separadamente, lo cual es útil porque da la facilidad de obtener los parámetros intrínsecos de la cámara y después realizar la estimación de posición de la cámara con respecto a un sistema de referencia.

A continuación se describe el modelo de cámara Tsai y las ecuaciones relacionadas a este en base a la transformación de perspectiva mostrada en la figura A.1.  $P_w = (x_w, y_w, z_w)$  son las coordenadas 3D de un punto del objeto en el sistema de coordenadas

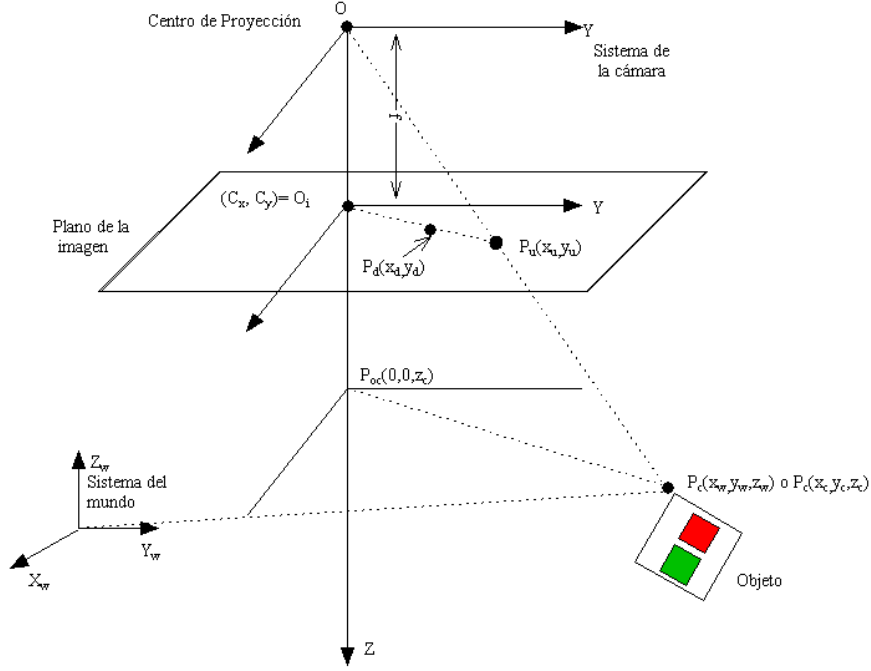


Figura A.1: Geometría de la cámara

. Se muestra la geometría de la cámara con proyección de perspectiva y distorsión radial del lente.

del mundo.  $P_c = (x_c, y_c, z_c)$  son las coordenadas 3D de un punto del objeto en el sistema de coordenadas de la cámara, centrado en el punto  $O$ , el cual es el centro óptico, con el eje  $z$  el mismo que el eje óptico.  $(X, Y)$  es el sistema de coordenadas paralelo a los ejes  $(x, y)$ . La distancia entre el plano de la imagen y el centro óptico es  $f$ .  $P_w = (X_w, Y_w)$  son las coordenadas del punto imagen  $P_c = (x_c, y_c, z_c)$  tomando en cuenta el modelo de cámara ideal con apertura infinitesimal.  $(X_d, Y_d)$  son las coordenadas reales de la imagen.  $(X, Y)$  son las coordenadas utilizadas en el computador, es decir el número de píxeles en la imagen discreta [26, p. 308].

El proceso de transformación de  $(x_w, y_w, z_w)$  a  $(X, Y)$  se ilustra en la figura A.2.

Enseguida se describen los pasos mostrados en la figura A.2:

- Paso 1: Transformación desde el sistema de coordenadas del mundo  $P_w = (x_w, y_w, z_w)$  al sistema de coordenadas de la  $P_c = (x_c, y_c, z_c)$

$$\mathbf{P}_c = \mathbf{R}\mathbf{P}_w + \mathbf{t} \quad (\text{A.1})$$

donde  $\mathbf{R}$  es la matriz de rotación con dimensiones 3x3 y  $\mathbf{t}$  es un vector de traslación

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (\text{A.2})$$



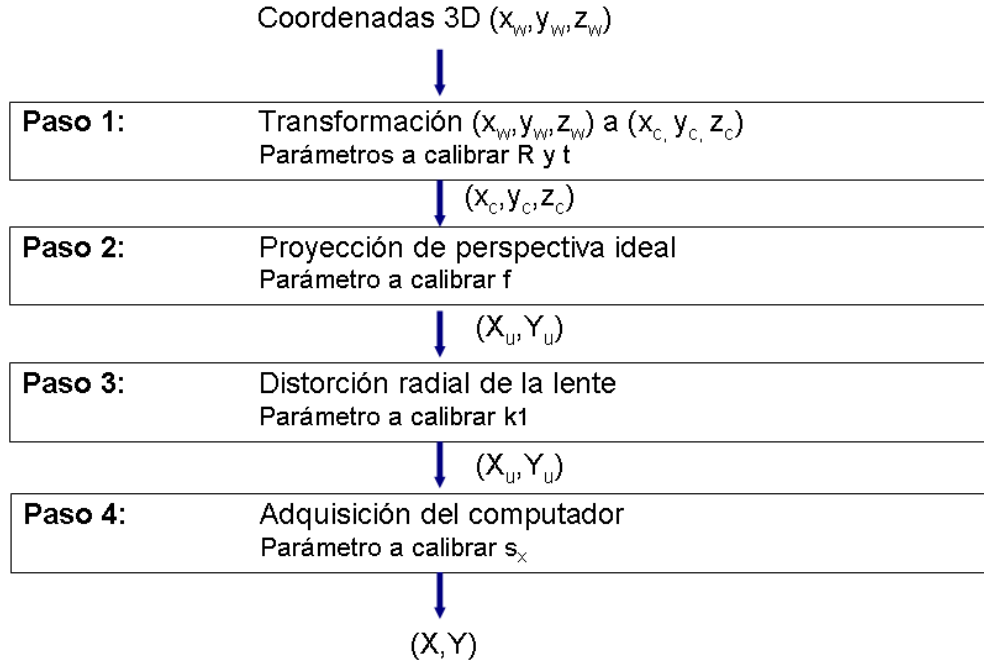


Figura A.2: Transformación de coordenadas

. Proceso de transformación de coordenadas del mundo 3D a las coordenadas de la imagen en el computador.

$$\mathbf{t} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (\text{A.3})$$

- Paso 2: Se transforma el sistema de coordenadas tridimensional de la cámara y una imagen del punto no distorsionada ( $X_u, Y_u$ ).

$$X_u = f \frac{x_c}{z_c} \quad (\text{A.4})$$

$$Y_u = f \frac{y_c}{z_c} \quad (\text{A.5})$$

- Paso 3: Se describe la distorsión radial de las lentes y las coordenadas de la imagen en el plano.

$$X_u = X_d + D_x \quad (\text{A.6})$$

$$Y_u = Y_d + D_y \quad (\text{A.7})$$

donde ( $X_d, Y_d$ ) son las coordenadas reales de la imagen(distorsionadas) y  $D_x$  y  $D_y$  están dadas por:

$$D_x = X_d(k_1 r^2) \quad (\text{A.8})$$

$$D_y = Y_d(k_1 r^2) \quad (\text{A.9})$$

$$r = \sqrt{X_d^2 + Y_d^2} \quad (\text{A.10})$$

donde  $k_1$  es el coeficiente de distorsión radial.

- Paso 4: Se transforma de coordenadas reales  $(X_d, Y_d)$  a coordenadas de la imagen tal y como las proporciona la computadora  $(X, Y)$ .

$$X = X_d + c_x \quad (\text{A.11})$$

$$Y = Y_d + c_y \quad (\text{A.12})$$

donde  $(c_x, c_y)$  es el número de fila y columna del centro de la imagen en memoria del computador

### A.3. Mapeo Inverso

El cálculo de las coordenadas del mundo 3D a través de las filas y columnas de pixeles es útil en la mayoría de las aplicaciones de visión computacional aplicada a la robótica. En nuestro caso el mapeo inverso se hace al suponer que todos los puntos del mundo son coplanares, es decir todas las coordenadas consideradas cuentan  $z_w = 0$ . Despejando  $P_w$  de la ecuación A.1.

$$\mathbf{R}^{-1} \begin{bmatrix} X_c - T_x \\ Y_c - T_y \\ Z_c - T_z \end{bmatrix} = \begin{bmatrix} X_w \\ Y_w \\ 0 \end{bmatrix} \quad (\text{A.13})$$

Expresando los valores de  $\mathbf{R}^{-1}$  con lo que la ecuación A.13 se representa de la siguiente forma.

$$\begin{bmatrix} y_1 & y_2 & y_3 \\ y_4 & y_5 & y_6 \\ y_7 & y_8 & y_9 \end{bmatrix} \begin{bmatrix} X_c - T_x \\ Y_c - T_y \\ Z_c - T_z \end{bmatrix} = \begin{bmatrix} X_w \\ Y_w \\ 0 \end{bmatrix} \quad (\text{A.14})$$

La incógnita buscada en esta ecuación es  $Z_c$ . Su valor se obtiene a través de los términos que involucran  $Z_w$ , la cual sabemos que es 0. La ecuación considerada queda expresa en la ecuación A.15.

$$y_7 (X_c - T_x) + y_8 (Y_c - T_y) + y_9 (Z_c - T_z) = 0 \quad (\text{A.15})$$

Manipulando las ecuaciones A.4 y A.5 para expresar  $X_c$  y  $Y_c$  en términos de variables conocidas, se obtiene así la siguiente expresión

$$\frac{X_u}{f} = \frac{X_c}{Z_c} = \alpha \quad (\text{A.16})$$

$$\frac{Y_u}{f} = \frac{Y_c}{Z_c} = \beta \quad (\text{A.17})$$

Se despeja  $Y_c$  y  $Z_c$ .

$$Z_c = \frac{X_c}{\alpha} \quad (\text{A.18})$$

$$Y_c = \beta \frac{X_c}{\alpha} \quad (\text{A.19})$$

Sustituyendo  $Y_c$  y  $Z_c$  en la ecuación A.15 se obtiene

$$y_7 (X_c - T_x) + y_8 \left( \beta \frac{X_c}{\alpha} - T_z \right) + y_9 \left( \frac{X_c}{\alpha} - T_c \right) = 0 \quad (\text{A.20})$$

Con esta última ecuación se calcula  $X_c$ , para posteriormente obtener  $Y_c$  y  $Z_c$ . Finalmente, una vez obtenido el vector  $P_c$  se transforma a las coordenadas en 3D con respecto al sistema de referencia, este vector es  $P_w$ .

## A.4. Obtención de los parámetros intrínsecos y extrínsecos

Enseguida se explica el procedimiento utilizado para obtener los parámetros intrínsecos y extrínsecos de las cámaras estáticas. La obtención de estos parámetros se hace mediante el uso de un *toolbox* de *Matlab* desarrollado por Jean-Yves Bouguet [5].

El programa de calibración cuenta con una interfase gráfica mostrada en la figura A.3

Las opciones que el programa presenta son mostradas en forma de botones. Las opciones necesarias para realizar la calibración de una cámara se detallan abajo:

- *Image names.* Establece el nombre base de las imágenes para la calibración de la cámara.
- *Read image.* Carga en memoria las imágenes que permiten la calibración de la cámara.
- *Extract grid corners.* Extrae las esquinas del patrón de cuadros para la calibración.



Figura A.3: Toolbox de Calibración.

Interfase gráfica del programa utilizado para la obtención de los parámetros internos y externos de las cámaras.

- *Calibration.* Calcula los parámetros intrínsecos de la cámara.
- *Comp. Extrinsic.* Calcula los parámetros extrínsecos de la cámara.

El programa utiliza una serie de imágenes de un tablero plano que le sirven como patrón para estimar los parámetros intrínsecos. El número de imágenes utilizadas en la calibración es veinte, en diferentes ángulos y a diferentes distancias de la cámara. En la figura A.4 se ilustra un ejemplo del conjunto de imágenes. El procedimiento que se debe seguir para obtener los parámetros intrínsecos de las cámaras a través del programa es el siguiente:

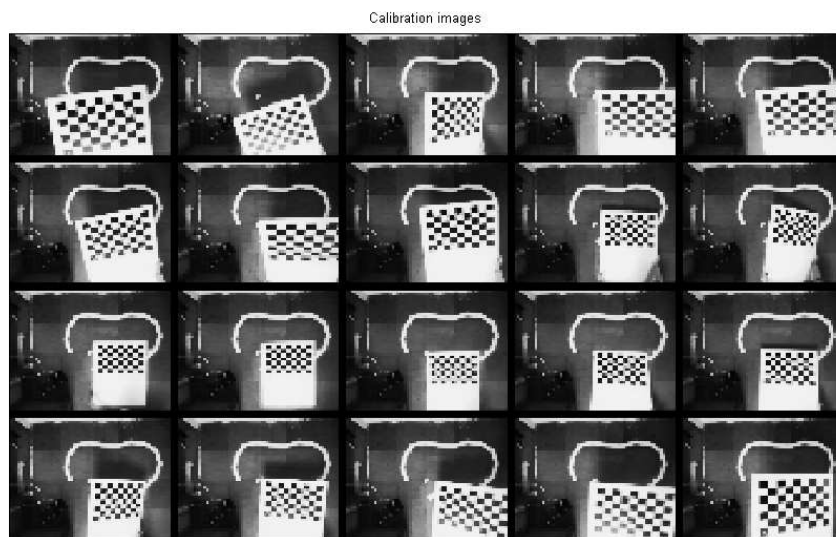


Figura A.4: Ejemplo de un conjunto de imágenes de calibración.

Se muestra una serie de imágenes utilizadas para realizar la calibración.

1. Se establece el nombre base de las imágenes de calibración mediante la opción *Image names*.

2. Se carga el conjunto de imágenes de calibración en memoria para ser utilizadas. El *toolbox* soporta diversos formatos de imágenes (RAS, BMP, TIFF, JPG y PPM), los utilizados en esta investigación fueron BMP.
3. *Extracción de esquinas*. Se presiona el botón *Extract Grid Corners* en la ventana del programa de calibración. Se seleccionan las esquinas del tablero patrón en las imágenes de calibración. En la figura A.5 se puede apreciar un ejemplo de cómo se seleccionan las esquinas y cómo quedan referenciados los ejes coordenados  $x$  e  $y$

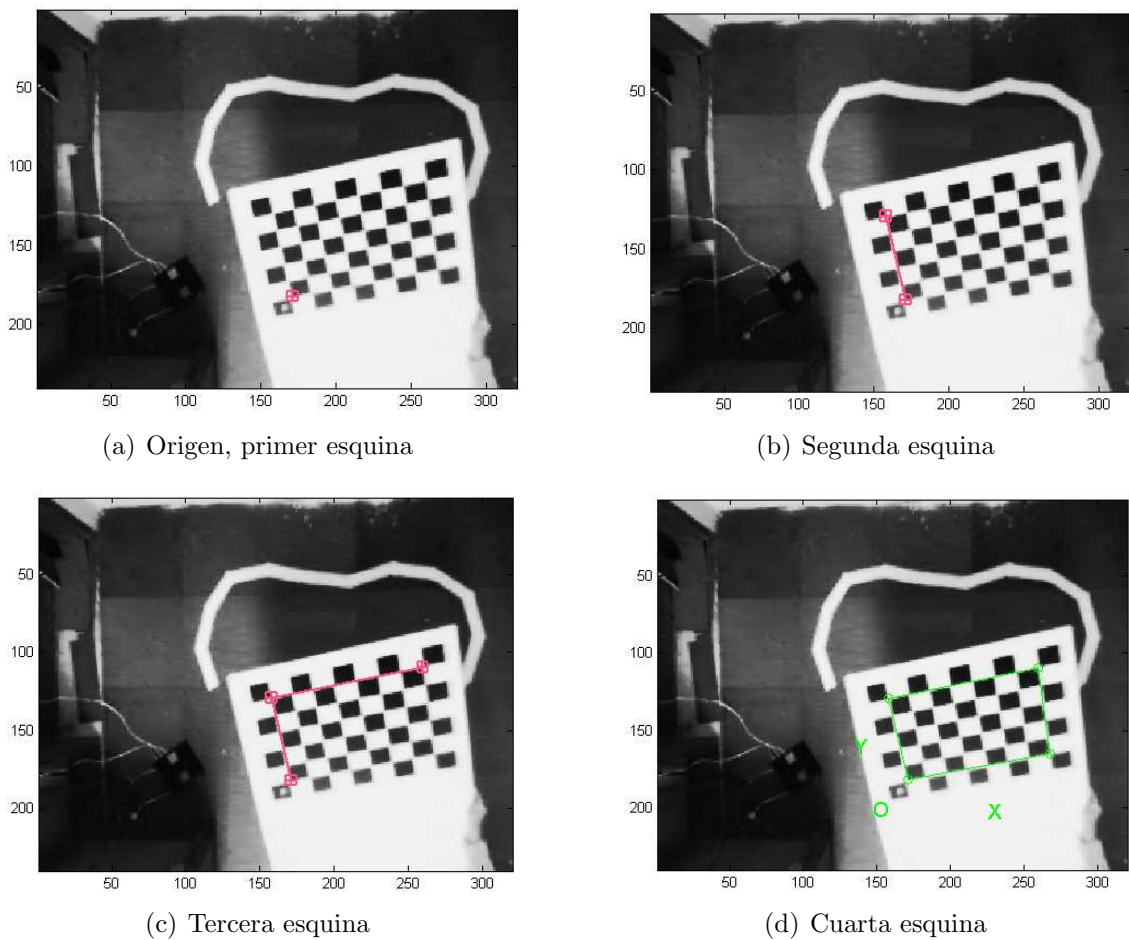


Figura A.5: Selección de esquinas de calibración.

Se muestra un ejemplo de la selección de cuatro esquinas en una imagen de calibración.

4. *Calibración*. Después de la extracción de las esquinas, seleccione el botón *Calibration* dentro del programa de calibración para ejecutar el procedimiento principal

de calibración. La calibración se hace en dos pasos: Primero la inicialización, y después la optimización no lineal.

- La etapa de inicialización calcula una solución para los parámetros de calibración basados en ninguna distorsión del lente.
- La etapa de optimización no lineal minimiza el total del error de reproyección mediante mínimos cuadrados sobre todos los parámetros (longitud focal, punto principal, coeficientes de distorsión). La optimización es realizada por el método de gradiente descendente iterativo.

Una vez obtenidos los parámetros intrínsecos, se calculan los parámetros extrínsecos. Para obtener estos parámetros se debe oprimir el botón *Comp. Extrinsic* con la finalidad de seleccionar un marco de referencia. Éste es calculado mediante la elección de una imagen que contenga un patrón de calibración que indique lo que se tomará como referencial. La figura A.6 muestra un ejemplo de cómo queda localizado el marco de referencia en una imagen de calibración ( $O, X, Y, Z$ ).

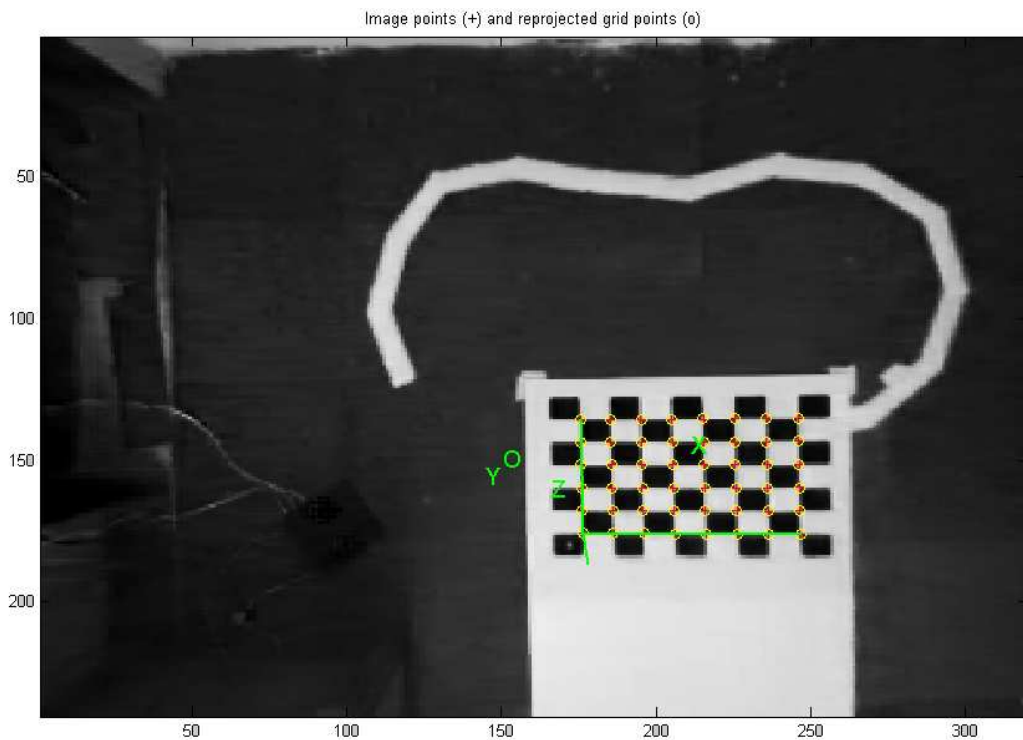


Figura A.6: Marco de referencia .

Ejemplo de la localización de la cuadrícula del marco de referencia ( $O, X, Y, Z$ ).

Los parámetros extrínsecos son codificados en las matrices de rotación ( $\mathbf{Rc\_ext}$ ) y de traslación ( $\mathbf{Tc\_ext}$ ) los cuales corresponden a las matrices representadas en las ecuaciones A.2 y A.3, respectivamente.

## A.5. Resumen

En este apéndice se describió el modelo de cámara utilizado para realizar el mapeo entre las coordenadas dadas en píxeles y las del mundo 3D de los puntos ubicados en el espacio de trabajo y viceversa. Además se explicó el procedimiento para realizar la calibración de las cámaras empleadas mediante la obtención de sus parámetros intrínsecos y extrínsecos a través de una herramienta computacional de uso público. Los parámetros obtenidos durante la calibración son guardados en archivos de configuración, posteriormente los agentes cámara utilizan estos archivos con la finalidad de leer los parámetros y mapear las coordenadas entre píxeles y el mundo 3D por donde se desplazan los objetos de interés.

## Apéndice B

### Interfase computacional y archivos de configuración

En este apéndice se explica la información que el sistema de seguimiento multicámara presenta en forma gráfica para el usuario. Además, se detalla los datos que deben contener los archivos de configuración de los agentes cámara para la correcta proyección entre las coordenadas en píxeles y el mundo en 3D.

#### B.1. Interfase computacional

Las ventanas presentadas por el sistema multicámara de seguimiento son las siguientes:

- Ventana de Mapa de Entorno. En esta ventana se presenta un bosquejo de las áreas analizadas por los agentes cámara. Además se muestra la localización de los objetos interés con respecto a estas delimitaciones, figura B.1.

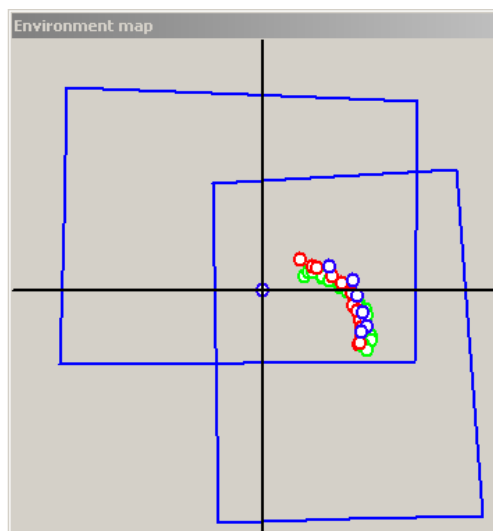


Figura B.1: Mapa de Entorno.

Se presenta la ventana que el sistema de seguimiento ofrece del entorno analizado por los diversos agentes cámara.



- Ventana de Pizarrón. Se muestra el condensado de las estimaciones de las posiciones de los objetos interés realizadas por los agentes cámara, figura B.2.

AgentID	ObjectID	PosX	PosY	TimeMark
0	0	75.127068	-33.219563	5169903
1	0	70.684464	-28.623669	5170144
FUSED	0	71.732246	-30.568800	5169683

Figura B.2: Pizarrón.

En el pizarrón se condensa la información de las estimaciones de posición de los objetos realizadas por los agentes cámara.

- Ventana de percepción. Cada agente cuenta con una ventana de percepción. En esta ventana se muestra la información que el agente cámara está percibiendo de su entorno, figura B.3.

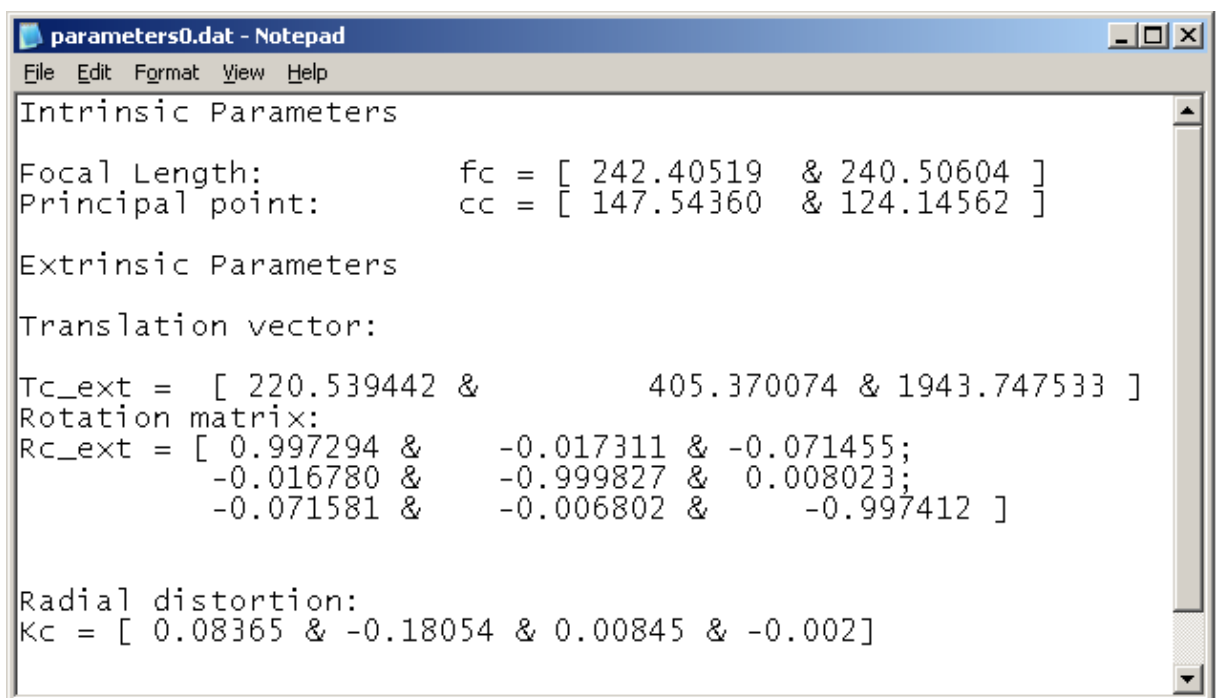


Figura B.3: Ventana de Percepción.

La percepción del ambiente realizada por cada agente se despliega a través de esta ventana.

## B.2. Archivos de configuración

Los archivos de configuración guardan los parámetros de calibración de las cámaras utilizadas como sensores por los agentes cámara, figura B.4. Se puede observar que las columnas de las matrices son separadas por el símbolo  $\&$  y las líneas por  $;$ . De esta forma, después de realizar la calibración de las cámaras empleando el método explicado en el apéndice anterior, se guardan los parámetros en archivos de texto para que los agentes cámara puedan utilizarlos para realizar el mapeo entre coordenadas de píxeles al mundo 3D y viceversa cada vez que sea necesario.



```
parameters0.dat - Notepad
File Edit Format View Help
Intrinsic Parameters
Focal Length:          fc = [ 242.40519 & 240.50604 ]
Principal point:      cc = [ 147.54360 & 124.14562 ]

Extrinsic Parameters
Translation vector:
Tc_ext = [ 220.539442 &          405.370074 & 1943.747533 ]
Rotation matrix:
Rc_ext = [ 0.997294 &    -0.017311 & -0.071455;
          -0.016780 &    -0.999827 &  0.008023;
          -0.071581 &    -0.006802 & -0.997412 ]

Radial distortion:
Kc = [ 0.08365 & -0.18054 & 0.00845 & -0.002]
```

Figura B.4: Archivo de configuración.

Se muestra un ejemplo del contenido de un archivo de configuración de un agente cámara.

## Bibliografía

- [1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking, 2002.
- [2] A. Cavallaro D. Douxchamps T. Ebrahimi P. Figueiredo B. Macq B. Mory L. Nunes J. Orri M. Trigueiros B. Abreu, L. Botelho and A. Violante. Video-based multi-agent traffic surveillance system. MODEST Consortium -ADETTI (Lisbon, Portugal), EPFL (Lausanne, Switzerland), LEP (Paris, France), UCL (Louvain, Belgium), 2000.
- [3] J. Black. *Multi View Image Surveillance and Tracking*. PhD thesis, City University, London, April 2004.
- [4] S. Blackman and R. Popoli. *Design and Analysis of Modern Systems Tracking*. Artech House, Boston, MA, 1999.
- [5] J. Bouguet. Camera calibration toolbox for matlab. 2002.
- [6] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *IEEE International Conference on Computer Vision*, volume 2, pages 1197–1203.
- [7] D. Corkill. Collaborative software - blackboard and multiagent systems & the future. Technical report, Department of Computer Science, University of Massachusetts, 2004.
- [8] N. Paragios D. Comanaciu and M. Greiffenhagen. Real-time video analysis at siemens corporate research. *Real-time Imaging Group*, 2002.
- [9] M. Nöllich E. Pagello E. Menegatti, E. Mumolo. A surveillance system based on audio and video sensory agents cooperating with a mobile robot. Smartlab Department of Electrotechnics, Electronics and Computer Science, The University of Trieste, Italy, 2003.
- [10] J. Ferber. *Multi-Agent Systems- An Introduction to Distributed Artificial Intelligence*. Addison Wesley, Great Britain.

- [11] T. Gevers and A. Smeulders. Color-based object recognition. *Pattern Recognition*, 32:453–464, 1999.
- [12] A. Gruen and T. S. Huang. *Calibration and Orientation of cameras in computer vision*. Information Sciences. Springer, Berling, 2001.
- [13] Z. Hafed. Object tracking - principles and challenges to keeping an eye on things. *IEEE Potentials*, pages 10–13, 1999.
- [14] D. Hall and J. Llinas. *Handbook of multisensor data fusion*. CRC Press, New York, 2001.
- [15] S. Hong. Target tracking and data association. *KEES Radar Society, National Radar Workshop and Tutorial*, 2004.
- [16] S. Chen J. Dong and J. Jeng. Event-based blackboard architecture for multi-agent system. Department of Computer Science, Watson Research Center, 2004.
- [17] J. Garbó J. García and J. Molina. Agent-based coordination of cameras. *International Journal of Computer Science & Applications, Technomatics Research Foundation*, 2(1):33–37, 2005.
- [18] P. Remagnino J. Orwell and D. Greenhill. A multi-agent framework for visual surveillance. School of Computer Science and Electronic Systems, Kingston University, 1999.
- [19] Emil Kalman, Rudolph. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [20] R.Ñiu L. Snidaro, G. Foresti and P. Varshney. Sensor fusion for video surveillance. Department of Mathematics and Computer Science, University of Udine, Italy, 2004.
- [21] J. Liu and J. Wu. *Multi-Agent Robotic Systems*. Computational Intelligence. CRC Press, New York, 2001.
- [22] J. Mallet and Jr. M. Bove. Eye society. In *International Conference on Multimedia and Expo*.
- [23] H. Soto Montalvo. Un estudio de sistemas multiagente y multirobot. Technical report, Departamento de Informática, Estadística y Telemática, Universidad Rey Juan Carlos, 1999.

- [24] A. I. Shihab P. Remagnino. Distributed intelligence for multi-camera visual surveillance. *Pattern Recognition*, 2003.
- [25] K. Baker P. Remagnino, T. Tan. Agent orientated annotation in model based visual surveillance. In *16th National Conference on Computer Vision*.
- [26] G. Pajares and J. de la Cruz. *Visión por computador - Imágenes digitales y aplicaciones*. Alfaomega, Madrid, España, 2002.
- [27] A. Solera Ramírez. El filtro de kalman. *Banco Central de Costa Rica, Nota Técnica, Departamento de Investigaciones Económicas*, 2003.
- [28] T. Roberts, S. McKenna, and I. Ricketts. Adaptive learning of statistical appearance models for 3d human tracking, 2002.
- [29] J. Roux. An introduction to kalman filter: probabilistic and deterministic approaches. 2003.
- [30] S. Russell and P. Norvig. *Artificial Intelligence – A modern approach*. Artificial Intelligence. Prentice Hall, New Jersey, 2003.
- [31] Zeeshan S. Kahn, O. Javed and M. Shah. Human tracking in multiple cameras. Computer Vision Lab, University of Central Florida, 1999.
- [32] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.
- [33] Dan Simon. Kalman filtering. *Embedded Systems Programming*, (June):72–79, 2001.
- [34] A. F. Bobick S.S. Intille. A framework to recognizing multi-agent action from visual evidence. In *16th National Conference in Artificial Intelligence*.
- [35] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.
- [36] N. Ukita and T. Matsuyama. Real time multi-target tracking by cooperative distributed active vision agents. Graduate School University, Kyoto University, 2001.
- [37] L. Wang W. Hu, T. Tan and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on systems, man and cybernetics*.
- [38] Welch and Bishop. An introduction to the kalman filter. 2001.
- [39] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, 2004.