

## CAPÍTULO 9

### **ESTIMACIÓN DE OBJETOS CON FUSIÓN BAYESIANA EN EQUIPOS DE ROBOTS MÓVILES<sup>2</sup>**

C. E. AGÜERO <sup>1</sup>, JOSÉ M. CAÑAS <sup>1</sup>, V. MATELLÁN <sup>2</sup> y F. MARTÍN<sup>1</sup>

<sup>1</sup> Grupo de Robótica. Grupo de Sistemas y Comunicaciones - GSYC.  
Universidad Rey Juan Carlos (Madrid, España). E-mail:  
{caguero,jmplaza,fmartin}@gsyc.urjc.es

<sup>2</sup> Escuela de Ingeniería Industrial e Informática. Universidad de León  
(León, España). E-mail: [Vicente.matellan@unileon.es](mailto:Vicente.matellan@unileon.es)

Una de las aplicaciones habituales dentro de los sistemas multi-robot es la percepción distribuida. Este mecanismo de cooperación permite que cada robot sea capaz de modelar su entorno, más allá de lo que percibe a través de sus propios sensores, apoyándose en las observaciones de otros robots del equipo. Las técnicas probabilísticas de fusión sensorial pueden emplearse para combinar percepciones de varios robots y obtener una única estimación global. En este artículo se presentan dos aproximaciones al problema de la percepción distribuida de la pelota dentro del escenario de la *RoboCup*.

---

<sup>2</sup> Los resultados de este trabajo han sido parcialmente financiados por los proyectos COCOGROM del Ministerio de Educación y Ciencia (ref. DPI2007-66556-C03-01), MAVROM de la Comunidad de Madrid y Universidad Rey Juan Carlos (ref. URJC-CM-2007-CET-1694) y RoboCity2030 del programa de actividades de I+D de la Comunidad de Madrid (ref. S-0505/DPI/0176).

## 1 Introducción

El problema de la percepción en robótica ha sido habitualmente abordado desde un punto de vista individual. En este documento se describe el trabajo y los resultados obtenidos del desarrollo de un mecanismo de percepción compartido entre un grupo de robots móviles en un entorno dinámico: el escenario de la RoboCup[robocup].

Esta competición internacional trata de promover y evolucionar campos como la Inteligencia Artificial y la robótica a través de un problema común: un partido de fútbol entre dos equipos de robots. Existen diversas ligas con sus propios requerimientos en cuanto a número y tipo de robot. En nuestro caso, el trabajo se vertebra alrededor de la liga de plataforma estándar con robots de 4 patas[legged], donde sólo pueden ser utilizados los robots Sony aiBo[aiibo]. Cada equipo está formado por cinco robots y el reglamento permite que los jugadores puedan comunicarse entre sí usando sus dispositivos de red inalámbricos. En la figura 1 podemos observar varias instantáneas de la competición. El campo de juego simula un campo de fútbol en miniatura de aproximadamente 32 m<sup>2</sup> con dos balizas situadas a ambos lados del centro del campo para facilitar la auto-localización. Todos los elementos significativos están coloreados de manera conocida para favorecer su percepción.



Fig 1. Distintos instantes de la competición RoboCup en la categoría de plataforma estándar y robots de cuatro patas

AiBo se comercializó desde 1999 hasta 2006 por la compañía Sony. Su nicho de mercado principal fue la robótica de entretenimiento, es decir, se vendía como mascota virtual, con gran éxito en el mercado asiático. Además de su logrado acabado final y de su robustez mecánica, Sony desarrolló el software de un completo conjunto de comportamientos jerárquicos que hacían que aiBo imitara la conducta de un perro real.

AiBo captó la atención del público interesado en la robótica en el momento que Sony puso a su disposición un entorno de programación: *Open-R*. Centros de todo el mundo comenzaron paulatinamente a utilizar a aiBo como su plataforma de referencia, dado su bajo precio comparado con otros robots habituales en el mercado.

El sensor principal del robot es una cámara situada en su cabeza. A través de esta cámara se perciben los objetos interesantes de este entorno: Pelota, porterías, otros jugadores, líneas del campo y balizas. La percepción de la pelota es especialmente crítica. Minimizar el tiempo hasta localizarla significa llegar antes a ella y, por tanto, aumentar las probabilidades de alcanzarla antes que el equipo contrario.

Si cada robot puede compartir su información sobre la estimación de la pelota con el resto de compañeros, podremos mantener una mejor y más estable estimación sobre la misma. El objetivo de este trabajo es obtener dicha estimación distribuida combinando de manera adecuada las observaciones que se van generando procedentes de cada robot. Por supuesto, las incertidumbres asociadas tanto a las observaciones como a la posición del observador deberán ser tenidas en cuenta. Abordaremos el proceso de fusión desde dos implementaciones diferentes y extraeremos las ventajas e inconvenientes asociadas a cada método.

El resto del artículo se organiza de la siguiente manera: En la sección 2 revisaremos los métodos de fusión de información habituales en la literatura. Los algoritmos propuestos de estimación compartida de la pelota serán descritos en la sección 3. En la sección 4 presentaremos todos los experimentos realizados y, finalmente, describiremos las conclusiones extraídas de este artículo en la sección 5.

## **2 Métodos de fusión de información**

Las técnicas de fusión sensorial se han empleado habitualmente en aplicaciones para un solo robot. Con ellas es posible modelar el entorno combinando varias fuentes de información multimodal procedentes del mismo robot. Por ejemplo, en las aplicaciones de construcción de mapas se integran observaciones procedentes de sensores láser, de ultrasonidos o

incluso de cámaras con información de movimiento procedente de los sensores odométricos. En el capítulo III del libro *Probabilistic Robotics*[probRobotics] se hace un repaso muy detallado de las técnicas probabilísticas de construcción de mapas.

En la figura 2 podemos observar otro ejemplo de fusión multisensorial dentro de un único robot. Las imágenes muestran la reconstrucción del entorno generada por el robot *Junior*[Montemerlo08a] combinando múltiples sensores láser y rádar. *Junior* fue el vehículo desarrollado por la Universidad de Stanford y liderado por Sebastian Thrun para participar en la competición DARPA Urban Challenge[urban] en noviembre de 2007.

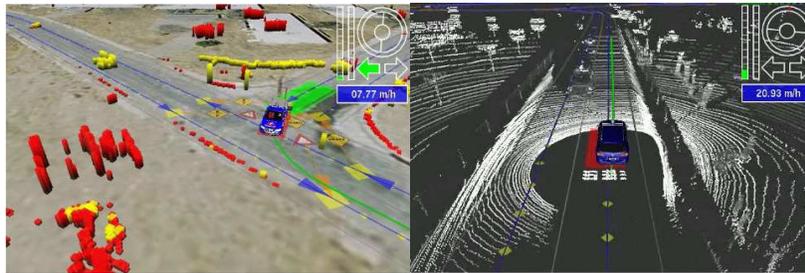


Fig. 2. Reconstrucciones 3D realizadas por el robot *Junior* en distintos momentos de la competición *DARPA Urban Challenge*

La fusión sensorial multi-robot tiene como objetivo mantener una estimación más precisa, estable y robusta del entorno, que la que puede conseguir un robot por sí solo. Esto se consigue gracias a la combinación de información procedente de otros robots. Existen dos grandes familias predominantes de algoritmos de fusión sensorial: Los probabilísticos y los no probabilísticos. Los métodos probabilísticos han demostrado, en general, ser los más apropiados para problemas de estimación debido a sus propiedades para modelar la incertidumbre en el estado de el/los objetos a estimar. Aunque a continuación repasaremos las características fundamentales de unos y otros, en [handbook08] se puede encontrar una revisión detallada de los algoritmos de ambas familias.

El núcleo de las técnicas probabilísticas es la regla de Bayes. Esta regla permite inferir el estado de un objeto  $x$  en base a una serie de observaciones  $z$  sobre él. La inferencia Bayesiana necesita que exista una relación entre  $x$  y  $z$  modelada como una función de densidad de probabilidad (en adelante FDP). La ecuación 1 muestra la regla de Bayes.

$$P(x|z) = \frac{P(z|x) * P(x)}{P(z)} \quad (1)$$

La potencia de la regla de Bayes estriba en que se pasa de calcular la probabilidad de estar en un estado  $x$  dada una observación  $z$ , a calcular la probabilidad de obtener la observación  $z$  suponiendo que se está en el estado  $x$ . Esta última probabilidad es más sencilla de calcular puesto que podemos construirnos a priori un *modelo de observación*. Este modelo se creará tomando observaciones en posiciones conocidas del entorno. De este modo, modelaremos la distribución de probabilidad del modelo de observación y tendremos una estimación de  $P(z|x)$ .

El conjunto de posibles posiciones donde el algoritmo de estimación puede concluir que se encuentra un objeto se denomina espacio de estados. Nuestro espacio de estados va a estar modelado por una FDP. El manejo eficiente de FDP's puede realizarse de varias maneras.

Una primera aproximación es emplear métodos analíticos, donde se define una fórmula parametrizada para expresar la función. Por ejemplo utilizando la media y la varianza de la estimación para modelar una distribución gaussiana bidimensional.

Otra aproximación es utilizar métodos discretizados para manejar las FDP's. En este caso la función de densidad de probabilidad se modela con una rejilla, donde cada celda almacena la probabilidad de que la observación se encuentre en el área abarcada por ella. En general, los métodos basados en rejillas suelen ser muy costosos computacionalmente. La razón de su elevado tiempo de cómputo es la constante iteración por todas sus celdas de los algoritmos, ocasionando tiempos de proceso exponenciales.

Una última aproximación comprende los métodos muestreados, donde la FDP es representada por un conjunto de muestras con un peso asociado. Este último enfoque evalúa un subconjunto del espacio de estados posibles. De esta manera el tiempo de cómputo es radicalmente menor y los resultados pueden ser muy similares. En este trabajo vamos a comparar dos implementaciones diferentes de algoritmos de fusión bayesiana: La primera de ellas basada en técnicas analíticas y la segunda basada en rejillas de probabilidad.

Los métodos probabilísticos no están exentos de problemas. Son sistemas complejos que necesitan manejar una gran cantidad de probabilidades para ser capaces de operar de manera adecuada y tienen problemas a la hora de asignar probabilidades en caso de desconocimiento o ambigüedad. Es por esto que han aparecido otras técnicas de fusión sensorial alternativas a las basadas en métodos probabilísticos. Las principales son el cálculo de intervalos, la lógica borrosa y la teoría de evidencias.

En el cálculo de intervalos se modela la incertidumbre sobre el conocimiento de cierto estado acotando los valores posibles entre un intervalo. Por ejemplo,  $x \in [a,b]$  describe el parámetro  $x$  entre los valores  $a$  y  $b$ . Ésta es la única asunción posible y ni siquiera se puede afirmar que los valores de  $x$  se distribuyen de manera uniforme dentro del intervalo. Varios intervalos acotados pueden combinarse con reglas básicas de manipulación como la suma, resta, multiplicación, división, aritmética de matrices, etc. En general, el cálculo de intervalos es recomendable cuando no tenemos o es difícil obtener un modelo probabilístico de observación, pero los sensores están correctamente caracterizados y su error es acotado. En [Moore1967] se puede encontrar una descripción detallada de esta técnica.

La lógica borrosa es otra alternativa a los métodos probabilísticos que ha alcanzado gran éxito en determinadas aplicaciones (especialmente las basadas en reglas). En los sistemas lógicos clásicos podemos definir una función de membresía que asocia un determinado elemento a un grupo o conjunto. Únicamente podemos decir si dicho elemento pertenece o no al grupo. En el ámbito de la lógica borrosa se dice que todo elemento tiene un grado de membresía con el grupo expresado entre 0 y 1. Así, un elemento puede pertenecer a varios grupos en mayor o menor grado. Por supuesto las reglas borrosas o difusas se pueden componer usando los mismos operadores que en la teoría de conjuntos clásica: Unión, intersección, negación, etc. La lógica borrosa se recomienda en sistemas controlados por un supervisor o cuando se quieren fusionar tareas de alto nivel. En [DuboisPrade80] se puede encontrar una amplia descripción de estos sistemas.

Los métodos basados en teoría de evidencias permiten modelar explícitamente la ignorancia para distinguir entre los posibles estados de una creencia, es decir, la ambigüedad. Ahora, el conjunto de estados posibles para una determinada variable estará formada por todas las combinaciones posibles de estados. Por ejemplo, para expresar la creencia

de si una determinada celda de un mapa está ocupada o vacía tendríamos los siguientes conjuntos de estados:

$$2^x = \{\{\text{ocupado, vacío}\}, \{\text{ocupado}\}, \{\text{vacío}\}, \emptyset\}$$

Cada uno de estos conjuntos de estados tendría asignado un valor que expresa posibilidad. El objetivo del estado  $\{\text{ocupado, vacío}\}$  sería modelar la ignorancia sobre si la celda está ocupada o vacía.

El uso de conjuntos de estados permite una descripción más rica de la representación de creencias. Por supuesto, el coste añadido es un incremento exponencial en la complejidad del sistema. Las técnicas basadas en razonamiento de evidencias han alcanzado cierto éxito en el dominio del razonamiento automático. En [Pagac98] se puede encontrar una descripción ampliada de esta técnica.

### **3 Percepción distribuida de la pelota**

Uno de los comportamientos habituales en el entorno de la RoboCup es buscar la pelota. Básicamente, esta tarea está compuesta por un conjunto de movimientos sobre la cabeza del robot que recorren el campo de juego. En un partido normal hay 10 robots jugando sobre el campo, lo que significa gran cantidad de objetos móviles que pueden provocar multitud de oclusiones sobre la pelota.

Uno de los objetivos de este trabajo es minimizar el tiempo que un robot emplea en localizar la pelota. Para ello usaremos una estimación compartida compuesta por todos los robots del equipo. En nuestro equipo [teamReport05], cada robot se posiciona en una determinada zona del campo según su rol de juego [aguero06sw]: Portero, defensa, chutador o delantero. Esto conlleva que tendremos un grupo de robots cubriendo una gran superficie del campo. Si no compartieran ningún tipo de información lo habitual sería que sólo uno o dos de los robots percibieran la pelota. El resto de miembros estarían desperdiciando un tiempo muy valioso buscándola.

Un equipo de trabajo real siempre usa sus miembros de una manera colaborativa. Nuestra propuesta es utilizar la infraestructura de comunicaciones presente en nuestro equipo para compartir la estimación de la pelota. Así, se mantendrá un estado global de la misma compartido por todos los miembros del equipo. La ventaja es clara: un robot podrá conocer la posición de la pelota incluso sin estar percibiéndola directamente.

Cada robot del equipo siempre mantiene una estructura de datos denominada LPS (*Local Perception System*). LPS almacena la estimación local a cada objeto relevante del juego (balizas, porterías y pelota). La estimación consiste principalmente en distancia y ángulo hasta el objeto desde el sistema de referencia solidario al robot.

Para poner en práctica nuestro enfoque colaborativo hemos incluido una nueva estructura de datos que almacena información compartida. Ahora, cada robot podrá consultar tanto su información local como la global fusionada. Cabe destacar que la información obtenida de la fusión de estimaciones entre varios robots siempre será un complemento a la estimación local.

La estimación compartida se va componiendo en cada robot a medida que se recibe información del resto de compañeros y de las propias observaciones. La información recibida de otro robot incluye la estimación de la pelota expresada como  $(x,y)$ , junto con la posición del observador, además de sus dos medidas respectivas de incertidumbre asociadas, una para la observación y otra para la posición del observador.

### **3.1 Fusión analítica de observaciones visuales**

La base de este método es modelar las observaciones locales como distribuciones gaussianas bidimensionales. El primer paso del proceso es la obtención de la observación por parte de cada robot. Esta observación consistirá en una imagen procedente de la cámara, que será procesada en diversas etapas por el software perceptivo, y que terminará con una estimación local en coordenadas polares, es decir, distancia y ángulo desde la posición del robot.

$$C_m = \begin{pmatrix} \theta_r^2 & 0 \\ 0 & \theta_\phi^2 \end{pmatrix} = \begin{pmatrix} (\alpha r)^2 & 0 \\ 0 & \beta^2 \end{pmatrix} \quad (2)$$

A continuación modelaremos esa observación con una función de densidad de probabilidad Gaussiana asociada a la estimación anterior incorporándole incertidumbre radial y angular, según el modelo de observación previamente calculado. La incertidumbre será representada como una matriz de covarianza similar a la mostrada en la ecuación (2), donde  $(\alpha, \beta)$  representan el error en distancia y ángulo respectivamente. Nuestros experimentos sobre el modelo de observación han revelado que el error en distancia es lineal ( $\alpha*r$ ) y el error en ángulo puede ser considerado constante ( $\beta$ ). La ecuación (3) muestra la fórmula de la FDP bidimensional para una distribución gaussiana, donde  $C'$  es la matriz de covarianzas,  $\mu_1$  es la matriz de medias y  $S$  la matriz de estados  $(x,y)$ .

$$\rho(Ob_s) = \frac{1}{2\pi\sqrt{|C'|}} e^{-\frac{1}{2}(S-\mu_1)^T C'^{-1}(S-\mu_1)}, \quad S = \begin{pmatrix} x \\ y \end{pmatrix}, \mu_1 = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \quad (3)$$

siendo  $C'$  = matriz de covarianzas de la observación  $P(\text{pelota} | \text{robot})$

El siguiente paso del algoritmo consiste en construir la FDP asociada a la posición del observador. La fuente de información aquí es el algoritmo de auto-localización presente en el robot. La salida de este algoritmo es una posición  $(x,y,\theta)$  y una matriz de covarianza para modelar la incertidumbre en las tres dimensiones. La ecuación 4 determina la FDP para nuestra estimación de la posición del observador, donde  $C''$  es la matriz de covarianzas,  $\mu_2$  es la matriz de medias y  $M$  la matriz de estados  $(z,t,r)$  para representar la posición y orientación del robot.

$$\rho(Loc) = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{|C''|}} e^{-\frac{1}{2}(M-\mu_2)^T C''^{-1}(M-\mu_2)}, \quad M = \begin{pmatrix} z \\ t \\ r \end{pmatrix}, \mu_2 = \begin{pmatrix} \mu_z \\ \mu_t \\ \mu_r \end{pmatrix} \quad (4)$$

Siendo  $C''$  = matriz de covarianzas de la posición del robot

El siguiente paso del algoritmo persigue transformar a forma canónica la función de densidad de probabilidad de la observación e incorporar al observador. Con el fin de combinar las estimaciones procedentes de otros robots, necesitamos tener todas las distribuciones de probabilidad en un marco común, esto es, en un sistema de referencia global. Para ello en este paso hay que incorporar la incertidumbre del observador. Sin embargo, hemos encontrado que esto no tiene una formulación cerrada, pues desde la ecuación 5 no podemos llegar fácilmente a la fórmula de otra distribución gaussiana.

$$\rho(\text{pelota}_{xy}) = \int_z \int_t \int_r P(\text{pelota}_{xy} | \text{robot}_{ztr}) P(\text{robot}_{ztr}) dr dt dz \quad (5)$$

Dejando a un lado de momento la incertidumbre de posición del observador, es decir, asumiendo que se conoce su posición y orientación sin error abordamos la última etapa del algoritmo: la fusión entre varios robots. La estimación global de la pelota es el resultado de combinar dos matrices de covarianza (incertidumbre de cada estimación) y dos estimaciones en posición, según muestran la ecuación (8) para el caso de dos robots. Se ha utilizado un enfoque probabilístico basado en la regla de Bayes para la fusión de las estimaciones. Con las matrices de covarianza y posición obtenidas, podremos combinar a su vez estimaciones procedentes de otros robots en el mismo instante temporal, gracias a la propiedad asociativa presente en esta formulación.

De la estimación compartida se extraerá el estado de la pelota y su incertidumbre. El estado almacenado será la posición  $(x,y)$  en el campo de juego y su incertidumbre nos dará una idea de la precisión en la estimación. Las pruebas experimentales mostrarán cómo de efectiva es esta estimación comparada con las estimaciones locales individuales de cada robot.

$$\begin{aligned} C &= C_1 - C_1[C_1 + C_2]^{-1}C_1 \\ \hat{X} &= \hat{X}_1 + C_1[C_1 + C_2]^{-1}(\hat{X}_2 - \hat{X}_1) \end{aligned} \quad (8)$$

La figura 3 muestra varios ejemplos de la estimación distribuida generada según las observaciones de tres robots. En la figura las observaciones locales de cada robot están representadas por elipses naranjas. El tamaño de esa elipse representa la incertidumbre de la observación. La elipse de color azul es el resultado de la fusión de estimaciones locales. Los puntos negros son puntos de control y representan posiciones reales de la pelota en el campo.

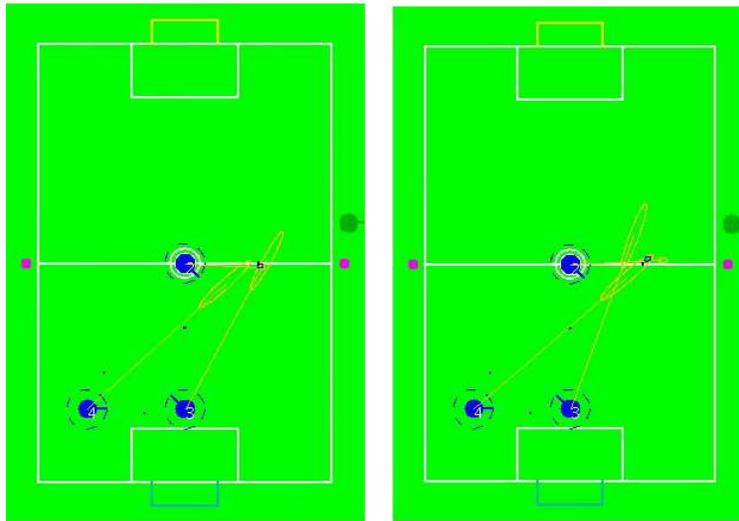
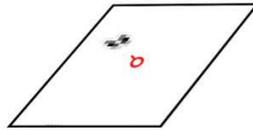


Fig. 3. Mezcla de observaciones (elipse azul) procedente de tres estimaciones locales (elipse naranja)

### 3.2 Fusión de rejillas probabilísticas

Este segundo método basado en rejillas requiere que discreticemos el campo de juego en celdas de igual tamaño. En nuestro caso hemos configurado el tamaño de celda a 15cm. x 15cm. La primera fase del algoritmo es similar al método anterior: Obtención de la percepción local en coordenadas polares y generación de la matriz de covarianzas para modelar la incertidumbre gracias al modelo de observación.

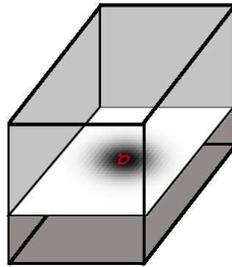
El siguiente paso consiste en construir la rejilla de probabilidad de la observación. La ecuación 3 muestra la fórmula de la función de densidad de probabilidad para una distribución gaussiana de dos dimensiones. En nuestro caso asumiremos esta formulación debido a que nuestro modelo de observación puede aproximar la incertidumbre de la observación a una distribución gaussiana. Para cada celda de nuestra rejilla *obs* calcularemos el valor de esta FDP obteniendo un resultado similar al de la figura 4.



**Observation**

Fig. 4. Rejilla probabilística asociada a una observación

A continuación necesitamos modelar la FDP de la posición del observador. Para ello utilizaremos un cubo de probabilidad *loc*, donde cada plano horizontal estará asociado con una posible orientación del robot. El algoritmo de auto-localización proporcionará una terna  $(x,y,\theta)$ , junto con una matriz de covarianzas para expresar la incertidumbre en  $(x,y)$ . Para nuestras pruebas hemos considerado un error en orientación modelado como una distribución gaussiana de desviación típica constante. De manera similar al paso anterior, obtenemos todos los parámetros de la FDP asociada a la autolocalización y evaluamos esta función en el centro de cada celda de la rejilla *loc* obteniendo el resultado mostrado en la figura 5.



**Localization**

Fig. 5. Rejilla probabilística asociada a la posición del observador

La situación actual es que disponemos de dos rejillas: *obs* mantiene la FDP local de la observación y *loc* contiene la FDP en coordenadas globales de la posición del observador. Con la operación de convolución descrita en (9) podemos combinar estas dos rejillas para crear una tercera y definitiva denominada *ball\_local*, que contiene la PDF de la observación teniendo en cuenta las incertidumbres en la posición del observador como la de la propia observación. La figura 6 ilustra el proceso de convolución entre las rejillas *obs* y *loc*.

$$\rho(\text{pelota}) = \sum_z \sum_t \sum_r P(\text{pelota}_{xy} | \text{robot}_{ztr}) P(\text{robot}_{ztr}) \tag{9}$$

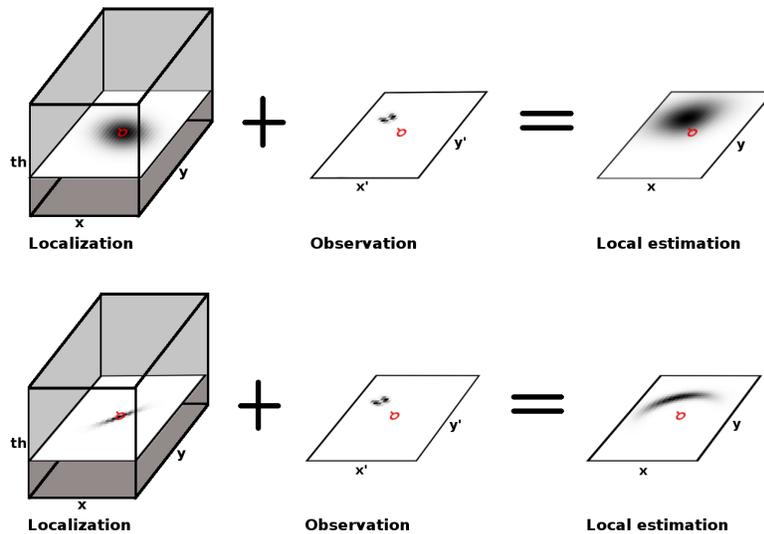


Fig. 6. Proceso de fusión entre la observación local y la auto-localización

Cada robot deberá actualizar periódicamente todas sus rejillas y comunicar al resto *ball\_local*. El proceso de fusión consiste en aplicar la regla de Bayes para combinar las rejillas procedentes del resto de miembros del equipo con la generada localmente. La ecuación 10 muestra el desarrollo matemático de la combinación bayesiana de dos observaciones independientes. Se pretende calcular el valor de *c* dados las observaciones

$a$  y  $b$  desde robots diferentes. En las operaciones indicadas con \* y \*\* se asume independencia de las observaciones. El resultado final demuestra que basta con multiplicar las dos observaciones y dividir entre  $P(c)$ , que corresponde con la probabilidad a priori.

$$\begin{aligned}
 P(c | a, b) &= \frac{P(c | a)P(b | c, a)}{P(b | a)} \\
 P(b | c, a) &= P(b | c)^* = \frac{P(c | b)P(b)}{P(c)} \\
 P(c | a, b) &= \frac{P(c | a)P(c | b)P(b)**}{P(b | a)**P(c)} = \frac{P(c | a)P(c | b)}{P(c)} \quad (10)
 \end{aligned}$$

Utilizando el anterior desarrollo matemático actualizaremos la rejilla *ball\_shared* con la estimación distribuida. Para cada celda de *ball\_shared* obtendremos su valor multiplicando esa misma celda en todas las *ball\_local* recibidas y dividiendo por una constante.

En la figura 7 podemos observar un ejemplo de rejillas probabilísticas correspondientes a la estimación de la pelota. En la imagen se puede advertir la rejilla resultante de combinar dos rejillas locales.

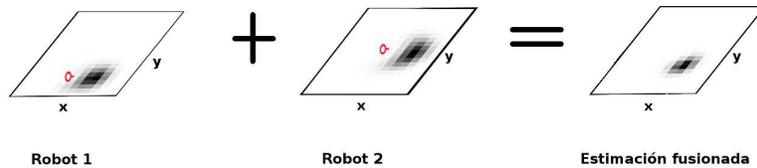


Fig. 7. Resultados de combinar dos rejillas de probabilidad con la estimación de la pelota

## 4 Experimentos

Para validar nuestro trabajo hemos realizado una serie de experimentos utilizando un equipo de tres robots aiBo. Entre las características principales de este robot destacan la incorporación de una cámara y un sensor de infrarrojos en su cabeza, sensores de contacto en las patas, un acelerómetro interno y micrófono. Los actuadores principales permiten mover cada pata y la cabeza con tres grados de libertad cada uno. El robot carece de disco duro y en su lugar se emplea un tarjeta de memoria de tipo *Memory Stick* donde residen el sistema operativo y los programas de usuario.

Por último, también dispone de red inalámbrica, dotando al robot de capacidades comunicativas. El software incluido incorpora una pila de comunicaciones TCP/IP y se ha desarrollado un protocolo adhoc de comunicación entre los robots. En nuestro caso esta infraestructura se utiliza para intercambiar información sobre las percepciones de la pelota. En la figura 8 se pueden apreciar los principales sensores de aiBo y un instante de las pruebas sobre en el laboratorio.

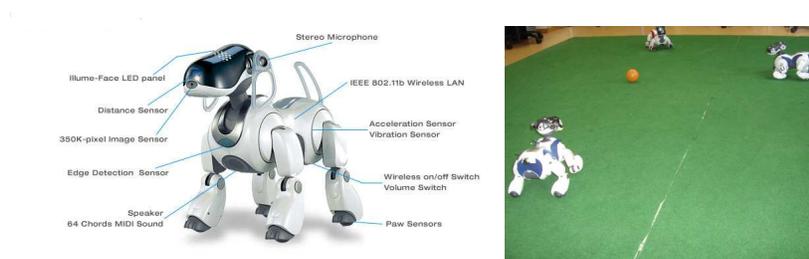


Fig. 8. Robot aiBo de Sony con sus principales sensores (izquierda) y escenario de las pruebas (derecha)

Los experimentos que vamos a describir en esta sección están organizados en varios apartados, según el objetivo de la prueba y los parámetros a medir. Los objetivos principales serán obtener medidas de precisión, estabilidad y comportamiento ante situaciones anómalas como oclusiones o falsos positivos. Para cada prueba expondremos los resultados obtenidos con las dos implementaciones realizadas y se extraerán las conclusiones oportunas.

#### 4.1 Estimación compartida Vs Estimación local

Una primera prueba consiste en comparar los dos tipos de estimaciones, las locales y la compartida. La percepción compartida será única a cada instante, mientras que estimaciones locales habrá tantas como robots diseminados por el entorno. Para conseguir una comparación justa, vamos a medir en esta prueba la media del error en todas las estimaciones locales con la estimación compartida.

El montaje del experimento es el mostrado en la figura 9 donde podemos observar tres robots que perciben la pelota en una posición del campo. Su campo visual aparece sombreado en la figura y la pelota es el objeto de color naranja. En nuestro laboratorio hemos realizado pruebas con más posiciones de la pelota y los resultados numéricos obtenidos están resumidos en la tabla 1.

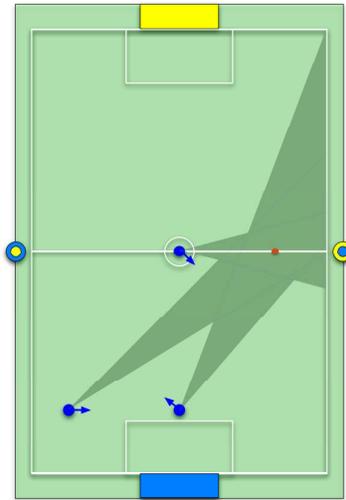


Fig. 9. Montaje experimental para pruebas de precisión

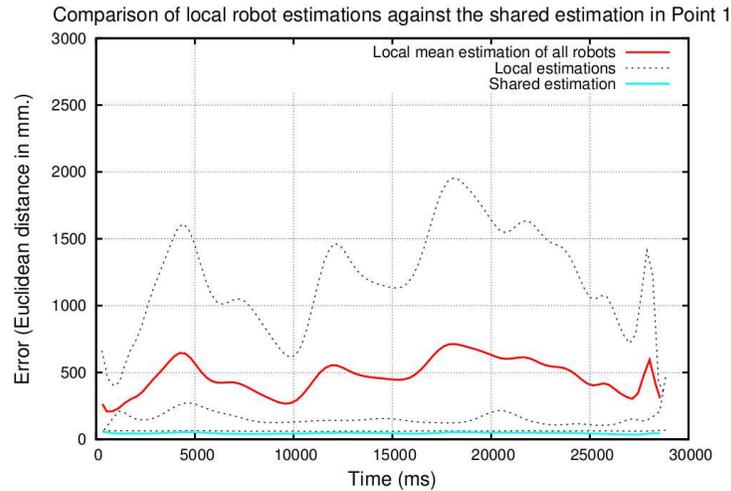


Fig. 10. Error cometido en estimación local y compartida en los diferentes puntos del experimento

Experimento #	Error medio local	Error medio compartido	Desviación típica local	Desviación típica compartida
1	474.08	46.39	786.66	10.24
2	66.03	47.86	44.20	3.88
3	159.85	65.28	187.21	6.09
4	131.95	52.62	94.34	4.64

Tabla 1. Datos obtenidos de comparar todas las estimaciones en los diferentes puntos del experimento

En la figura 10 las líneas discontinuas corresponden a las estimaciones locales de cada robot. La línea roja representa la media del error de todas las estimaciones locales. A su vez, la línea de color turquesa representa el error de la estimación compartida. Cuando hablamos de error nos referimos a la distancia en milímetros entre la estimación propuesta y la posición real de la pelota. Como se puede ver en la figura la estimación distribuida es muy similar a la mejor estimación local y comete menor error que la media de las estimaciones locales, al menos cuando la posición del observador es conocida.

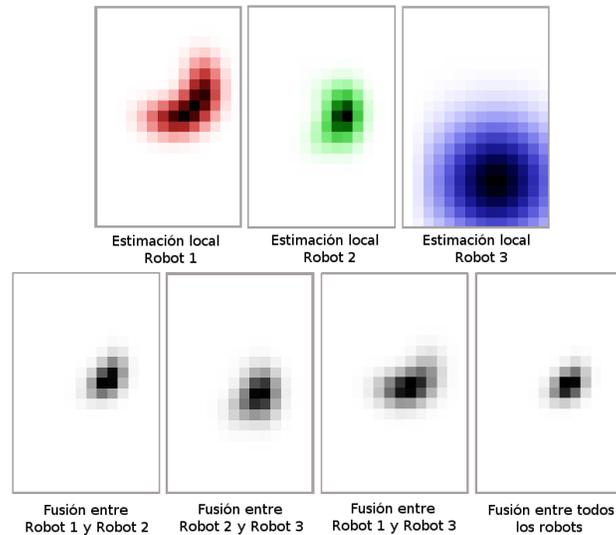


Fig. 11. Diferentes rejillas locales y combinadas entre varios robots

La figura 11 (parte superior) muestra tres estimaciones locales de la pelota en colores verde, rojo y azul respectivamente usando la implementación basada en rejillas. En este caso las estimaciones de la pelota si que incorporan la incertidumbre tanto de la observación como de la posición del observador. La estimación de los robots 1 y 2 es parecida, mientras que la del robot 3 cuenta con mayor incertidumbre. En la figura 11 (parte inferior) se observan las rejillas resultantes de combinar distintas estimaciones dos a dos o las estimaciones de los tres robots simultáneamente. Como era de esperar, las rejillas obtenidas tienden a parecerse más a las observaciones con menor incertidumbre.

Los resultados de este experimento arrojan datos favorables para la estimación compartida confirmando las expectativas sobre las ventajas de fusionar. Podemos afirmar que la fusión sensorial es positiva en ambas implementaciones y produce una reducción del error en la estimación de la pelota. Además, la estimación es más estable y exenta de fluctuaciones como suele ocurrir con algunas estimaciones locales. Los robots que perciben la pelota a largas distancias son los principales beneficiados de la nueva estimación, pues su precisión e incertidumbre se ven mejoradas drásticamente.

## 4.2 Tolerancia a errores en la estimación

El objetivo de este experimento es constatar la tolerancia del algoritmo de fusión Bayesiana ante errores en la posición del observador. Para ello hemos modificado artificialmente la posición de uno de los robots tanto en distancia como en ángulo respecto de su posición real. La figura 12 muestra las diferentes posiciones que se han utilizado en el experimento. Los robots que tienen un color más suave representan las diferentes posiciones que ha ido tomando el observador. En la figura de la izquierda se muestran las transiciones en posición y en la figura de la derecha los cambios en orientación.

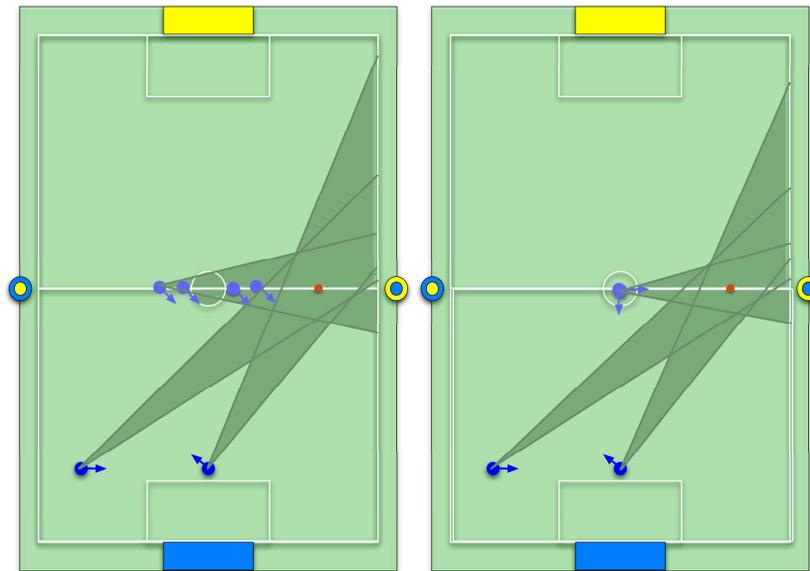


Fig. 12. Montaje experimental para pruebas de tolerancia a errores

La gráfica 12 muestra en colores sólidos el error de la estimación compartida ante diferentes valores en la posición del robot 1. La línea punteada representa el error de la estimación compartida con el robot 1 en su posición original. Por su parte, la gráfica 13 representa en colores sólidos el error de la estimación compartida ante diferentes valores en la orientación del robot 1. De nuevo, la línea punteada muestra el error de la estimación compartida con el robot 1 en su posición original.

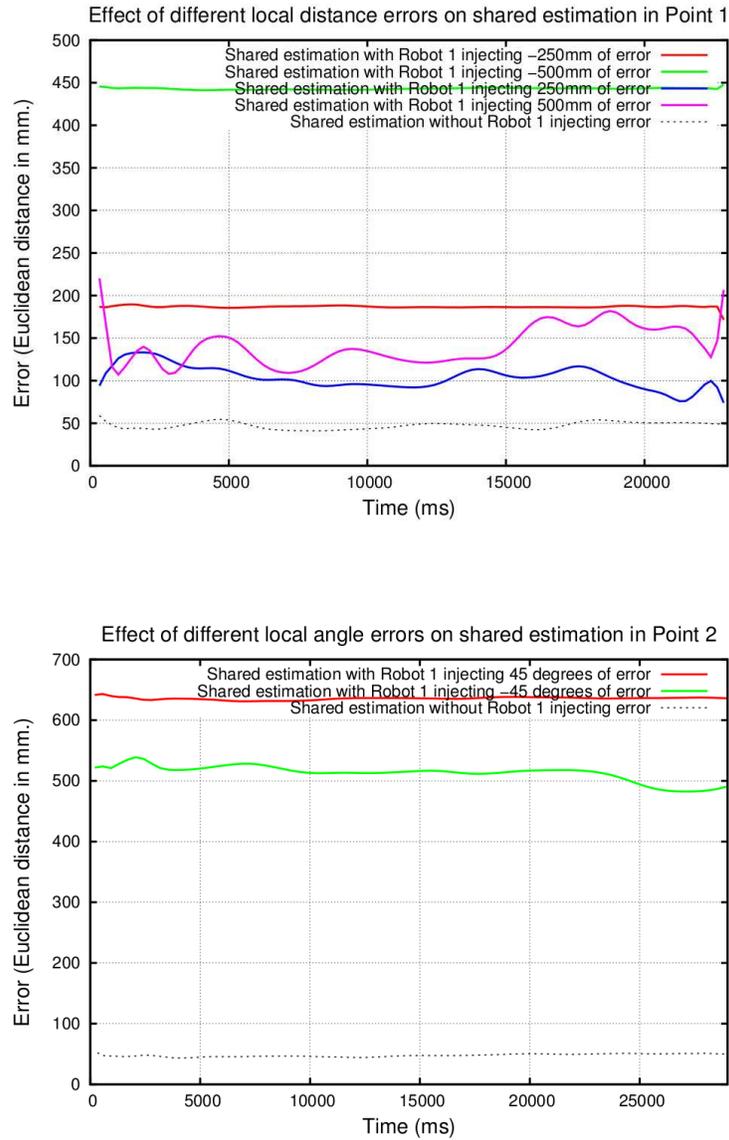


Fig. 13. Resultados obtenidos ante errores en la estimación

Los experimentos muestran que si forzamos mayor error en la posición del observador que está teniendo más peso en la decisión (robot 1), la calidad de la estimación se deteriora bastante. Las pruebas realizadas afectan al robot que mejor observación tiene y, por tanto, mayor influencia sobre la estimación compartida. Es por esto que el deterioro en su observación arrastra a la estimación distribuida. Los datos reflejan un aumento considerable del error, incluso cerca de la media de error local. Si además forzamos un error en la orientación del mejor observador, los datos empeoran todavía más.

En general, podemos concluir que el mecanismo de fusión bayesiano tolera bien la combinación de cualquier FDP independientemente de si la estimación tiene una alta o baja incertidumbre. Lógicamente cuanto menor sea la incertidumbre de las FDP's a combinar, mejor será la estimación compartida. Sin embargo, si los modelos de observación y los algoritmos de localización no reflejan adecuadamente la realidad, la estimación puede degradarse bastante y no ajustarse a los valores esperados.

### 4.3 Comportamiento ante oclusiones en uno o varios robots

El objetivo de este experimento es simular que uno o varios robots no son capaces de percibir la pelota, para comprobar su impacto en la estimación compartida. En la figura 14 podemos apreciar diferentes pruebas limitando el rango perceptivo de uno de los robots. Los círculos negros representan barreras que impiden a los robots percibir la pelota.

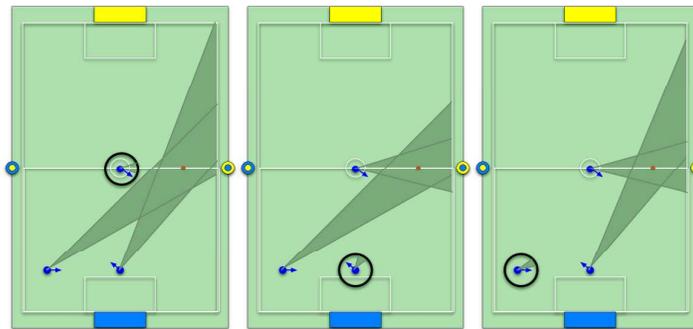


Fig. 14. Montaje experimental para pruebas de oclusiones en las observaciones de un robot

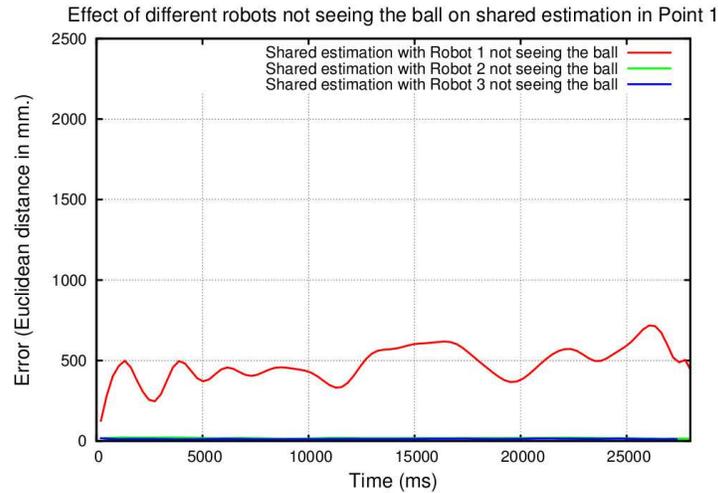


Fig. 15. Resultados obtenidos ante oclusiones en un robot

Los datos muestran que el impacto en la estimación compartida es relativo a qué robot sufre las oclusiones. En el caso del robot con mejor calidad en la observación se produce un gran deterioro en la estimación. Sin embargo, este es el peor de los casos y el error en la estimación compartida es similar al error medio local. Para el resto de situaciones la compartición mejora notablemente la media de error local, aún en presencia de oclusiones en un robot.

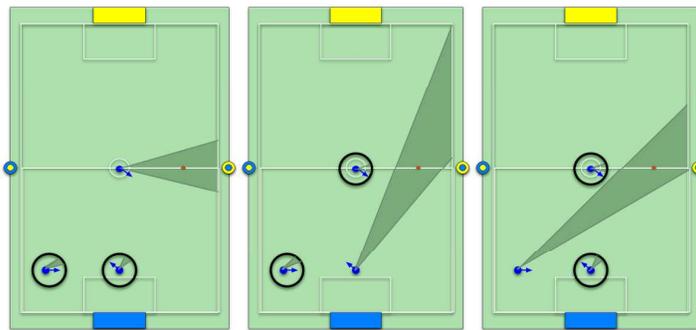


Fig. 16. Montaje experimental para pruebas de oclusiones en las observaciones de dos robots

La figura 16 muestra otro experimento donde limitamos el campo visual de dos robots. En este caso la estimación compartida es prácticamente la misma que la observación local del robot que visualiza la pelota. Dependiendo del robot que sea capaz de ver la pelota tendremos mejor o peor calidad.

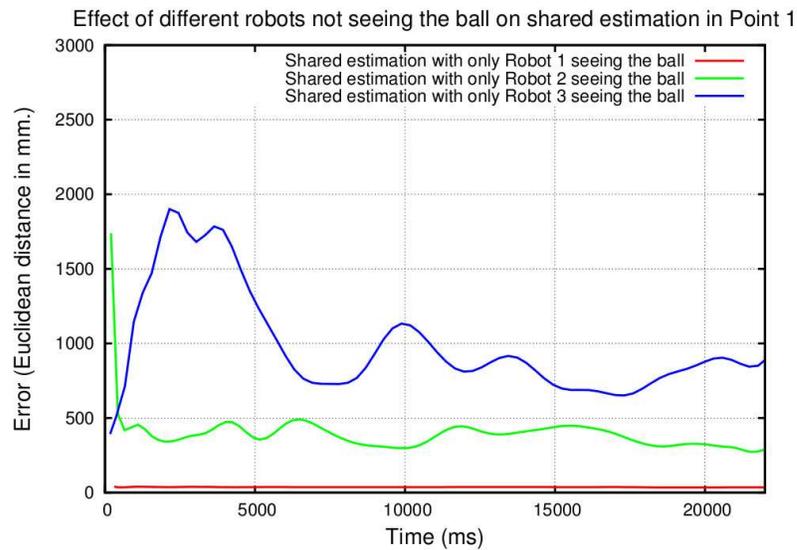


Fig. 17. Resultados obtenidos ante oclusiones en dos robots

En el experimento basado en rejillas probabilísticas intervienen dos robots. En la figura 18 se muestran las dos estimaciones de los robots 1 y 2 respectivamente. El robot 2 no percibe la pelota y, por tanto, su rejilla muestra todas las celdas con la misma probabilidad. La rejilla combinada es prácticamente la misma que proporcionó el robot 1. La ventaja para el robot 2 es infinita, pues ha pasado de no saber dónde podría estar la pelota a conocer su posición con relativa certeza.

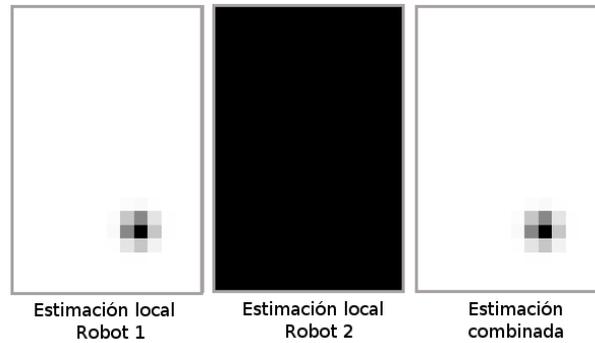


Fig. 18. Resultados obtenidos ante oclusiones en dos robots usando rejillas

Las pruebas demuestran que ante oclusiones existe una gran ventaja utilizando el enfoque colaborativo: los robots que no perciben directamente la pelota, ahora si son capaces de mantener una estimación de ella. Este aporte supone un gran avance en estos robots y queda patente en ambas implementaciones.

#### 4.4 Comportamiento ante falsos positivos

El objetivo de este experimento es medir cómo afecta un falso positivo a la estimación compartida. Para ello, como podemos apreciar en la figura 19 hemos situado una segunda pelota en el campo. Esta segunda pelota únicamente es visible por uno de los robots. Hemos realizado dos experimentos en la variante analítica del algoritmo. En el primero de ellos la estimación de la segunda pelota tiene baja incertidumbre debido a que está a escasa distancia del observador. En la segunda prueba la pelota está situada a una distancia media-alta y, por tanto, su incertidumbre es mayor.

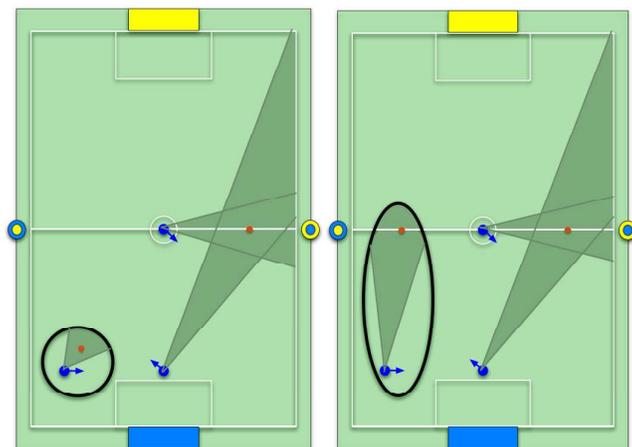


Fig. 19. Montaje experimental para pruebas de falsos positivos

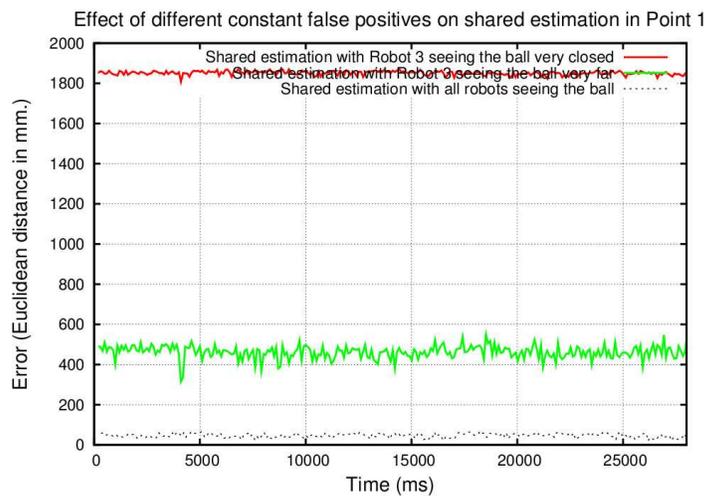


Fig. 20. Resultados obtenidos ante un falso positivo con alta y baja incertidumbre

El experimento para la implementación con rejillas usa dos robots con estimaciones no compatibles. Cada uno de ellos está percibiendo la pelota en lugares distintos y, como se observa en la figura 21 la estimación combinada concluye que la pelota está en un punto intermedio.

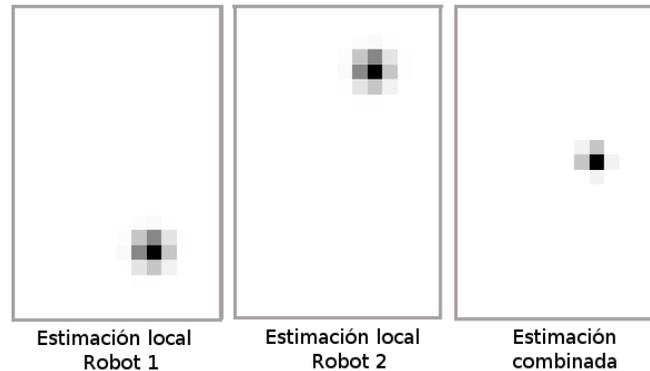


Fig. 21. Resultados obtenidos ante un falso positivo utilizando rejillas probabilísticas

Las pruebas de precisión muestran que los falsos positivos son el peor enemigo de las técnicas bayesianas. El resultado de combinar dos estimaciones geoméricamente no compatibles es una tercera situada a medio camino de las dos anteriores. Éste es el peor escenario posible que se puede dar y el error aumenta con la incertidumbre asociada al falso positivo.

## 5 Conclusiones

A la vista de los experimentos podemos concluir que el mecanismo de percepción distribuida mejora sustancialmente la precisión a la hora de estimar la pelota. Esta mejora tiene dos pilares principales: Disminución del error y reducción de la incertidumbre.

La gran estabilidad o robustez de la estimación se consigue gracias a la fusión Bayesiana. Este método tolera muy bien las malas aportaciones de

determinados miembros, siempre que exista alguna estimación de buena calidad. Sin embargo, también es reseñable que el método no se comporta adecuadamente ante situaciones de falsos positivos con baja incertidumbre. Ante dos observaciones de buena calidad no compatibles geoméricamente, los métodos bayesianos suelen concluir que la estimación se encuentra en punto intermedio de las dos observaciones con baja incertidumbre.

En cuanto a las desventajas poco tenemos que decir porque este mecanismo no sustituye a los sistemas de percepción local, es un complemento. Y como complemento los únicos aspectos negativos que podemos destacar son el consumo extra tanto en tiempo de CPU como en ancho de banda de comunicaciones.

Existe un detalle a destacar y es que el hecho de tener ahora dos estimaciones para la pelota (la estimación local y la global) aumenta las posibilidades de decisión en los comportamientos. Este hecho puede interpretarse como algo positivo si suponemos que la estimación local está equivocada y optamos por tener en cuenta la estimación compartida. Sin embargo, también podría darse algún caso en que sucediera lo contrario. Por tanto, simplemente concluiremos que podría provocar más dudas a la hora de elegir, aunque nuestra opinión es que contar con más información siempre es positivo.

En cuanto a las diferentes implementaciones del algoritmo de fusión bayesiana podemos extraer algunas conclusiones. El método analítico descrito presenta una gran ventaja sobre la implementación basada en rejillas: el tiempo de cómputo. En la figura 22 ilustramos esta comparativa y cabe destacar que las operaciones matriciales son muy rápidas en este método, no así las operaciones de convolución que son necesarias en el método de rejillas.

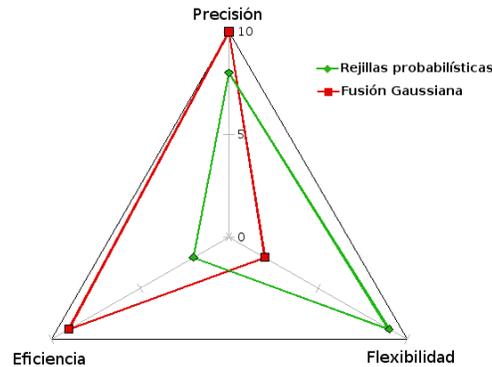


Fig. 22. Gráfico comparativo entre la fusión gaussiana y la fusión de rejillas probabilísticas

En cuanto a la precisión obtenida, el método basado en rejillas probabilísticas podría llegar teóricamente a las mismas resoluciones que su hermano analítico. Sin embargo, lo que sucede en la realidad es que si aumentamos la resolución de las celdas los tiempos de cómputo se vuelven inabordables.

Hemos querido hacer referencia a otro parámetro en la comparación de los algoritmos: la flexibilidad. Con él queremos expresar la permisividad del algoritmo a modelar distintas configuraciones de incertidumbre (por ejemplo multimodalidad) o facilidad para incorporar incertidumbre procedente de otras fuentes (por ejemplo de la auto-localización). En este apartado la variante analítica está claramente en desventaja pues uno de sus requisitos es operar exclusivamente con distribuciones gaussianas. Así, no podemos modelar situaciones donde aparezcan varios objetos. Además, como hemos podido comprobar en el método basada en rejilla, la incorporación a la observación de la incertidumbre del observador no es fácilmente modelable con una distribución gaussiana.

Una posible línea futura a desarrollar tiene que ver con el hecho de que una estimación global compartida puede utilizarse como si de una baliza extra se tratara. De esta manera, utilizando los parámetros de distancia y ángulo hasta la pelota se pueden realimentar los algoritmos de auto-localización incorporando una nueva observación.

Finalmente, los métodos propuestos en este trabajo no son otra cosa que el camino recorrido hasta la fecha en el campo de la estimación distribuida. Los siguientes pasos a estudiar son la utilización de filtros de Kalman o técnicas muestreadas basadas en Monte Carlo para seguir afinando esta técnica y paliar las limitaciones encontradas.

Otra línea futura abordable es incorporar continuidad temporal a las estimaciones. Hasta el momento las observaciones fusionadas tienen un carácter instantáneo y no hay *memoria* de estimaciones pasadas. Este paso supondría un avance en la estabilidad de la nueva observación compartida.

## Referencias

- [waf06urjc] C. E. Agüero, F. Martín, H. Martínez, and V. Matellán. Communications and basic coordination of robots in TeamChaos. In Actas VII Workshop de Agentes Físicos, pages 3-9, 2006.
- [aguero06sw] C. E. Agüero, V. Matellán, V. Gómez, and J. Cañas. Switch! Dynamic roles exchange among cooperative robots. In Proceedings of the 2nd International Workshop on Multi-Agent Robotic Systems - MARS 2006, pages 99-105. INSTICC Press, 2006.
- [canovas.rcup04] J. Canovas, K. LeBlanc, and A. Saffiotti. Robust multi-robot object localization using fuzzy logic. In Proc. of the Int. RoboCup Symposium, Lisbon, PT, 2004.
- [cao97cooperative] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7-23, March 1997.
- [DuboisPrade80] D. Dubois and H. Prade. *Fuzzy sets and systems - Theory and applications*. Academic press, New York, 1980.
- [Gerkey04role] B. Gerkey and M. Mataric. On role allocation in RoboCup. In *RoboCup 2003: Robot Soccer World Cup VII*, 2004, pages 43-53. Springer-Verlag, 2004.
- [robocup] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *ICJAI-95 - Workshop on Entertainment and AI/ALIFE*, 1995.

- [mezzanine] U. R. R. Lab. Mezzanine, 2007. <http://playerstage.sourceforge.net/mezzanine/>.
- [Lerman05] K. Lerman, C. V. Jones, A. Galstyan, and M. J. Mataric. Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, 25(4):225-242, 2006.
- [Montemerlo08a] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Ho\_mann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrowskaya, M. Pueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 2008.
- [Moore1967] R. E. Moore. *Interval Analysis (Automatic Computation S.)*. Prentice Hall, 1967.
- [Pagac98] D. Pagac, E. M. Nebot, and H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4):623-629, 1998.
- [handbook08] B. Siciliano and O. Khatib, editors. *Springer Handbook of Robotics*. Springer, 2008.
- [sony] Sony Global - AIBO Global Link. <http://www.sony.net/Products/aibo/>.
- [aibo] Sony. Sony AIBO SDE and Open-R SDK. <http://openr.aibo.com>, 2006.
- [StroupeDistr] A. Stroupe, M. C. Martin, and T. Balch. Distributed sensor fusion for object position estimation by multi-robot systems. In *IEEE International Conference on Robotics and Automation*, May, 2001. IEEE, May 2001.
- [teamchaos] TeamChaos. Team report 2005, 2006. Online at [www.teamchaos.es](http://www.teamchaos.es).
- [robocup] The Robocup Federation. Robocup Four-Legged League. <http://www.tzi.de/spl/bin/view/Website/WebHome>.
- [probRobotics] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, September 2005.
- [urban] DARPA Urban Challenge. <http://www.darpa.mil/grandchallenge/>.