



Universitat d'Alacant

Universidad de Alicante

Esta tesis doctoral contiene un índice que enlaza a cada uno de los capítulos de la misma.

Existen asimismo botones de retorno al índice al principio y final de cada uno de los capítulos.

[Ir directamente al índice](#)

Para una correcta visualización del texto es necesaria la versión de [Adobe Acrobat Reader 7.0](#) o posteriores

Aquesta tesi doctoral conté un índex que enllaça a cadascun dels capítols. Existeixen així mateix botons de retorn a l'índex al principi i final de cadascun dels capítols .

[Anar directament a l'índex](#)

Per a una correcta visualització del text és necessària la versió d' [Adobe Acrobat Reader 7.0](#) o posteriors.



Universidad de Alicante
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Universitat d'Alacant
Universidad de Alicante

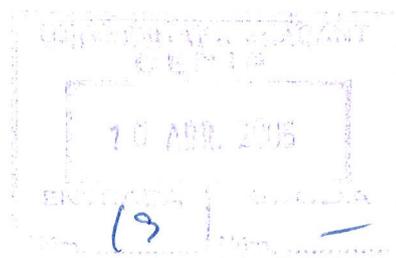


TESIS DOCTORAL

FSR-BAY: Modelo Probabilístico para la
Fusión Sensorial Robótica

Autor

Fidel Aznar Gregori



Directores

Ramón Rizo Aldeguer

Mar Pujol López



Abril de 2006



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

Agradecimientos

Mi mas sincero agradecimiento al Departamento de Ciencia de la Computación e Inteligencia Artificial y muy especialmente a todos mis compañeros del grupo de investigación I3A. Concretamente quiero resaltar la labor de mis directores de tesis, Ramón Rizo y Mar Pujol por su inestimable ayuda y comprensión, a Faraón Llorens por su constante apoyo y a mis compañeros Mireia Sempere, Rafael Molina, Carlos Villagrà y Paco Mora por sus aportaciones a este trabajo.

A mi familia (los que están y los que no), en especial a mis padres y hermana por el cariño y apoyo prestado.

A Ruth, por entender que aunque le quitase un poco de tiempo para hacer esta tesis le quiero igual.



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant

A mis padres y hermana

A Ruth



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

Resum

Els humans i els animals han evolucionat desenvolupant la capacitat d'utilitzar els seus sentits per a sobreviure. La fusió sensorial, que és un dels pilars d'aquesta evolució, es realitza de forma natural per animals i humans per a aconseguir una millor interacció amb l'entorn circumdant. L'emergència de nous sensors, tècniques de processament avançat, i maquinari de procés millorat, han fet viable la fusió de molt tipus de dades. Actualment els sistemes de fusió sensorial s'han utilitzat de manera extensiva per al seguiment d'objectes, identificació automàtica, raonament, etc. A banda de moltes altres àrees d'aplicació (com la monitorització de sistemes complexos, el control automàtic de fabricació industrial. . .) les tècniques de fusió també s'utilitzen en el camp de la intel·ligència artificial i la robòtica.

Aquesta tesi aporta el model FSR-BAY, per a la fusió sensorial robòtica. Aquest model té en compte alguns aspectes que des del nostre punt de vista han sigut tractats de manera secundària per la majoria de les arquitectures de fusió actuals: la informació incompleta i incerta, les capacitats d'aprenentatge i l'utilitzar una representació homogènia de la informació, independent del nivell de fusió. També es proporcionen dos casos d'estudi del model proposat aplicat a un agent autònom. El primer cas tracta la fusió cooperativa de la informació utilitzant per a fusionar informació provinent de diversos sensors d'un mateix tipus. El segon cas fusiona de manera competitiva informació tant heterogènia com homogènia.



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

Resumen

Los humanos y los animales han evolucionado desarrollando la capacidad de utilizar sus sentidos para sobrevivir. La fusión sensorial, que es uno de los pilares de esta evolución, se realiza de forma natural por animales y humanos para conseguir una mejor interacción con el entorno circundante. La emergencia de nuevos sensores, técnicas de procesamiento avanzado, y hardware de proceso mejorado, han hecho viable la fusión de muchos tipos de datos. Actualmente los sistemas de fusión sensorial se han utilizado de manera extensiva para el seguimiento de objetos, identificación automática, razonamiento, etc. Aparte de otras muchas áreas de aplicación (como la monitorización de sistemas complejos, el control automático de fabricación industrial...) las técnicas de fusión también se utilizan en el campo de la inteligencia artificial y la robótica.

Esta tesis aporta el modelo FSR-BAY, para la fusión sensorial robótica. Este modelo tiene en cuenta algunos aspectos que desde nuestro punto de vista han sido tratados de manera secundaria por la mayoría de las arquitecturas de fusión actuales: la información incompleta e incierta, las capacidades de aprendizaje y el utilizar una representación homogénea de la información, independiente del nivel de fusión. También se proporcionan dos casos de estudio del modelo propuesto aplicado a un agente autónomo. El primer caso trata la fusión cooperativa de la información utilizando para fusionar información proveniente de varios sensores de un mismo tipo. El segundo caso fusiona de manera competitiva información tanto heterogénea como homogénea.



Universitat d'Alacant
Universidad de Alicante



Universitat d'Alacant
Universidad de Alicante

Abstract

Throughout their evolution, both humans and animals have developed the capacity to use their senses to help them to survive. One of the pillars of this evolution; sensory fusion, is achieved naturally by animals and humans to obtain the best possible interaction with the surrounding environment. In the field of computers the emergence of new sensors, advanced processing techniques, and improved hardware have made possible the fusion of many different types of data. Nowadays, sensory fusion systems have been used extensively to follow objects, for automatic identification, reasoning, etc. Apart from the many other areas of application (such as the motorization of complex systems, the automatic control of industrial fabrication processes.) fusion techniques are also being used in the fields of artificial intelligence and robotics.

This thesis presents the FSR-BAY model for robotic sensory fusion. This model takes into consideration certain aspects that in our opinion have been treated in a secondary manner by the majority of today's fusion architects: Incomplete and uncertain information learning capacities and the use of a homogenous representation of the information, independent of the level of fusion. Two studies of the proposed model applied to an autonomous agent are also described. The first case deals with cooperative fusion of the information using various sensors of the same type to provide the information, and the second case describes the situation of competitive fusion when the information is provided both homogeneously and heterogeneously.





Universitat d'Alacant
Universidad de Alicante

Índice general

1. Introducción	25
1.1. Fusión sensorial	26
1.1.1. El mundo real	27
1.1.2. ¿Qué aporta la fusión?	28
1.1.3. Límites de la fusión sensorial	31
1.2. Motivación y objetivos	31
1.3. Organización de la tesis	33
1.4. Resumen	34
2. Fusión Sensorial: Tipología y Arquitecturas	37
2.1. Tipos de Fusión Sensorial	37
2.1.1. Categorización a tres niveles	38
2.1.2. Clasificación basada en la entrada/salida	38
2.1.3. Clasificación basada en la configuración sensorial	39
2.2. Arquitecturas para la Fusión Sensorial	40
2.2.1. Arquitecturas Centradas en la Información	40
2.2.2. Arquitecturas Funcionales	44
2.2.3. Arquitecturas Basadas en el Comportamiento	49
2.2.4. Arquitecturas Híbridas	53
2.2.5. Arquitecturas Híbridas Organizativas	53
2.2.6. Arquitecturas Híbridas con Jerarquías de Estados	58
2.2.7. Arquitecturas Híbridas Orientadas a Modelo	59
2.3. Resumen	63
3. Programación e inferencia Bayesiana	67
3.1. Introducción a la PB	68
3.1.1. Conceptos Básicos	70
3.1.2. Definición de un programa Bayesiano	72



3.1.3.	Ejecución de un programa Bayesiano	74
3.2.	Inferencia Bayesiana	75
3.2.1.	Simplificación	76
3.2.2.	Búsqueda de las modas de una distribución	78
3.2.3.	Marginalización de variables	82
3.3.	Especificación de modelos probabilísticos utilizando PB	85
3.3.1.	Redes Bayesianas	85
3.3.2.	Estimación Bayesiana recursiva	87
3.3.3.	Mixture models	90
3.3.4.	Aproximaciones de máxima entropía	92
3.4.	Mapas Bayesianos	93
3.4.1.	Definición	93
3.4.2.	Localización, predicción y control	94
3.4.3.	Utilización de un mapa Bayesiano	95
3.4.4.	Grafo inducido por un mapa Bayesiano	98
3.4.5.	Operadores de combinación	99
3.5.	Resumen	99
4.	FSR-BAY: Modelo Probabilístico para la Fusión Sensorial Robótica	101
4.1.	Introducción	101
4.1.1.	Líneas Generales	104
4.2.	Modelo Propuesto	106
4.2.1.	Unidades de Procesamiento	107
4.2.2.	Conexión Bayesiana de Unidades de Procesamiento	109
4.3.	Operador de fusión complementaria	112
4.4.	Operador de fusión competitiva heterogénea	121
4.5.	Operador de fusión competitiva homogénea	127
4.6.	Sobre la independencia condicional	129
4.7.	Resumen	129
5.	Caso de estudio: Fusión Bayesiana Complementaria	131
5.1.	Introducción	132
5.2.	Plataforma robótica utilizada	134
5.3.	Unidad de localización	136
5.3.1.	Mapa Bayesiano del Pasillo, c_p	138
5.3.2.	Mapa Bayesiano de la Puerta, c_u	146
5.3.3.	Mapa Bayesiano de fin de pasillo, c_e	149



5.3.4.	Mapa Bayesiano de la pared, c_a	152
5.3.5.	Fusión de Mapas Bayesianos	154
5.3.6.	Desarrollo de una conducta simple	167
5.4.	Conducta con una unidad Bayesiana complementaria	175
5.4.1.	Arquitectura del sistema	176
5.4.2.	Unidad de ejecución	177
5.4.3.	Experimentación	178
5.5.	Detección de incoherencias	180
5.5.1.	Validación Experimental	181
5.6.	Comparación del sistema presentado con un sistema de control difuso	185
5.6.1.	Sistema de control difuso	185
5.7.	Resumen	191
6.	Caso de estudio: Fusión Bayesiana Competitiva	195
6.1.	Aprendiendo a ver con unidades Bayesianas	196
6.1.1.	Descripción del entorno utilizado	196
6.1.2.	Unidad de rango para evitar obstáculos (u_1)	199
6.1.3.	Unidad visual para evitar obstáculos (u_2)	201
6.1.4.	Fusionando unidades Bayesianas	203
6.1.5.	Obtención de la unidad u_2	204
6.1.6.	Experimentación	205
6.1.7.	Discusión	213
6.2.	Generación de Mapas 3D Mediante Fusión Bayesiana de Láser y Visión Omnidireccional	214
6.2.1.	Operador de fusión competitivo	216
6.2.2.	Unidades Bayesianas propuestas	217
6.2.3.	Fusión de unidades de proceso del color	223
6.2.4.	Resultados y experimentación	225
6.2.5.	Comparación entre la reconstrucción con y sin fusión	230
6.2.6.	Discusión	233
6.3.	Resumen	234
7.	Conclusiones y líneas futuras	237





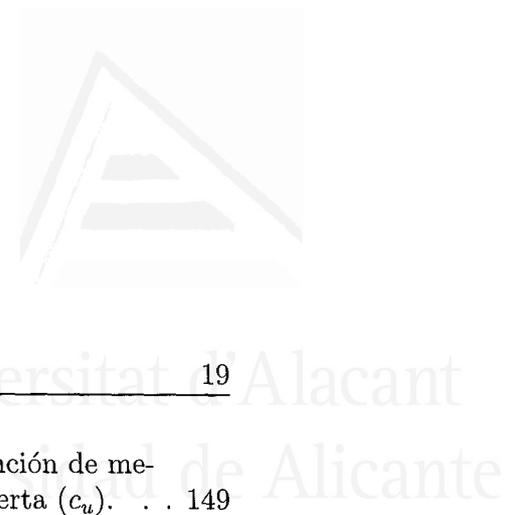
Universitat d'Alacant
Universidad de Alicante

Índice de figuras

1.1. Anillo de sónares del robot Pioneer I	29
2.1. El modelo JDL	41
2.2. Arquitectura de fusión en cascada	44
2.3. El modelo Boyd	45
2.4. Posibles relaciones entre las primitivas <i>Sentir, Planificar y Actuar</i>	47
2.5. El modelo Ómnibus	48
2.6. Comparativa entre la arquitectura de categorización y la arquitectura <i>Sentir-Planificar-Actuar</i>	50
2.7. La arquitectura LAAS	55
2.8. La arquitectura AuRA	57
2.9. La arquitectura SFX	58
2.10. La arquitectura 3T	60
2.11. La arquitectura Saphira	62
3.1. Estructura de un programa Bayesiano	72
3.2. Algoritmo para la resolución de las preguntas de predicción, localización y control	96
3.3. Segundo algoritmo para la resolución de las preguntas de predicción, localización y control	97
4.1. R : posibles acciones a realizar.	110
4.2. Conexión secuencial de UP	111
4.3. Operador de fusión complementaria	113
4.4. Operador de fusión complementaria	122
5.1. Dimensiones y forma del robot Pioneer I	134
5.2. Anillo de sónares del robot Pioneer I	135



- 5.3. Patrón del haz de los sensores de ultrasonidos MINI-S PB. 136
- 5.4. Entorno donde se realizarán las pruebas. Representa una parte del pasillo del Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante. 137
- 5.5. Mapas Bayesianos propuestos. a) Mapa del pasillo (c_p), d_x define la distancia del robot al centro del pasillo, θ representa el ángulo entre el robot y el pasillo. b) Mapa de la puerta (c_u), d_x y d_y definen la distancia del robot al centro del pasillo y al inicio de la puerta respectivamente, θ representa el ángulo entre el robot y el centro del pasillo. c) mapa de fin de pasillo (c_e), d_x define la distancia del robot al centro del pasillo, d_y la distancia del robot a la pared final y θ el ángulo del robot respecto a la esquina. d) Mapa de la pared (c_a), d'_y representa la distancia del robot a la pared detectada. Por otra parte θ representa el ángulo del robot respecto a la pared. Se considera positivo un ángulo si se especifica en sentido antihorario. 139
- 5.6. Formalización del mapa Bayesiano del pasillo utilizando el formalismo de la programación Bayesiana. 142
- 5.7. Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa pasillo (c_p). 142
- 5.8. Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (-500, 0, 25)$. b) Medidas en la posición $(x, y, \theta) = (-170, 0, 0)$. La distancia al obstáculo retornada por R_x se especifica en milímetros. 143
- 5.9. Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (170, 0, -25)$. b) Medidas en la posición $(x, y, \theta) = (500, 0, 25)$. La distancia al obstáculo retornada por R_x se especifica en milímetros. 144
- 5.10. Distribución $P(R_x | \theta \otimes d \otimes c_p)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-25, 0, 25\}$ para las posiciones $d_x = \{-500, -170, 170, 500\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros. 145
- 5.11. Formalización del mapa Bayesiano de la puerta utilizando el formalismo de la programación Bayesiana. 148



5.12. Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa puerta (c_u).	149
5.13. Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (-280, 270, 5)$. b) Medidas en la posición $(x, y, \theta) = (280, 530, 5)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.	150
5.14. Distribución $P(R_x \theta \otimes d_x \otimes d_y \otimes c_u)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-5, 5\}$ para las posiciones $d_x = \{-280, 280\}$ y $d_y = \{270, 530\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros.	151
5.15. Formalización del mapa Bayesiano de final de pasillo utilizando el formalismo de la programación Bayesiana.	153
5.16. Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa puerta (c_u).	154
5.17. Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (-600, 2625, 45)$. b) Medidas en la posición $(x, y, \theta) = (-200, 1940, -45)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.	155
5.18. Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (200, 2625, 0)$. b) Medidas en la posición $(x, y, \theta) = (600, 1050, 45)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.	156
5.19. Distribución $P(R_x \theta \otimes d_x \otimes d_y \otimes c_e)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-45, 0, 45\}$ para las posiciones $d_x = \{-600, -200, 200, 600\}$ y $d_y = \{1050, 1940, 2625\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros.	157
5.20. Formalización del mapa Bayesiano de la puerta utilizando el formalismo de la programación Bayesiana.	158
5.21. Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa pared (c_a).	158



- 5.22. Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (0, 1275, 20)$. b) Medidas en la posición $(x, y, \theta) = (0, 425, 0)$. La distancia al obstáculo retornada por R_x se especifica en milímetros. 159
- 5.23. Distribución $P(R_x | \theta \otimes d_x \otimes d_y \otimes c_a)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-20, 0, 20\}$ para las posiciones $d_y = \{425, 850, 1275\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros. 160
- 5.24. Distancia entre dos distribuciones Gaussianas sin tener en cuenta su desviación típica. En la segunda gráfica se solapan ambas distribuciones pero no se tiene en cuenta en la medida de distancia. 164
- 5.25. Probabilidad de que una distribución sea igual a otra. a) Prácticamente no existe probabilidad de que un valor generado por la primera distribución también sea generado por la segunda. b) La probabilidad de que un valor se genere por ambas distribuciones se representa en forma de área sombreada 165
- 5.26. Localización del mapa global para 60 medidas (indicadas por el centro de los rectángulos de la figura). La saturación del color y el tamaño de los rectángulos indican la probabilidad de la localización (mayor tamaño mayor probabilidad de localización). El resultado de la localización es pasillo (rojo), pared (verde), puerta (azul), fin de pasillo (magenta). Las localizaciones equiprobables se envuelven con áreas del mismo color. El robot siempre tiene una posición de 0° respecto al mundo (representada en la figura). 168
- 5.27. Localización del mapa global para 60 medidas (indicadas por el centro de los rectángulos de la figura). La saturación del color y el tamaño de los rectángulos indican la probabilidad de la localización (mayor tamaño mayor probabilidad de localización). El resultado de la localización es pasillo (rojo), pared (verde), puerta (azul), fin de pasillo (magenta). Las localizaciones equiprobables se envuelven con áreas del mismo color. El robot siempre tiene una posición de -90° respecto al mundo (representada en la figura). 169



 ÍNDICE DE FIGURAS

- 5.28. Localización del mapa global para 60 medidas (indicadas por el centro de los rectángulos de la figura). La saturación del color y el tamaño de los rectángulos indican la probabilidad de la localización (mayor tamaño mayor probabilidad de localización). El resultado de la localización es pasillo (rojo), pared (verde), puerta (azul), fin de pasillo (magenta). Las localizaciones equiprobables se envuelven con áreas del mismo color. El robot siempre tiene una posición de 90° respecto al mundo (representada en la figura). 170
- 5.29. Tabla que especifica la distribución $P(Vrot|L \otimes [L' = \langle 0, -170 \rangle])$, (para mejorar la legibilidad se representa únicamente el valor más probable para la variable $Vrot$). 171
- 5.30. Tabla que especifica la distribución $P(A|L_t \otimes L'_t)$ 173
- 5.31. Movimiento del robot en el entorno utilizando el mapa fusionado junto con una conducta simple de desplazamiento por el pasillo. Se muestran tres rutas generadas con distintas posiciones iniciales del robot. 174
- 5.32. Grafo inducido a partir de mapa Bayesiano de localización. . . 175
- 5.33. Arquitectura del sistema 176
- 5.34. Movimiento del robot utilizando la unidad de ejecución (azul discontinuo) y movimiento sin utilizar dicha unidad (rojo). El robot sin la unidad de ejecución cree detectar una puerta continuando recto y colisionando por tanto con la pared derecha. 179
- 5.35. Posiciones seleccionadas para realizar la validación experimental. Se ha intentado escoger posiciones que provoquen la elección de cada uno de los mapas descritos anteriormente. En cada posición se obtuvieron 30 lecturas a partir de las cuales se calculó la fiabilidad del sistema. 182
- 5.36. $P(L|R_x)$ para cada punto clave (representado en la figura 5.35) y cada unidad i (una unidad viene representada por su símbolo: pasillo, puerta, pared y fin de pasillo). 183
- 5.37. Para cada s3onar se calcula la fiabilidad $P(Rx_1^t | Rx_2^t \otimes Rx_3^t \otimes \dots \otimes Rx_7^t)$ seg3un cada subunidad y cada punto clave. Se muestra tambi3en la fiabilidad de la unidad global (F) en la 3ltima fila de la tabla. Para realizar las pruebas se contaminaron las lecturas del primer sensor aleatoriamente (fila R). La barra roja (la m3s clara de las dos) especifica el valor aleatorio utilizado y la barra azul el valor real que corresponder3a a la medida de dicho sensor. 184



5.38. Estructura del sistema de control difuso presentado.	186
5.39. Funciones de pertenencia para cada una de las variables de entrada. Los valores lingüísticos de esta variable son (de izquierda a derecha): <i>muy cerca, cerca, medio o lejos</i>	187
5.40. Funciones de pertenencia para la variable V_{rot} . Los valores lingüísticos de esta variable son (de izquierda a derecha): <i>Giro intenso a la izquierda, Giro a la izquierda, Pequeño giro a la izquierda, Centro, Pequeño giro a la derecha, Giro a la derecha, Giro intenso a la derecha, Dar la vuelta</i>	188
5.41. Funciones de pertenencia para la variable V_{tras} . Los valores lingüísticos de esta variable son (de izquierda a derecha): <i>velocidad baja, velocidad media, velocidad alta</i>	188
5.42. Reglas del sistema de control difuso.	189
5.43. Ruta desarrollada por el agente autónomo utilizando el sistema de control difuso propuesto.	190
5.44. Dos rutas desarrolladas por el robot utilizando el sistema de control difuso. Se puede observar como el robot desarrolla comportamientos sin sentido (giros en medio del pasillo, una colisión frente a una puerta...) debido a los errores provenientes de las lecturas de los sensores laterales.	191
5.45. Ruta desarrollada por el robot utilizando el sistema Bayesiano. Se muestran para 4 hitos distintos las lecturas del sónar de ultrasonidos, las lecturas contaminadas con ruido, la fiabilidad de un sensor en un mapa determinado, la fiabilidad global y la fiabilidad acumulada. Se puede observar como conforme pasan las iteraciones la fiabilidad de los sensores 1 y 7 decrece hasta que en la última iteración estos sensores no se tienen en cuenta para el funcionamiento global del sistema.	192
6.1. a) Forma y tamaño del robot Pioneer 3DX. b) Disposición del anillo de sónares frontal, el trasero es simétrico.	198
6.2. Ruta recorrida por el robot utilizando la unidad de evitación de obstáculos por ultrasonidos.	201
6.3. a) Ejemplo de imagen obtenida mediante la cámara virtual. b) Discretización de esta imagen en las 5 zonas utilizadas por la unidad de proceso.	202



ÍNDICE DE FIGURAS

6.4. Imagen virtual detectada por el robot en la posición 2 del mapa de la figura 6.6 junto con su discretización espacial (para mayor claridad no se muestra la discretización de intensidad). Además se adjunta la lectura del anillo completo de sónares de ultrasonidos (para la unidad solo se utilizarán del 2 al 6).	206
6.5. Imagen virtual detectada por el robot en la posición 3 del mapa de la figura 6.6 junto con su discretización espacial (para mayor claridad no se muestra la discretización de intensidad). Además se adjunta la lectura del anillo completo de sónares de ultrasonidos (para la unidad solo se utilizarán del 2 al 6).	207
6.6. Ejemplo de ruta recorrida por el robot utilizando la unidad de evitación de obstáculos por ultrasonidos. Además se muestran tres hitos donde para cada uno de ellos se especifica la lectura de la cámara, su discretización y las lecturas de los sónares de ultrasonidos.	208
6.7. Rutas desarrolladas utilizando únicamente la unidad de evitación de obstáculos mediante cámara.	209
6.8. Recorrido de entrenamiento del robot Pioneer 3DX en la fase de entrenamiento de la unidad u_2 . Se muestran imágenes reales del entorno obtenidas en 8 hitos distintos.	210
6.9. Lecturas de los sensores de ultrasonidos y discretización de las imágenes obtenidas para cada hito de la figura 6.8	211
6.10. Trayectoria realizada por el robot utilizando únicamente la unidad visual u_2	212
6.11. Robot Pioneer P3-AT con los sistemas de láser y omnivisión.	215
6.12. Mapa de puntos obtenido mediante la primera unidad Bayesiana, representando únicamente $\mu(P_W)$. La barra indica la profundidad del plano. Se observa un pasillo del departamento DCCIA. Partiendo del origen y moviéndose en el sentido del pasillo se puede ver una sala de 2 hasta 6 metros y dos puertas de despachos con su correspondiente ventanal superior a partir de los 14m. Se observan también dos bóvedas que proporcionan luz cenital al pasillo.	219
6.13. Plano del pasillo escogido para la realización de las pruebas. Se muestra también el espacio utilizado. La zona A de la figura 6.12 se sitúa en la A enmarcada en plano real del pasillo.	220
6.14. Forma, ecuación y medidas del espejo H3G de omnivisión proporcionado por Neovision.	222



- 6.15. Plano tangente a la hipérbola. Los puntos superiores del espejo son los que sufren más distorsión, no obstante son los menos importantes ya que a priori el techo no se utilizará para tareas de navegación. Los puntos cercanos a las paredes y el suelo presentan un error prácticamente inapreciable. 222
- 6.16. Ejemplo de una sección del entorno tomada por el láser junto con el espejo de omnivisión y el píxel que le corresponde (en amarillo). 224
- 6.17. Dado un determinado punto (x,y,z) es posible obtener más de una proyección para determinar su color. En esta figura se muestra un ejemplo de proyección utilizando dos imágenes obtenidas en dos posiciones distintas para un mismo punto. . . 226
- 6.18. Mapa de puntos con la información de color P_C obtenida por una unidad Bayesiana de proceso del color para todos los planos del mapa. Se representa una sección del pasillo y únicamente el color más probable de la distribución. 227
- 6.19. Reconstrucción de color P_C para cada plano obtenido fusionando la información de dos unidades. Se representa una sección del pasillo y únicamente el color más probable de la distribución. 228
- 6.20. Reconstrucción de color P_C para cada plano obtenido fusionando la información de dos unidades. Se representa la vista exterior del pasillo completo y únicamente el color más probable de la distribución. 229
- 6.21. Proyección de un punto en distintas imágenes obtenidas por el espejo de omnivisión. Es necesario fusionar esta información para mejorar el mapa tridimensional obtenido. 230
- 6.22. a) Imagen omnidireccional obtenida en un barrido del láser. b) Vista original del pasillo. c) Detalle de una reconstrucción utilizando únicamente una unidad. d) Reconstrucción conseguida fusionando dos subunidades. La flecha en la figura 6.19 indica el plano en el cual se han obtenido las tomas. 231
- 6.23. a) Imagen real del entorno. b,c) Reconstrucciones obtenidas sin y con fusión de unidades de color. 233



Universitat d'Alacant
Universidad de Alicante

- “¿Me podrías decir qué camino debo tomar desde aquí?
- Eso depende de adónde quieras ir - dijo el Gato
- No me importa mucho dónde... - dijo Alicia
- Entonces tampoco importa qué camino tomes - dijo el Gato
- ... con tal de que llegue a algún sitio - añadió Alicia como explicación.
- Oh, llegar a algún sitio seguro que llegarás - dijo el Gato - es sólo cuestión de que camines suficiente rato.”

(ALICIA EN EL PAÍS DE LAS MARAVILLAS), LEWIS CARROLL

Capítulo 1

Introducción

Actualmente se ha dedicado especial atención a la fusión sensorial en muchas áreas. Las técnicas de fusión sensorial combinan datos de múltiples sensores e información relacionada para conseguir inferencias más específicas que las que se podrían obtener utilizando un único sensor independiente. En este capítulo se revisarán los problemas derivados de trabajar con el mundo real y cómo la fusión sensorial puede ayudar a aliviarlos. También se definirá el término fusión sensorial y se resaltarán sus aportaciones a los procesos de fusión así como sus limitaciones. Por último se establecerán los objetivos de esta tesis y se describirá su hilo argumental.

Cabe destacar que este trabajo se enmarca dentro de las tareas desarrolladas por el grupo de investigación I3A (Informática Industrial e Inteligencia Artificial) de la Universidad de Alicante y es el resultado de su experiencia dentro de los campos de robótica móvil, planificación y scheduling, visión artificial, sistemas gráficos y sistemas de visualización. Destacamos de manera especial los siguientes proyectos de investigación como base del trabajo aquí presentado: “Una plataforma distribuida basada en agentes para auto-aprendizaje,



control y teleoperación de robots móviles a través de Internet” (TIC2001-0245-C02-02) concedido por la Comisión Interministerial de Ciencia y Tecnología (CICYT) y “Percepción visual y fusión sensorial en robótica” (GV04B685) concedido por la Generalitat Valenciana.

1.1. Fusión sensorial

El concepto de fusión sensorial no es nuevo. Tanto los humanos como los animales han evolucionado y han desarrollado la capacidad de utilizar sus sentidos para sobrevivir. Por ejemplo, determinar la calidad de una sustancia puede requerir utilizar más un sentido como el gusto y olfato (la visión en este caso puede no ser suficiente). De manera similar, cuando la visión queda limitada, el sentido del oído puede alertar de peligros inminentes. Así pues la fusión sensorial se realiza de forma natural por animales y humanos para conseguir una mejor interacción con el entorno circundante.

La emergencia de nuevos sensores, técnicas de procesamiento avanzado, y hardware de proceso mejorado, han hecho viable la fusión de datos incluso en tareas de tiempo real. Actualmente los sistemas de fusión sensorial se han utilizado de manera extensiva para el seguimiento de objetos, identificación automática, razonamiento,... En definitiva se tratan problemas como la robótica, control automático de procesos de fabricación industrial, aplicaciones médicas, etc. Cada una de estas aplicaciones tiene retos particulares y utiliza sensores específicos.

Los sistemas sensoriales remotos se han desarrollado para identificar y localizar entidades y objetos. Dentro de estos sistemas se engloban procesos de monitorización de recursos agrarios, la localización de procesos naturales y la monitorización del tiempo y desastres naturales. Estos sistemas se basan principalmente en la adquisición de imágenes utilizando sensores multiespectro. Una de las técnicas más utilizadas para la fusión sensorial en imágenes se basan el uso de redes neuronales adaptativas.

Otra área de aplicación es la monitorización de sistemas mecánicos complejos. Con el uso de estos sistemas se espera reducir los costes de mantenimiento e incrementara tanto la seguridad como la fiabilidad.



Las técnicas de fusión también se utilizan en inteligencia artificial y robótica. Sus aplicaciones principales son, entre otras, el desarrollo de interfaces aumentadas, que ayuden a la toma de decisiones y la implantación de sistemas autónomos (como automóviles autoguiados o incluso sondas espaciales). En esta tesis aplicaremos el modelo propuesto a este último grupo de aplicaciones.

1.1.1. El mundo real

Sentir es la pieza fundamental para desenvolverse. Los sentidos nos proporcionan los datos necesarios para recrear internamente una imagen de nuestro mundo o para saber manejarse en él.

Los sensores también son clave en el desarrollo de un robot autónomo ya que, al igual que en el ser humano, aportan los datos necesarios para interactuar de manera eficiente con el mundo. Dada una serie de datos obtenidos por uno o más sensores un robot ha de desenvolverse en el mundo, pudiendo basarse en un modelo o incluso obtener el modelo del mundo en el que interactúa.

Desafortunadamente el mundo real, en general, es muy complejo [RN96]:

- **El mundo real es inaccesible.** Los sensores no son perfectos y siempre percibirán exclusivamente aquellos estímulos que estén cercanos al robot.
- **El mundo real no es determinista.** Siempre existe la posibilidad de que las ruedas del robot derrapen o se agote la batería. Es por tanto necesario que el robot sea capaz de desenvolverse en un entorno con incertidumbre.
- **El mundo real no es episódico.** Los efectos producidos por las acciones cambian con el paso del tiempo. Un robot por tanto ha de ser capaz de desarrollar acciones secuenciales y a la vez tareas de aprendizaje.
- **El mundo real es dinámico.** Un robot ha de saber cuando ha de tener un comportamiento deliberativo y cuando uno reactivo.
- **El mundo real es contínuo,** y como tal no se pueden enumerar en su totalidad el conjunto de acciones posibles.



Una aproximación para abordar los problemas mencionados anteriormente es la fusión sensorial. Podemos definir la fusión sensorial como un proceso donde dadas las diversas lecturas de uno o varios sensores hay que ser capaz de combinarlas para extraer más información que meramente los datos individuales que las forman.

1.1.2. ¿Qué aporta la fusión?

Los sistemas que emplean métodos de fusión sensorial esperan un número de beneficios respecto a los sistemas tradicionales. Un sensor físico generalmente padece los siguientes problemas:

- **Privación sensorial.** La rotura de un elemento sensorial causa la pérdida de percepción del objeto deseado.
- **Cobertura espacial limitada.** Normalmente un sensor individual sólo cubre una región limitada (por ejemplo un sónar de ultrasonidos sólo detecta objetos dentro de una determinada área de acción).
- **Cobertura temporal limitada.** Algunos sensores necesitan un tiempo determinado para realizar y transmitir sus medidas.
- **Imprecisión.** Las medidas de sensores individuales están limitadas a la precisión del elemento sensitivo utilizado.
- **Incertidumbre.** La incertidumbre depende del objeto a ser observado en contraste con la imprecisión, que depende del sensor utilizado. La incertidumbre se manifiesta cuando faltan características (por ejemplo en oclusiones), cuando el sensor no puede medir todos los atributos relevantes de la percepción o cuando la observación es ambigua [Mur96]. Un sistema con un único sensor es incapaz de reducir la incertidumbre en sus percepciones debido a su punto de vista limitado del objeto [FH95]

Por ejemplo imaginemos un robot autónomo dotado únicamente de un anillo formado por siete sonares de ultrasonidos, distribuidos de manera que cubran la parte frontal del robot (ver figura 1.1). Inicialmente observamos como la configuración del anillo de sonares nos proporciona una configuración del espacio limitada ya que por la parte trasera del robot no disponemos de

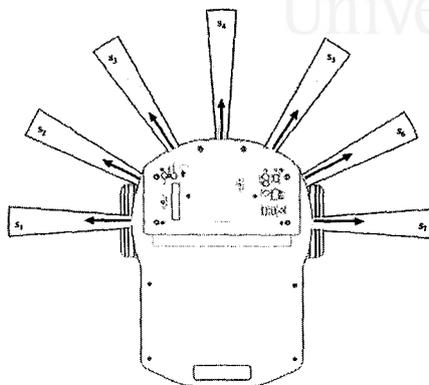


Figura 1.1: Anillo de sónares del robot Pioneer I

ningún artefacto sensorial que nos proporcione datos sobre el entorno en el que se desplaza el robot. Además los sonares de ultrasonidos se disparan con una determinada frecuencia. Si un objeto se desplazase más rápido que la frecuencia de disparo nos encontraríamos con un problema de cobertura temporal limitada.

Los investigadores en el campo de la robótica encuentran problemas al tratar con datos reales obtenidos mediante sonares. Dichos datos están contaminados con ruido, producido por los elementos sensitivos individuales (imprecisión) y producido por las reflexiones especulares del haz de sónar al incidir en un objeto (incertidumbre).

Fusionando los datos obtenidos por el anillo de sonares podemos reducir los problemas que padece un sensor físico (aumentar la cobertura espacial combinando diversas lecturas en diversas posiciones del robot, reducir la incertidumbre combinando la información de los distintos sonares del anillo, etc. . .).

Se pueden esperar las siguiente ventajas de la fusión de datos sensoriales de un conjunto de sensores, tanto heterogéneos como homogéneos:

- **Más robustez y confiabilidad.** Los sistemas multisensoriales tienen una redundancia intrínseca que capacita al sistema para proporcionar información incluso en el caso de algún fallo.
- **Se extiende la cobertura espacial y temporal.** Un sensor puede realizar observaciones en lugares donde otro sensor no puede o cuando otros no pueden.



- **Incremento de la confianza.** Una medida de un sensor se confirma mediante las medidas de otros sensores que cubran el mismo dominio.
- **Reducción de la ambigüedad y la incertidumbre.** La unión de la información reduce el conjunto de interpretaciones ambiguas de una determinada medida.
- **Robustez contra interferencias.** Incrementado la dimensionalidad del espacio de medidas (midiendo, por ejemplo, un objeto mediante sónares de ultrasonidos y sensores ópticos) el sistema se vuelve menos vulnerable ante interferencias.
- **Se mejora la resolución.** Cuando múltiples medidas independientes de la misma propiedad son fusionadas la resolución de los resultados es mejor que una única medida.

En [Rao98] se compara el resultado de fusionar un conjunto de medidas sensoriales utilizando un proceso apropiado con los datos de un sensor individual. Teniendo en cuenta este trabajo se puede desarrollar un proceso de fusión óptimo si la función de distribución de errores de un determinado sensor es conocida. Este proceso de fusión es como mínimo tan bueno como el mejor sensor individual.

Otra ventaja de la fusión sensorial es la posibilidad de reducir la complejidad del sistema. En un sistema de diseño tradicional las medidas sensoriales son embebidas, produciendo un gran número de medidas imprecisas o incorrectas. El uso de arquitecturas de fusión para trabajar con los datos sensoriales proporciona un método de abstracción que permite al programador trabajar de manera independientemente al tipo de sensor a utilizar. De esta manera se le facilita la implementación de aplicaciones complejas proporcionando un mecanismo sencillo para futuras modificaciones en el sistema sensorial[EHH01].

Por último cabe destacar que los métodos y las arquitecturas de fusión sensorial son de vital importancia para el desarrollo de aplicaciones robóticas, donde la cantidad y diversidad de los sensores utilizados requieren el uso de estas técnicas, siendo necesario analizar de manera conveniente qué arquitecturas y métodos utilizar para obtener los mejores resultados.



1.1.3. Límites de la fusión sensorial

Fowler en 1979 comentó que el proporcionar muchos datos de baja calidad y fusionarlos no produce datos de buena calidad sino que requiere una gran capacidad de procesamiento que puede llegar a reducir la calidad de la salida introduciendo retardos y niveles de confianza poco elevados [Fow79]. Multitud de investigadores ha intentado desmentir estos hechos. Se han presentado pruebas [NP80] de que añadiendo sensores se incrementa el rendimiento¹ del sistema en los casos específicos de votación mayoritaria y la teoría de máxima verosimilitud orientada a la fusión de decisiones.

También se han investigado los beneficios de incrementar el número de entradas en un proceso de fusión sensorial [Das97]. Aunque la investigación se limita a aumentar un sistema bisensorial con un tercer sensor, el trabajo muestra que ampliando el número de sensores se consigue un incremento o una pérdida de rendimiento dependiendo del algoritmo de fusión utilizado.

1.2. Motivación y objetivos

En las secciones anteriores hemos visto cómo la aparición de nuevos sensores, técnicas de procesamiento avanzado, y hardware de proceso mejorado, han hecho viable la fusión de datos, uno de los pilares fundamentales para el diseño de robots autónomos. Los sistemas que emplean métodos de fusión sensorial obtienen beneficios respecto a los sistemas tradicionales: más robustez y confiabilidad, se extiende la cobertura espacial y temporal, se incrementa la confianza, se reduce la ambigüedad y la incertidumbre, se aumenta la robustez contra interferencias y se mejora la resolución.

Al especificar un sistema de estas características el programador se obliga a determinar las propiedades de los datos o de la información a fusionar, los métodos de fusión, las arquitecturas... De esta manera, se establece un mejor diseño, implementación y análisis de los procesos siendo sencillo desarrollar métodos y algoritmos apropiados o incluso monitorizar la calidad de los procesos...

Una arquitectura de fusión orientada a la robótica es deseable que disponga de herramientas para tratar la incertidumbre y la no completitud de los

¹El rendimiento se define como la probabilidad de obtener la decisión correcta sin tener en cuenta los cálculos o las comunicaciones necesarias.



datos con los que trabaja. Como se ha visto en la sección 1.1.2 un sensor físico padece de una serie de problemas que pueden aliviarse utilizando mecanismos capaces de manejar información incompleta o incierta. De la misma manera, una vez analizadas las características del mundo real [RN96], es fácil darse cuenta de que al trabajar con este tipo de herramientas se consiguen sistemas más aptos para desenvolverse en el mundo real.

Cabe destacar que, aunque existen varias arquitecturas de fusión que sobresalen en el campo de la robótica estas tratan la incertidumbre, la no completitud de la información y las capacidades de aprendizaje de manera secundaria [ACF⁺98]. Además estas arquitecturas no consiguen una representatividad uniforme para la fusión de bajo y alto nivel.

Un modelo de fusión con estas características disfrutaría, entre otras, de las siguientes ventajas respecto a los sistemas de fusión clásicos: sería capaz de trabajar con la incertidumbre y con la no completitud de la información de manera explícita, proveería mecanismos de aprendizaje haciéndolo más versátil y adaptable y además posiblemente utilizaría una unidad de información básica, que se traduciría en una representatividad uniforme para la fusión de bajo y alto nivel.

El principal objetivo de esta tesis es avanzar en el campo de la fusión sensorial robótica proporcionando un modelo que cumpla las características anteriores. Para llevar a cabo este objetivo se han establecido distintos subobjetivos que se enumeran a continuación:

- **Estudio de las arquitecturas de fusión actuales.** Como paso previo al modelo propuesto, se realizará un análisis del estado del arte de las arquitecturas de fusión actuales. En este análisis se determinará el tratamiento de la incertidumbre en las arquitecturas genéricas orientadas a la fusión. Además se especificarán algunas de las arquitecturas para el control robótico actuales y se determinará como aplican los conceptos de fusión sensorial y como tratan de la incertidumbre.
- **Estudio de las aplicaciones de fusión Bayesiana más utilizadas en el campo de la robótica.** Metodológicamente se revisarán también algunos de los modelos Bayesianos para tratar con información incompleta e incierta que tengan especial importancia en el campo de la fusión sensorial. Se presentará el formalismo de la programación Bayesiana como marco conjunto de una gran parte de estos modelos y se verán algunas de sus aplicaciones dentro del campo de la fusión sensorial.



- **Proporcionar un modelo para la fusión sensorial que tenga como base la teoría Bayesiana de la probabilidad.** Una vez revisadas las arquitecturas de fusión actuales y vistas sus limitaciones se propondrá un modelo de fusión sensorial. Este modelo se centrará en la teoría Bayesiana de la probabilidad, disfrutando así de una base matemática robusta.
 - **Propuesta de un modelo con capacidad para tratar con información incompleta, incierta y con capacidades de aprendizaje.** El modelo presentado ha de tener en cuenta el tratamiento de la incertidumbre y la no completitud de la información.
 - **Proporcionar herramientas para la especificación formal de los mecanismos de fusión.** Se deberán proponer operadores de fusión que permitirán establecer los parámetros mediante los cuales se integrará nueva información.
 - **Permitir una correcta descomposición de la información utilizando una representación uniforme independiente del nivel de fusión a utilizar.** Una formalización correcta permite la descomposición de la información y por tanto se pueden dividir las tareas a realizar reduciendo de esta manera la complejidad del sistema. Además el modelo ha de proporcionar una representación uniforme de la información.
- **Estudio del modelo propuesto mediante diversas pruebas de concepto orientadas al campo de la robótica móvil.** Por último, como pruebas de concepto, se evaluará el modelo propuesto utilizando varios robots autónomos. Para desarrollar esta labor se utilizarán tanto simuladores (Saphira, Matlab) como robots reales (Pioneer I y Pioneer 3DX de la compañía ActivMedia²).

1.3. Organización de la tesis

A continuación se describirán brevemente los capítulos que componen esta tesis:

En el capítulo 2 se esbozará el estado del arte de las arquitecturas de fusión actuales. Además en este capítulo se verán arquitecturas específicas para

²<http://www.mobilerobots.com/>



la realización de sistemas robóticos y como algunas de estas arquitecturas utilizan procesos de fusión.

En el capítulo 3 se revisarán las bases del formalismo de la programación Bayesiana, que permite tratar información incompleta o incierta, necesario para especificar los modelos propuestos en esta tesis. Este formalismo además engloba varios modelos tal y como los filtros de Kalman, las redes Bayesianas...

Una vez analizado el estado del arte y el formalismo matemático a utilizar se propondrá en el capítulo 4 el modelo para la fusión sensorial, FSR-BAY, que contempla algunos aspectos desde nuestro punto de vista poco tratados en las arquitecturas anteriores. Este modelo se complementará con dos operadores que permitirán tratar diversas unidades de proceso especificando el tipo de fusión a utilizar.

En los capítulos 5 y 6 se aplicará el modelo propuesto a diversos ejemplos utilizando dos robots, Pioneer I y Pioneer 3DX. De esta manera se realizarán pruebas sistemáticas de los conceptos aquí presentados.

Por último en el capítulo 7 se desarrollarán las conclusiones del trabajo realizado y sus líneas futuras.

1.4. Resumen

Hoy en día la emergencia de nuevos sensores, las técnicas de procesamiento avanzado y hardware de proceso mejorado han hecho viable y necesario el uso de arquitecturas de fusión. Los sistemas que emplean métodos de fusión sensorial esperan un número de beneficios respecto a los sistemas tradicionales: más robustez y confiabilidad, extender la cobertura espacial y temporal, incrementar la confianza, etc. . . Actualmente los métodos y las arquitecturas de fusión sensorial son de vital importancia para el desarrollo de aplicaciones robóticas, donde la cantidad y diversidad de los sensores utilizados requieren el uso de estas técnicas. El principal objetivo de esta tesis es el avanzar en el campo de la fusión sensorial robótica proporcionando un modelo para la fusión sensorial que tendrá como base la teoría Bayesiana de la probabilidad para tratar con información incompleta e incierta. Este modelo contendrá una base matemática robusta que permitirá descomponer un problema en un conjunto de unidades de proceso. Se proporcionarán también varios operadores de fusión que permitirán establecer los parámetros mediante los cuales se integrará nueva información al sistema y varias pruebas experimentales que



CAPÍTULO 1. INTRODUCCIÓN

35

ilustran el funcionamiento y la versatilidad del modelo aquí presentado.





Universitat d'Alacant
Universidad de Alicante

“Las cosas pequeñas, si se ponen muy juntas, son más grandes que las grandes”

ANÓNIMO

Capítulo 2

Fusión Sensorial: Tipología y Arquitecturas

Los diseñadores de robots móviles suelen trabajar con un gran abanico de sensores de diversos tipos. Como se ha visto en el capítulo anterior la combinación de la información proveniente de estos sensores es fundamental para un funcionamiento robusto del sistema. En este capítulo se proporcionará una categorización de los diversos tipos de fusión sensorial existentes y un recorrido sobre el estado del arte de las arquitecturas y modelos para la fusión sensorial relevantes en la actualidad, detectando sus puntos débiles, teniendo en cuenta su ámbito de aplicación en robótica móvil.

2.1. Tipos de Fusión Sensorial

La mayoría de las clasificaciones de fusión sensorial dependen directamente de las arquitecturas utilizadas para el desarrollo del sistema. De esta manera existen clasificaciones que son difícilmente aplicables a otras arquitecturas distintas a la base sobre la cual se planteó la tipología.



A continuación se enumeran y describen brevemente las clasificaciones más utilizadas para las aplicaciones de fusión sensorial actuales.

2.1.1. Categorización a tres niveles

Los procesos de fusión se suelen clasificar en un modelo de tres niveles distinguiendo la fusión a bajo nivel, a nivel intermedio y a alto nivel.

- **Fusión a bajo nivel.** La fusión a bajo nivel o fusión de datos en bruto, combina diversas fuentes para producir nuevos datos que se esperan más informativos que las entradas.
- **Fusión de nivel intermedio.** La fusión de nivel intermedio o fusión de características combina varias fuentes, como vértices, esquinas, líneas, texturas o posiciones dentro de un mapa de características que se puede utilizar para la segmentación o la detección.
- **Fusión de alto nivel.** La fusión de alto nivel también conocida como fusión de decisión, combina las decisiones de diversos sistemas expertos. Los métodos de decisión para esta fusión incluyen la votación, la lógica difusa y los métodos estadísticos.

2.1.2. Clasificación basada en la entrada/salida

Utilizando el modelo de tres niveles anterior se ha propuesto una nueva clasificación más refinada [Das97]. La razón principal de este modelo es la existencia de diversos paradigmas donde la entrada y la salida del proceso de fusión pertenecen a distintos niveles. Por ejemplo en la extracción y selección de características la entrada son datos sin procesar y la salida pertenece a un nivel superior, el nivel de características. Los paradigmas de fusión, ambiguos algunas veces, han trabajado utilizando por un lado el nivel de entrada de los datos y por otro el nivel de salida. Para evitar estos problemas de clasificación se extendió la división de tres niveles a cinco categorías de fusión definidas por sus relaciones entrada/salida:

- Entrada de datos sin procesar, salida de datos sin procesar.
- Entrada de datos sin procesar, salida de características.



- Entrada en el nivel de características, salida en el nivel de características.
- Entrada en el nivel de características, salida en el nivel de decisión.
- Entrada en el nivel de decisión, salida en el nivel de decisión.

2.1.3. Clasificación basada en la configuración sensorial

Otro tipo de clasificación de los procesos de fusión sensorial es aquella que tiene en cuenta el tipo de configuración de los sensores a utilizar. Se distinguen principalmente tres tipos de configuración sensorial [DW88]:

Complementaria. Una configuración sensorial se llama complementaria si los sensores no dependen directamente entre ellos pero por otra parte se pueden combinar para conseguir una imagen más completa del fenómeno que está siendo observado. De esta manera se consigue reducir la no completitud de los datos sensoriales. Un ejemplo de configuración complementaria podría ser el empleo de múltiples cámaras para la observación de las partes de una habitación. Normalmente la fusión de datos complementarios es sencilla ya que los datos de los sensores independientes se pueden añadir entre sí [BS98].

Competitiva. Una configuración sensorial competitiva es aquella en la cual cada sensor delibera, de manera independiente, medidas de la misma propiedad. Se distinguen dos tipos posibles de configuración competitiva: la fusión de datos de diferentes sensores o la fusión de medidas de un único sensor tomadas a diferentes instantes de tiempo. Un caso especial de fusión competitiva es la tolerancia a fallos. Al igual que la tolerancia a fallos las configuraciones competitivas también pueden proporcionar robustez al sistema.

Cooperativa. Una red de sensores cooperativos utiliza información proporcionada por sensores independientes para obtener información que no estaría disponible con un único sensor. Un ejemplo de configuración de sensores cooperativos es la visión estereoscópica (combinando dos imágenes bidimensionales utilizando dos cámaras en diferentes puntos



de vista se obtiene una imagen tridimensional de la escena observada). La fusión cooperativa es la más difícil de diseñar ya que los datos resultantes son sensibles a inexactitudes provenientes de los sensores individuales que participan en la tarea [BS98]. Contrariamente a la fusión competitiva, la fusión cooperativa normalmente reduce la exactitud y la fiabilidad.

Estas tres categorías de configuraciones sensoriales no son mutuamente excluyentes. Muchas aplicaciones implementan aspectos de más de uno de estos tipos.

2.2. Arquitecturas para la Fusión Sensorial

La organización de un sistema robótico se define en [ACF⁺98] como el determinar sus capacidades para alcanzar sus metas y para reaccionar a diversos eventos. Dentro de la organización del sistema es fundamental el desarrollo de una buena estructura de control. La estructura de control de un robot autónomo debería contener comportamientos que en cierta manera deben estar relacionados con sus mecanismos sensoriales y con los mecanismos de fusión utilizados. El robot ha de ser capaz de proporcionar estos comportamientos para anticiparse a distintas situaciones efectuando las acciones que considere adecuadas. De esta manera podemos considerar que las arquitecturas de fusión sensorial son uno de los pilares fundamentales para el diseño de robots autónomos.

En esta sección realizaremos una pequeña revisión sobre algunas de las arquitecturas de fusión sensorial más utilizadas actualmente. Se verán algunas de las arquitecturas de fusión de propósito general y se determinarán algunas de las arquitecturas de control específicas del campo de la robótica resaltando los mecanismos de fusión utilizados.

2.2.1. Arquitecturas Centradas en la Información

Las arquitecturas de fusión centradas en la información se basan en una división del proceso de fusión de acuerdo con la abstracción de la información a ser fusionada. Por tanto, en este tipo de arquitecturas, no hay necesidad de establecer una secuencia de funciones de manera explícita [KMF00]. A continuación se describen dos de los modelos más referenciados para la fusión centrada en la información, el modelo JDL y el modelo de fusión en cascada.

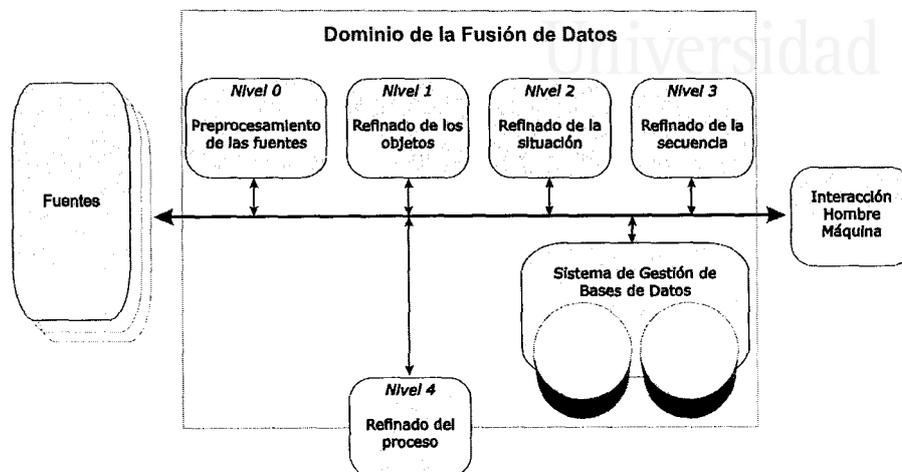


Figura 2.1: El modelo JDL

JDL

El modelo JDL [Wal90] fue desarrollado en 1985 por el organismo norteamericano *U.S. Joint Directors of Laboratories (JDL) Data Fusion Group* perteneciente al Departamento de Defensa de los Estados Unidos (DoD).

Se trata de un modelo de fusión funcional que contempla el uso de cinco niveles de procesamiento de datos junto con una base de datos, interconectados utilizando un bus. Estos niveles fueron pensados inicialmente para ser generales y útiles en varias áreas de aplicación.

Este modelo no especifica si los niveles se tienen que procesar en algún orden predeterminado y por tanto se pueden ejecutar de manera concurrente. No obstante los diseñadores de sistemas han asumido el orden de ejecución secuencial de manera constante en sus trabajos.

JDL es un modelo conceptual que identifica procesos, funciones y categorías técnicas, especificando los métodos aplicables a la fusión de datos.

A continuación se describen brevemente los niveles del modelo JDL y las partes que lo forman. Se toma como base el modelo propuesto por [Wal90] y revisado posteriormente por [SBW99] (JDL II).

- Inicialmente el modelo JDL se basa en la obtención de datos *fuentes*, que pueden ser proporcionados por una gran variedad de orígenes, como sensores, información a priori, bases de datos...



- El *preprocesamiento de las fuentes*, también conocido posteriormente como *nivel 0* [Bed99] es el encargado de reducir la carga de los datos a fusionar repartiendo dichos datos a los procesos adecuados.
- El *refinado de objetos* se trata de un proceso de fusión iterativo encargado del alineado de los datos (transformar los datos a un nivel referencia y a unas unidades consistentes) además de la asociación mediante métodos de correlación y el seguimiento de la posición actual junto a la posición futura de los objetos. Por último se contempla la identificación de los objetos utilizando métodos de clasificación. El producto de este nivel se suele llamar *mapa de situación*. De esta manera el *nivel 1*, intenta determinar de qué tratan los datos (identificación), dónde están y qué ocurrirá en un determinado momento (seguimiento).
- El *nivel 2* realiza el *refinado de la situación*. Este proceso iterativo de fusión trata de encontrar las relaciones espaciales y temporales entre entidades para agruparlas y formar interpretaciones abstractas o encontrar patrones entre las mismas.
- El *nivel 3* desarrolla el *refinado de la secuencia*¹. Basándose en el conocimiento a priori y las predicciones de la situación futura este nivel intenta esbozar inferencias sobre las vulnerabilidades y las oportunidades de operación.
- El *refinado de proceso* es el encargado de monitorizar el proceso de fusión y refinarlo para regular la adquisición de datos teniendo como finalidad llegar a unos resultados óptimos. El nivel 4 interactúa con cualquiera de los niveles anteriores.
- El sistema de gestión de bases de datos es el encargado de monitorizar, evaluar, añadir y actualizar la información del proceso de fusión.
- La interacción hombre/máquina es la parte encargada de proporcionar entradas y salidas adaptadas a usuarios humanos.

Partiendo de esta base se han propuesto algunas mejoras al modelo. Por ejemplo Morefield en el capítulo 21 de [Mv01] sugiere que la distinción de los niveles 2 y 3 es artificial y por tanto esos procesos se deberían unir. La

¹Thread refinement



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

43

última y más extensa revisión [LBR⁺04] propone diversas mejoras, como por ejemplo la necesidad de explotación de la fusión desde una perspectiva ontológica, ampliaciones del modelo JDL para tener en cuenta la fusión distribuida de datos (DDF) o extender el modelo para incluir control de calidad y consistencia en la fusión.

No obstante, aunque el modelo JDL ha sido muy popular para la fusión de información, sufre de los siguientes inconvenientes:

- Dado que sus orígenes fueron militares y aunque su ámbito de aplicación se ha expandido, por ejemplo a las aplicaciones comerciales, se encuentran problemas en su extrapolación a otros ámbitos. ¿Cómo se puede aplicar el modelo JDL fuera del dominio militar? Para poder utilizar dicho modelo tenemos que conseguir el *mapa de situación* equivalente al nivel 1, la medida de bondad del funcionamiento del sistema (para el nivel 4) y los otros niveles especificados previamente. El problema es que no siempre podemos disponer de todos los niveles descritos; por ejemplo no siempre podemos contar con la bondad del funcionamiento del sistema, los sensores pueden estar fuera de nuestro alcance y por tanto no se podría aplicar el *proceso de refinado* (nivel 4) [Wal01].
- El modelo JDL es un modelo centralizado tanto con los datos como con la información que maneja, lo cual lo convierte en difícil de extender y reutilizar [Elm02]. Cabe mencionar que en [LBR⁺04] se hacen algunas apreciaciones que pueden ayudar a suavizar este problema aunque no se da ninguna solución al mismo.
- El modelo es muy abstracto y hace difícil interpretar de manera correcta sus partes y aplicarlo adecuadamente a problemas específicos [Elm02].
- El modelo ayuda a una comprensión básica de la fusión pero no guía al desarrollador en la identificación de los métodos que tienen que ser utilizados [LH98].

El Modelo de Fusión en Cascada

El modelo de fusión en cascada, propuesto en 1997 [MHB⁺97] enfatiza la fusión en los niveles más bajos. Este modelo se basa en seis fases. *Sentir y procesado de la señal* corresponde con el nivel 0 de la arquitectura JDL. *Extracción de características y proceso de patrones* corresponde al nivel 1 del

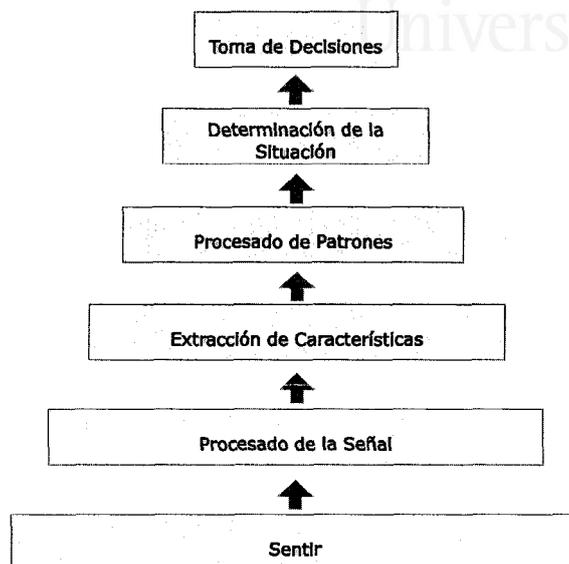


Figura 2.2: Arquitectura de fusión en cascada

JDL. *Determinación de la situación* corresponde con el nivel 2 del JDL. Por último, *la toma de decisiones* corresponde con el refinamiento de la situación del JDL (nivel 3).

Al ser similar esta arquitectura con el modelo JDL, sufre de las mismas desventajas. Además este modelo no considera ninguna retroalimentación de datos de manera explícita por lo que no representa la realidad de muchos sistemas.

2.2.2. Arquitecturas Funcionales

Mientras que las arquitecturas referenciadas anteriormente están centradas en el nivel de abstracción de la información, las arquitecturas funcionales están organizadas principalmente desde un punto de vista funcional. En este caso la secuencia de acciones a desarrollar es explícita [KMF00].

El Modelo Boyd

Dos años después de la propuesta inicial del modelo JDL un piloto norteamericano, Boyd, estaba intrigado por el sorprendente éxito del avión F-86.

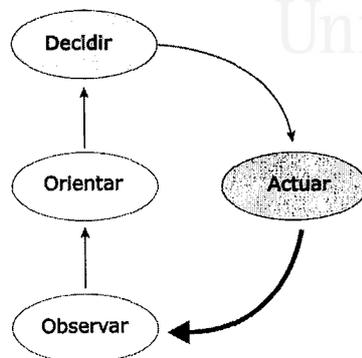


Figura 2.3: El modelo Boyd

Reflexionando llegó a la conclusión de que la ventaja del F-86 residía ampliamente en sus controles de vuelo que le permitían la transición de una maniobra aérea a otra más rápidamente que otros aviones. Los análisis posteriores revelaron implicaciones más amplias de esta teoría. Boyd expresó este concepto en un proceso cíclico al que llamó el circuito de **Observar-Orientar-Decidir-Actuar** (OODA). Tan pronto como un lado actuaba y observaba las consecuencias el circuito comenzaba nuevamente. La importancia de las teorías tácticas de Boyd se basan en la hipótesis de que este ciclo de operación continua se realizaba en niveles aun más elevados. Aunque propuestas por un aviador, estas teorías abarcaron mucho más que un anteproyecto para operaciones aéreas. Se consideró que la fusión, en general, se podía representar por este proceso.

El ciclo de control de Boyd (también conocido como OODA) representa el mecanismo clásico de apoyo a la decisión en las operaciones de información militares [Boy87]. Como los sistemas de apoyo a la decisión se integran con sistemas de fusión, el ciclo de Boyd se ha utilizado también para la fusión sensorial.

Este ciclo está dividido en cuatro fases:

- La fase de *observación* es comparable al nivel 0 de JDL y parte de la fase de *obtención del ciclo de inteligencia militar*².
- La fase de *orientación* corresponde a las funciones de los niveles 1, 2

²El ciclo de inteligencia son los pasos imaginarios por medio de los cuales se reúne la información, se convierte en inteligencia y se pone a la disposición de los usuarios.



y 3 del modelo JDL. Además incluye los elementos estructurales y las fases de recolección de la inteligencia militar.

- La fase de *decisión* es comparable al nivel 4 del modelo JDL e incluye además las actividades de diseminación del ciclo de inteligencia.
- La fase de *actuación* no tiene análogo en el modelo JDL siendo la parte del modelo que explícitamente cierra el ciclo teniendo en cuenta el efecto de las decisiones en el mundo real.

El modelo de Boyd representa las etapas de un ciclo de control cerrado y da una visión de la tarea del sistema. Las cuatro fases del modelo OODA se pueden reescribir como **sentir-interpretar-planificar-actuar**. Esta es la nomenclatura utilizada frecuentemente en el campo de la inteligencia artificial y en la robótica. En [Mur00] se describen una serie de arquitecturas en términos de la relación entre las primitivas **sentir**³, **planificar** y **actuar**. Dependiendo de la relación entre estas primitivas se establecen tres paradigmas diferenciados:

El paradigma jerárquico fue el predominante en los primeros años de la inteligencia artificial robótica, donde la mayoría del esfuerzo se dedicaba a la planificación. El énfasis en estos primeros sistemas estaba en la construcción de un mundo detallado. Una vez obtenido el mundo se planeaban cuidadosamente los pasos siguientes. El problema de este tipo de arquitecturas es que mientras el robot se dedica a construir su modelo y deliberar sobre qué hacer, el mundo puede haber cambiado.

Los dos últimos paradigmas, reactivo e híbrido, se verán en las secciones siguientes, ya que forman parte de otras arquitecturas pendientes de mencionar.

El modelo OODA así como el modelo SPA (**sentir, planificar y actuar**) sufren de la falta de una estructura apropiada para identificar y separar las diferentes tareas de fusión sensorial [Elm02]. Además, se basan en la obtención de datos por un robot para *completar* un modelo del mundo previamente adscrito lo cual presenta varios inconvenientes:

- Se supone que el robot contiene un modelo del mundo que es capaz de representarlo de manera fiable. El incluir un modelo previo del mundo

³Sentir estará formado por las dos primeras fases del ciclo Boyd, observar y orientar (sentir e interpretar).



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

47

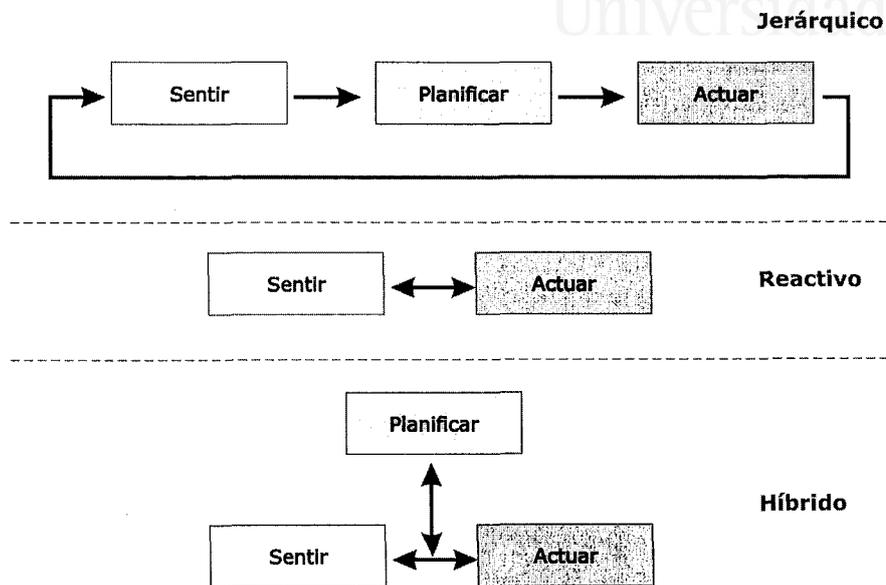


Figura 2.4: Posibles relaciones entre las primitivas *Sentir*, *Planificar* y *Actuar*

puede suponer el perder generalidad.

- Asumir que un modelo es capaz de representar correctamente al mundo real puede ser incorrecto: siempre existen variables ocultas que no se han tenido en cuenta en el modelo y que por tanto pueden provocar que en un determinado punto el mundo y el modelo dejen de actuar de igual manera.
- Las fases de **sentir** y **planificar** han de ser lo suficientemente rápidas como para no tener cambios en el mundo entre las dos fases. Este es el principal problema de estas arquitecturas, no son adaptables a mundos cambiantes. Los robots programados con este tipo de arquitecturas suelen exhibir comportamientos lentos e incorrectos para mundos dinámicos.

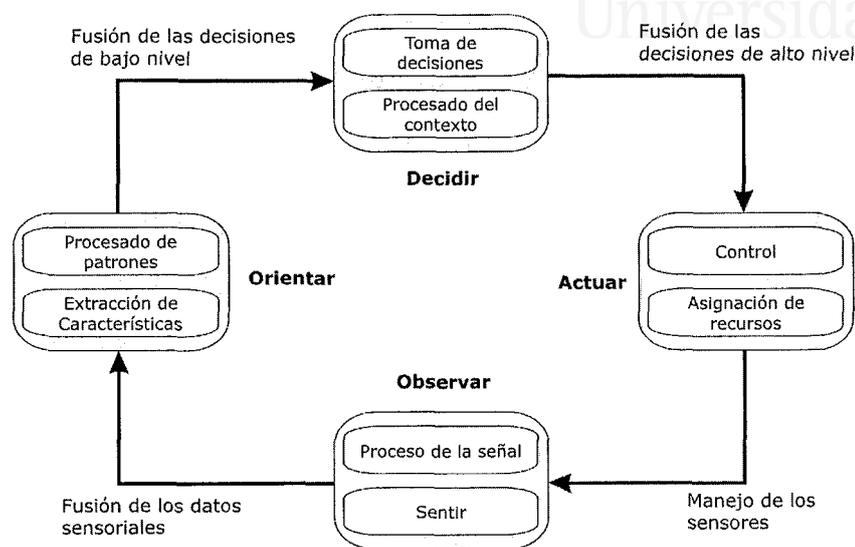


Figura 2.5: El modelo Ómnibus

El modelo Ómnibus

El modelo Ómnibus [Bed99] fue presentado en 1999 por Bedworth y O'Brien. Este modelo soluciona algunas de las limitaciones de las arquitecturas previas para la fusión y recalca sus ventajas. Contrariamente al modelo JDL el modelo Ómnibus define el orden de los procesos y hace la naturaleza cíclica explícita. Utiliza una terminología que no asume que las aplicaciones estén orientadas a defensa. El modelo muestra una estructura cíclica comparable al ciclo de Boyd pero proporcionando una estructura mucho más detallada en los niveles de procesamiento.

La finalidad del modelo es el utilizar muchas veces la misma aplicación de manera recursiva en dos niveles diferentes de abstracción. Como primer paso el modelo se utiliza para caracterizar la estructura conjunta del sistema. Después las mismas estructuras se utilizan para modelar las siguientes subtareas del sistema.

Aunque la separación jerárquica de las tareas de fusión sensorial es sofisticada, en el modelo Ómnibus no se soporta la partición horizontal de las mismas de manera que reflejen, por ejemplo, sensores distribuidos y proceso de datos. De esta manera este modelo no proporciona una descomposición en módulos implementados de manera separada para ser reutilizados en di-



ferentes aplicaciones [Elm02].

2.2.3. Arquitecturas Basadas en el Comportamiento

Arquitectura de Categorización

Rodney Brooks desarrolló la arquitectura de categorización⁴ a mitad de los años 80 en el MIT [Bro86] y la extendió en 1991 [Bro91] para proporcionar un mayor grado de control distribuido. Su aproximación de comportamientos puramente reactivos contrastó con los métodos tradicionales de inteligencia artificial desarrollados hasta el momento. Brooks dijo que el paradigma *Sentir-Planificar-Actuar* (SPA) utilizado en algunos de los primeros robots era contraproducente para la construcción de robots reales. De esta manera argumentó que “*construir modelos del mundo y razonar a partir de ellos con representaciones explícitas del conocimiento eran un impedimento para la respuesta temporal de los robos*”. Por consiguiente, según Brooks, la unión de la percepción y de la acción lleva a la inteligencia, la cognición sólo está en el ojo del observador.

Las bases o principios de las arquitecturas basadas en el comportamiento se pueden resumir mediante los siguientes preceptos:

- Los comportamientos complejos no tienen por qué ser producto de un sistema de control complejo
- La inteligencia está en el ojo del observador
- El mundo es nuestro mejor modelo
- La simplicidad es una virtud
- La robustez en presencia de ruido o fallos en los sistemas sensoriales ha de ser una meta
- Planificar es una manera de evitar concretar qué hacer en el siguiente paso.
- Los sistemas se tienen que construir de manera incremental
- No utilizar representaciones, calibraciones, ordenadores complejos ni comunicaciones que requieran un elevado ancho de banda

⁴Del término inglés Subsumption.

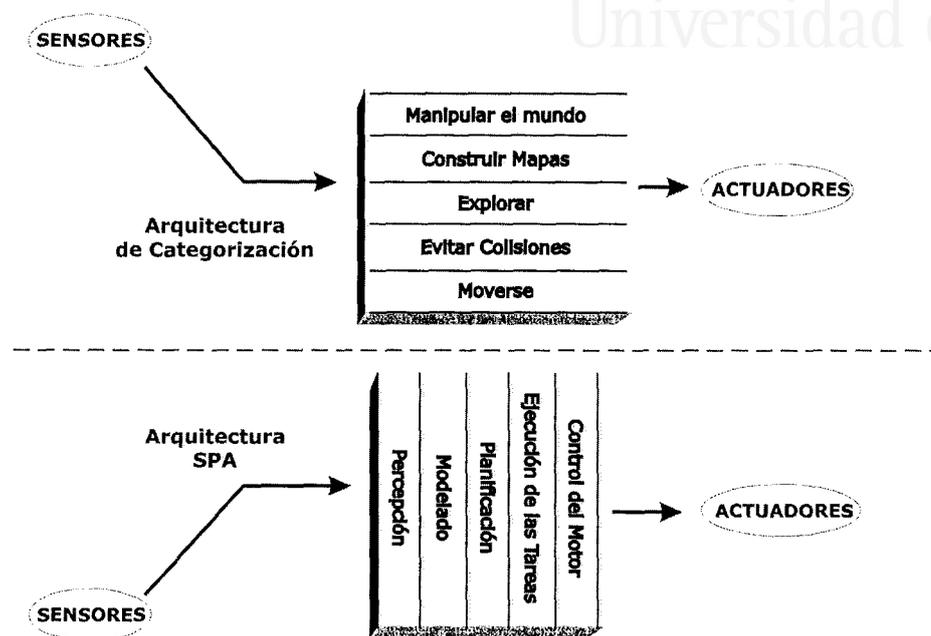


Figura 2.6: Comparativa entre la arquitectura de categorización y la arquitectura *Sentir-Planificar-Actuar*

Brooks propone un sistema de capas de control, incluido en la arquitectura, pero distribuido. Este sistema difiere de los modelos clásicos por la utilización de una serie de dimensiones que no corresponden con ninguna de las existentes previamente. La diferencia con el modelo **Sentir-Planificar-Actuar** se puede ver en la siguiente figura (ver 2.6).

El modelo SPA (en la parte inferior de la figura) se ejecuta secuencialmente. El modelo de la parte superior desarrolla la arquitectura de categorización, donde se realizan todas las fases en paralelo.

Las capas de la arquitectura de categorización están compuestas de redes de máquinas de estado finito con temporizadores (Augmented Finite State Machines – AFSM). Estos temporizadores habilitan cambios de estado después de periodos de tiempo preprogramados. Cada AFSM tiene una entrada y una salida. Cuando la entrada de la AFSM excede un determinado umbral el comportamiento de dicha AFSM se activa (su salida se activa). Las entradas de las AFSM provienen de sensores o de otras AFSM.

Cada AFSM acepta una señal de supresión y una señal de inhibición



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

51

como mecanismo de coordinación. La señal de supresión sobrescribe cualquier entrada normal con la especificada. La señal de inhibición causa que la salida quede totalmente eliminada. Estas señales permiten comportamientos que sobrescriben otros de manera que el sistema pueda producir comportamientos coherentes.

La arquitectura de categorización es paralela y asíncrona, ya que las AFSM operan de manera independiente entre ellas. Todas las AFSM están siempre funcionando y como tienen sus propios temporizadores no necesitan sincronización explícita con las demás.

El uso de AFSM resulta en una unión firme entre la percepción y la acción produciendo unas respuestas altamente reactivas características de los sistemas de categorización. No obstante los patrones de comportamiento en estos sistemas están preestablecidos. Las AFSM son las únicas unidades procesadas por esta arquitectura, es decir, no existen símbolos. Cada módulo desarrolla una tarea independiente como controlar un determinado sensor o actuador. Esto se puede comparar con arquitecturas más cognitivas donde se pueden agrupar los módulos sensoriales o los actuadores en un sólo módulo de la arquitectura. Las acciones simples, realizadas por los módulos, se pueden combinar en comportamientos más complejos utilizando una serie de capas jerárquicas.

Una característica de las arquitecturas de categorización es la falta de un modelo del mundo. Según Brooks al no tener una representación no se pierde el tiempo en procesarla, no obstante la arquitectura depende únicamente del mundo para saber qué hacer, es puramente reflexiva.

A modo de resumen observamos como la arquitectura de categorización proporciona principalmente las siguientes ventajas:

- *Adecuación al hardware.* Se pueden incluir de manera directa las AFSM en hardware.
- *Soporta paralelismo.* Cada capa puede ejecutarse independientemente.
- *Alta velocidad de respuesta.* Al no tener un modelo interno el tiempo de respuesta es muy bajo.

Por otra parte esta arquitectura también tiene otras tantas desventajas:

- *Muy baja flexibilidad en tiempo de ejecución.* Se trata de una arquitectura reflexiva, sin modelo del mundo donde las prioridades y las acciones están preestablecidas.



- *Dificultad para representar tareas complejas.* Las tareas complejas son muy difíciles de diseñar utilizando este tipo de arquitecturas reactivas.
- *Baja modularidad.* Las capas superiores interfieren con las inferiores y no son independientes.

Arquitectura Motora

Utilizando la técnica de los campos de potencial la arquitectura motora⁵ [Ark89] intenta crear un modelo más general, basado en comportamientos. Esta arquitectura consiste en una serie de *esquemas*, los cuales han de examinar una parte de los estímulos (datos) de entrada y elaborar un vector de salida basado en dichos estímulos que actuará sobre una parte de los efectores. Estos recibirán un vector de acción que corresponderá con una meta en particular de la arquitectura. Un robot que opere con una arquitectura motora podrá tener una serie de *esquemas*, cada uno de los cuales examinará una parte de los datos sensoriales dando un vector de acción. De esta manera se puede fusionar la información sensorial de una gran variedad de sensores (como visión, ultrasonidos, láser. . .) e integrarla en la decisión de las acciones motoras a tomar.

Los comportamientos del robot son obtenidos como la suma de todos los vectores de acción.

Si un único *esquema* no es suficiente para obtener un determinado comportamiento entonces se pueden agrupar un conjunto de *esquemas* para obtener un nuevo comportamiento.

La diferencia fundamental entre este tipo de arquitecturas y las de categorización viene dada porque con las arquitecturas motoras la salida es la mezcla de una serie de comportamientos formados por diferentes *esquemas*. Por el contrario en las arquitecturas de categorización el robot realiza únicamente una acción en un tiempo determinado.

Aunque este tipo de arquitecturas permiten la elaboración de sistemas más complejos que los basados en las arquitecturas de categorización sufren de la misma desventaja principal: las tareas complejas son muy difíciles de diseñar utilizando este tipo de arquitecturas reactivas.

⁵Del término inglés Motor Schema



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

53

2.2.4. Arquitecturas Híbridas

Las arquitecturas de fusión que se han revisado en las secciones anteriores plantean diversos inconvenientes. Las arquitecturas centradas en la información, aunque aportan una buena herramienta para el estudio y comprensión del problema a desarrollar, no guían de manera suficiente al diseñador para la implantación de su sistema. Los modelos funcionales pueden tener el problema de proporcionar sistemas demasiado lentos a los estímulos y necesitar descripciones complejas del mundo. Por otra parte los sistemas basados en el conocimiento tienen inconvenientes para resolver problemas complejos y suelen favorecer poco la modularidad y la reutilización de código. Por ello en el campo de la robótica surgen multitud de arquitecturas, basadas en ideas extraídas de los modelos de fusión anteriores, que intentan mejorar o tratar aspectos no contemplados previamente.

En esta sección describiremos algunas de las arquitecturas robóticas híbridas, haciendo especial hincapié en los mecanismos de fusión utilizados. Según Robin Murphy [Mur00] estas arquitecturas se pueden dividir en tres grandes grupos, organizativas, con jerarquía de estados y orientadas a modelo.

2.2.5. Arquitecturas Híbridas Organizativas

Las arquitecturas híbridas organizativas plantean la funcionalidad o la deliberación como un conocimiento global o como modelos del mundo que abarcan desde el tiempo pasado al tiempo futuro. La reactividad se establece como un comportamiento inicializado por una apreciación persistente en el tiempo. La diferencia fundamental con el resto de arquitecturas híbridas viene dada por la organización de la responsabilidad en su parte deliberativa. Esta organización se basa en manejadores que pueden representarse como agentes y que tienen determinadas atribuciones.

En este tipo de arquitecturas el comportamiento final del robot emerge por la interacción de la capa deliberativa con la capa reactiva del agente (donde se realizan mayormente los procesos de fusión sensorial).

LAAS

La arquitectura LAAS presentada en [ACF⁺98] se desarrolló como una arquitectura integrada para el diseño y la implementación de robots móviles con respecto al desarrollo de tareas en tiempo real y la reutilización de



código. Esta arquitectura incluye varias tareas, eventos y ciclos para alcanzar capacidades de anticipación y tareas en tiempo real. La arquitectura se descompone en tres niveles, con diferentes limitaciones temporales y manipulando diferentes representaciones de datos:

- Un nivel funcional, que incluye todas las acciones básicas y todas las capacidades de percepción. Estas funciones de procesamiento y bucles de control (procesado de la imagen, evitación de obstáculos, control del movimiento. . .) son encapsulados dentro de módulos intercomunicados. Para conseguir que este nivel sea independiente del hardware y por tanto portable se proporciona el interfaz entre sensores y efectores a través de un *nivel lógico del robot*.
- Un nivel de control de la ejecución o ejecutivo, que controla y coordina la ejecución de las funciones distribuidas en los módulos respecto a los requerimientos de las tareas.
- El nivel de decisión, incluye las capacidades de planificación de tareas y supervisión de la ejecución, siendo al mismo tiempo reactivo a los eventos del nivel anterior. Este nivel se puede descomponer en dos o más capas basándose en el mismo diseño conceptual pero utilizando diferentes representaciones o algoritmos con diferentes propiedades temporales.

La arquitectura LAAS agrupa la fusión de bajo y medio nivel como módulos en el nivel funcional. La fusión de alto nivel está representada en el nivel de decisión. Los requerimientos temporales son diferentes entre el nivel de decisión y el nivel funcional. Aunque esta arquitectura proporciona un buen mecanismo para la división en módulos de sistemas grandes, no proporciona un buen mecanismo para la comunicación en tiempo real y la representación de los datos fusionados en los niveles superiores al funcional [Elm02].

AuRA

La arquitectura AuRA (Autonomous Robot Architecture) fue propuesta en 1997 por Ronald Arkin [Ark97]. Es una arquitectura orientada a navegación autónoma compuesta por dos niveles diferenciados, uno reactivo y el otro jerárquico.



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

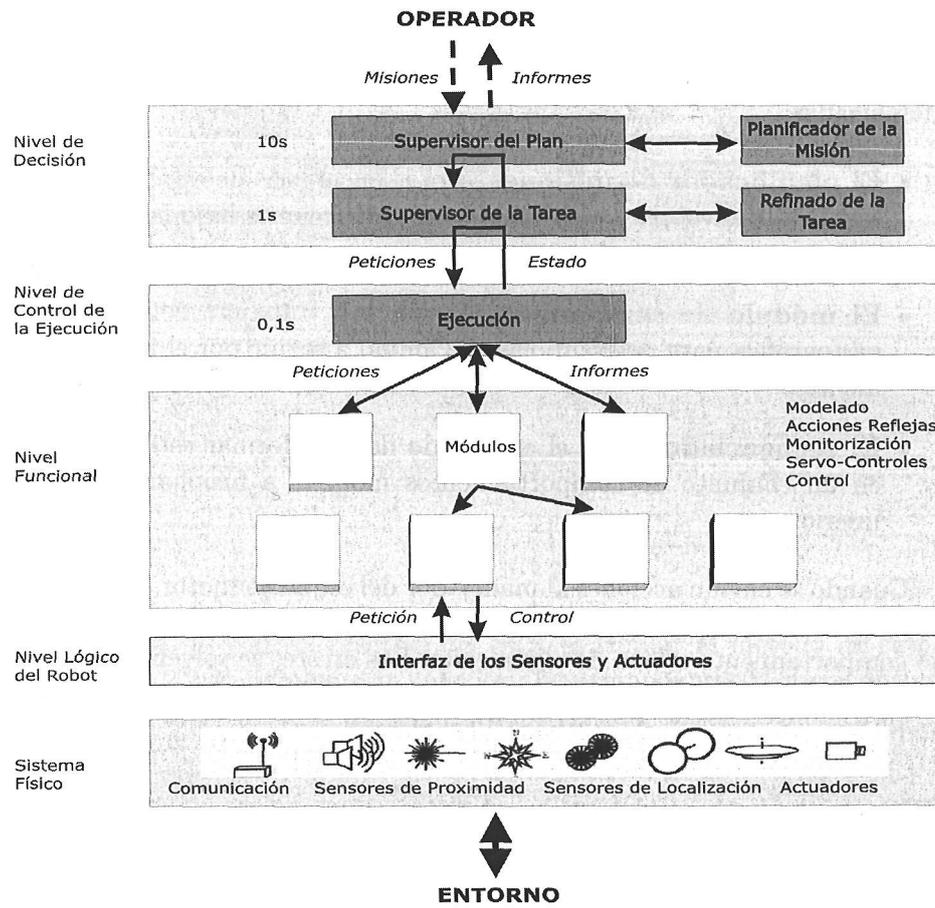


Figura 2.7: La arquitectura LAAS



El componente reactivo está formado por un esquema Motor (ver 2.2.3, arquitectura motora) que contempla un conjunto de comportamientos de bajo nivel. Cada comportamiento genera como salida un vector de respuestas que se combinan con el resto de los vectores de los comportamientos para obtener el comando final, que se transmitirá al robot físico. Los comportamientos se revisan por un manejador de esquemas que será el encargado de monitorizar y controlar su evolución.

La parte jerárquica de esta arquitectura está formada por tres partes diferenciadas:

- **El planificador de misiones**, será el encargado de establecer las metas de alto nivel para el sistema y las restricciones bajo las que se debe operar.
- **El módulo de razonamiento espacial**, trabajará con información cartográfica para descomponer el camino a seguir por el robot en fragmentos.
- **El secuenciador**, será el encargado de transformar estos fragmentos en un conjunto de comportamientos motores a fusionar por el nivel inferior.

Cuando se envían acciones al manejador del esquema motor, el nivel jerárquico queda a la espera de una notificación que confirme la correcta ejecución del comportamiento o bien de algún error. Los errores se solventan empezando por el secuenciador y terminando en el planificador (si todos los otros módulos son incapaces de resolver el problema).

SFX

Sensor Fusion Effects es una arquitectura basada en agentes propuesta por Murphy [Mur00] como una extensión a AuRA centrada en la fusión sensorial y en el manejo de errores. También se trata de una arquitectura a dos niveles, uno deliberativo y otro reactivo inspirados en modelos cognitivos.

La capa reactiva se subdivide en dos capas, una relativa a los comportamientos estratégicos y otra a los tácticos. De esta manera para situaciones transitorias se priman comportamientos tácticos capaces por ejemplo de evitar un determinado obstáculo. En ausencia de contingencias se priman los comportamientos estratégicos, encargados de desarrollar objetivos de un



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

57

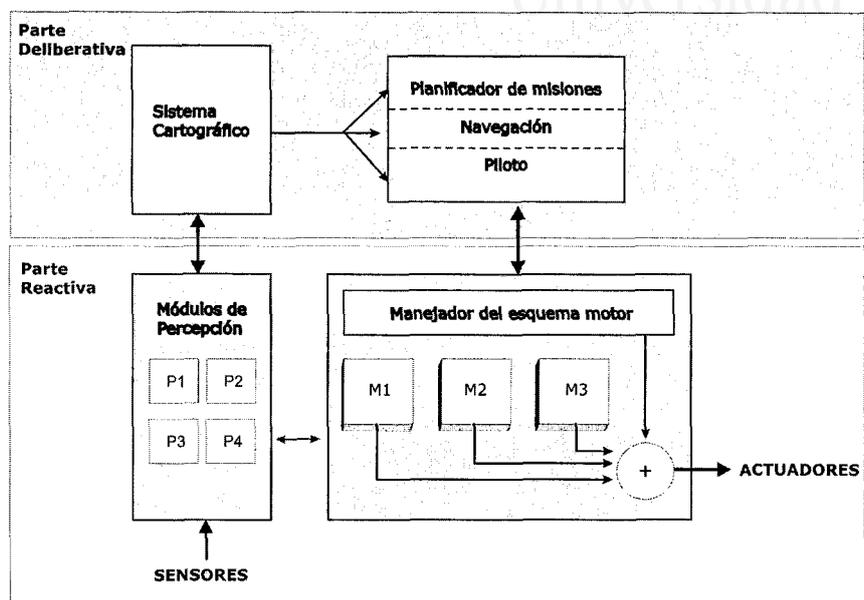


Figura 2.8: La arquitectura AuRA

nivel superior. El control reactivo del SFX es similar al método de categorización (ver 2.2.3) pero en este caso los comportamientos tácticos obtienen el control en determinadas situaciones sobrescribiendo los comandos de alto nivel.

La capa deliberativa se divide en módulos, implementados como agentes, que interactúan entre ellos. Existe un agente supervisor, llamado planificador de la misión, que establece el interfaz a alto nivel con el usuario y controla la evolución del sistema. A un nivel inferior existen tres sistemas encargados de atender a los sensores y efectores, *el manejador de tareas*, *el manejador de sensores* y *el manejador de efectores*. El manejador de sensores es capaz de obtener mediante fusión la fiabilidad de los sensores y solventar problemas con los mismos. Al mismo nivel también podemos encontrar manejadores deliberativos como el sistema cartográfico, encargado de construir un modelo del mundo y planificar las rutas a seguir por el robot.

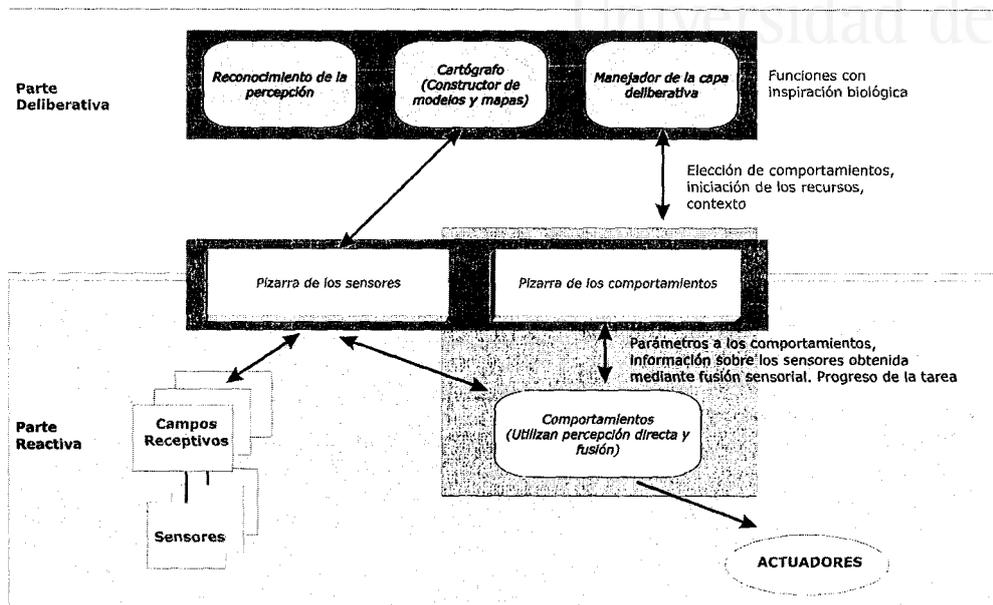


Figura 2.9: La arquitectura SFX

2.2.6. Arquitecturas Híbridas con Jerarquías de Estados

Las arquitecturas híbridas con jerarquías de estados plantean la funcionalidad o la deliberación como un conocimiento tanto pasado como futuro. La reactividad se establece como un comportamiento puramente reflexivo y sólo con conocimientos locales, por tanto requiere únicamente conocimiento del presente. El apartado deliberativo se diseña utilizando estados temporales que utilizarán la velocidad de ejecución como medida de prioridad. Para el *presente* se utiliza el controlador, para el *pasado* se utiliza el secuenciador y para el *futuro* el planificador.

En este tipo de arquitecturas el comportamiento final del robot emerge por la generación y la monitorización de un conjunto de comportamientos, por las habilidades o conjuntos de comportamientos y por los sistemas de categorización.



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

59

3T

La arquitectura 3T (*3 Tiers*) [BFG⁺97] pretende alcanzar un comportamiento robusto para la resolución de tareas utilizando tres niveles.

El primer nivel, el de capacidades o habilidades reactivas está formado por un conjunto de acciones de control dependientes del entorno que han sido obtenidas de manera sistemática (para evitar resultados dependientes de algún contexto de ampliación). De esta manera se llega a una representación uniforme para las capacidades que facilita su manipulación. Esta representación está formada por una especificación de las entradas y salidas, un algoritmo de procesamiento, una rutina de inicialización, una función de activación y por último otra de desactivación. Las capacidades se controlan por medio de un gestor que establece un interfaz uniforme con el siguiente nivel. Las capacidades siempre operan en el *presente*.

El segundo nivel, el secuenciador, es el encargado de transformar las tareas proporcionadas por el nivel superior en habilidades a desarrollar por el primer nivel. El secuenciador dispone de una librería de habilidades recomendadas para cada tarea a realizar. También es capaz de procesar un tipo especial de habilidades, los eventos, encargados de notificar circunstancias relevantes. El secuenciador trabaja sobre el *presente* y el *pasado*.

El tercero y último nivel, el planificador, es el encargado de proporcionar una perspectiva global al sistema. El planificador trabaja sobre el *pasado*, *presente* y *futuro*.

La arquitectura 3T pretende que los tres niveles definidos operen de manera concurrente y asíncrona. De esta manera, los algoritmos lentos deben ir en el planificador y los rápidos en el manejador de capacidades. No obstante, esta división puede implicar clasificaciones no del todo lógicas, por ejemplo los sistemas de visión, que corresponden a funciones sensoriales de bajo nivel, se deberían disponer en el planificador en lugar de en las capacidades.

2.2.7. Arquitecturas Híbridas Orientadas a Modelo

Las arquitecturas híbridas orientadas a modelo plantean la deliberación como todo aquello que esté relacionado con un comportamiento condicionado por una meta o un objetivo. La reactividad se establece como pequeñas unidades de control que operan en el presente pero pudiendo utilizar conocimiento global proporcionado por un sensor virtual. La responsabilidad en el apartado deliberativo se diseña utilizando componentes de comportamiento,

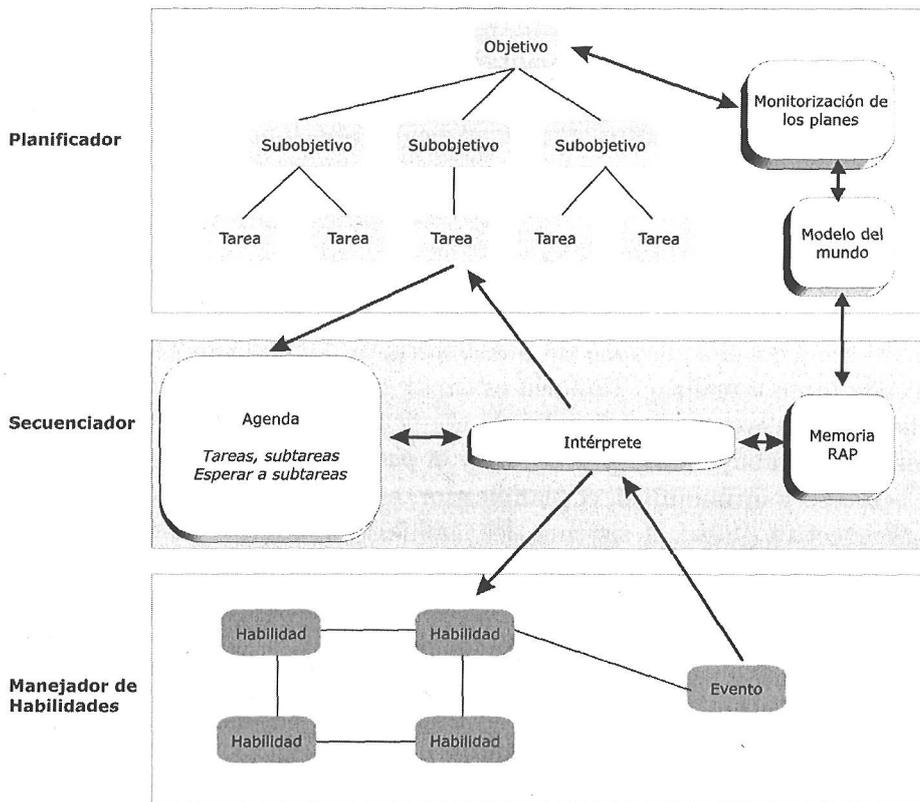


Figura 2.10: La arquitectura 3T



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

61

funciones deliberativas y modelos y estados del mundo (volviendo por tanto a las arquitecturas jerárquicas o funcionales (ver 2.2.2)) pero diferenciándose de éstas en los siguientes aspectos:

- **Utiliza sensores virtuales.** Aunque el modelo del mundo proporciona la percepción se usan sensores virtuales para aumentar el conocimiento del *presente* con el *pasado*.
- **El modelo se restringe a objetos de interés.** Este modelo es una de las partes donde se podrá fusionar información que provenga de distintos sensores, quedando limitada al sistema de representación elegido.
- **El procesamiento** de la percepción es **distribuido y asíncrono**.
- **Los errores sensoriales y la incertidumbre se pueden filtrar** utilizando fusión sensorial (respecto a las lecturas pasadas) para mejorar la fiabilidad.

En este tipo de arquitecturas el comportamiento final del robot emerge por la generación y la monitorización de una secuencia de comportamientos que podrán ser fusionados por lógica difusa o por votación.

Saphira

La arquitectura Saphira plantea como finalidad principal la construcción de agentes móviles autónomos con capacidad para atender, aprender y ejecutar tareas con robustez [KMRS97].

Se propone una arquitectura por niveles construida en torno a un mecanismo de representación interna, el espacio perceptual local (LPS). Existe una parte perceptora encargada de trasladar los datos de los sensores al LPS y extraer información de los mismos, y una parte efectora sobre la que se ejecutan los diferentes comportamientos. Existen tres tipos de comportamientos, los comportamientos de bajo nivel (reactivos), los comportamientos dirigidos por metas de nivel intermedio y los comportamientos de tareas de alto nivel.

En la arquitectura se establece una relación cliente/servidor con un servidor robótico consiguiendo mejorar la portabilidad y aislando al sistema de las dependencias de hardware.

El control de Saphira se basa en comportamientos. Los comportamientos reactivos se definen y coordinan utilizando lógica difusa [SRK97]. De

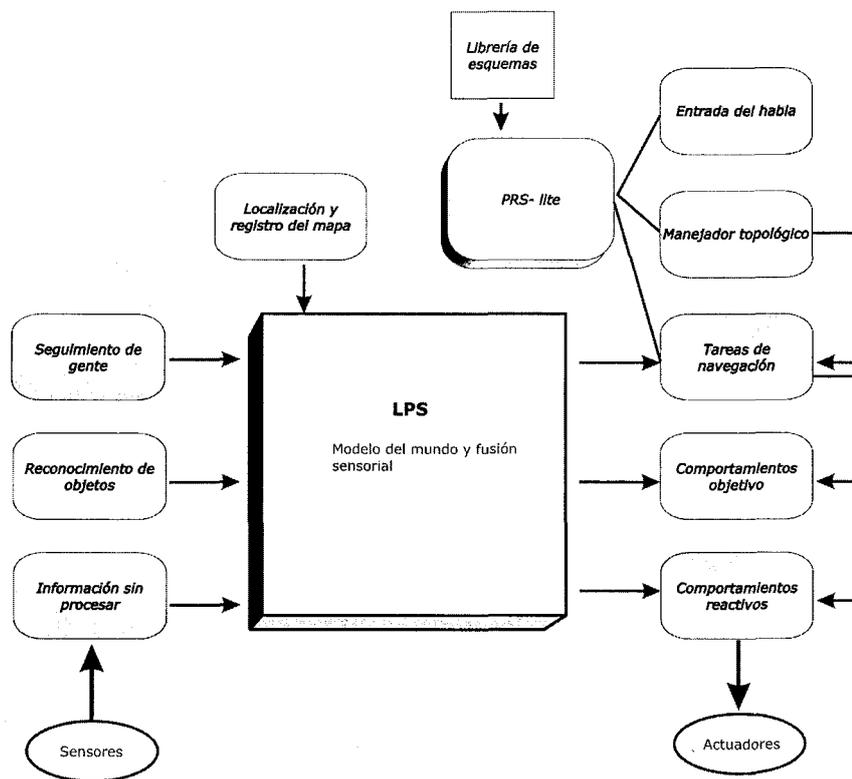


Figura 2.11: La arquitectura Saphira



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

63

esta manera las salidas de los comportamientos son reglas difusas que serán combinadas para obtener los comandos de velocidad y dirección del robot.

La coherencia del sistema se basa en mantener actualizados unos descriptores de objetos del entorno, denominados artefactos, sobre el LPS. Para ello, se generan características e hipótesis de objetos realizándose procesamiento *bottom-up* para la simbolización (características, hipótesis, artefactos) como procesamiento *top-down* de verificación (artefactos, hipótesis).

La selección y coordinación de comportamientos la realiza el controlador PRS-Lite (Procedural Reasoning System). PRS-Lite es un sistema para la representación y la deliberación sobre acciones y procedimientos en un entorno dinámico [IGR92], [IC93] que está basado en la arquitectura de agentes BDI [Hub99]. Es un sistema de planificación híbrido que intenta combinar la deliberación con la reactividad mediante el uso de planes meta-control. Algunas características de este controlador son: capacidad de integración de actividades dirigidas por objetivos o dirigidas por eventos, reactividad o descomposición jerárquica de tareas. Adicionalmente, este módulo incorpora capacidades como: gestión de procesos continuos en interacción, un amplio conjunto de mecanismos de control declarativos o el uso de niveles de satisfacción como respuesta tras la realización de una tarea.

PRS-Lite se basa en la utilización de esquemas de actividad que se definen como conjuntos ordenados de metas, integrados a su vez por metas simples. Una meta puede pertenecer a dos categorías, acción o secuenciación donde estas últimas permiten la descomposición jerárquica dando lugar a estructuras en árbol.

2.3. Resumen

Tal y como hemos visto, atendiendo al tipo de combinación utilizada, podemos clasificar las aplicaciones de fusión sensorial en distintos tipos. Algunas de las clasificaciones más utilizadas actualmente para estas aplicaciones son: la categorización a tres niveles, donde los procesos de fusión se suelen clasificar en un modelo jerárquico de tres niveles; la clasificación basada en entrada y salida, una clasificación más refinada que la anterior y la clasificación basada en la configuración sensorial, donde se tiene en cuenta el tipo de configuración de los sensores a utilizar.

En el campo de la robótica la combinación de información proveniente de varios sensores es fundamental para un funcionamiento robusto del robot,



siendo las arquitecturas de fusión sensorial uno de los pilares fundamentales para el diseño de robots autónomos. En este capítulo se ha planteado una revisión de las arquitecturas más utilizadas actualmente para la formalización de sistemas que requieran procesos de fusión sensorial. Estas arquitecturas las podemos agrupar como: centradas en la información, funcionales y basadas en comportamientos.

Las arquitecturas de fusión centradas en la información se basan en una división del proceso de fusión de acuerdo con la extracción de la información que tiene que ser fusionada. En este tipo de arquitecturas no se establece una secuencia de acciones explícitamente. Dentro de estas arquitecturas podemos destacar el modelo JDL o el modelo de fusión en cascada. Aunque este tipo de arquitecturas aportan una buena herramienta para el estudio y comprensión del problema a desarrollar, no guían de manera suficiente al diseñador para la consecución de su sistema.

Otro tipo de arquitecturas de fusión son las que están organizadas principalmente desde un punto de vista funcional, siendo en este caso explícita la secuencia de acciones a desarrollar. De estas arquitecturas podemos destacar el modelo Boyd o el modelo Ómnibus. Los modelos funcionales pueden tener el problema de proporcionar sistemas demasiado lentos a los estímulos y necesitar descripciones complejas del mundo.

También destacamos las arquitecturas basadas en el comportamiento. En estas arquitecturas se propone una unión entre la percepción y la acción. No obstante este tipo de sistemas tienen inconvenientes para resolver problemas complejos y suelen favorecer muy poco la modularidad y la reutilización de código.

Debido a todos estos inconvenientes, surgen en el campo de la robótica multitud de arquitecturas basadas en las ideas de los modelos de fusión anteriores, que intentan mejorar o tratar aspectos no contemplados previamente. Se han descrito algunas de estas arquitecturas híbridas y se han clasificado en tres grandes grupos: organizativas, jerarquía de estados y orientadas a modelo.

Las arquitecturas híbridas organizativas plantean la deliberación como un conocimiento global que abarca desde el tiempo pasado al futuro. La reactividad se establece como un comportamiento iniciado por una apreciación persistente del tiempo. Dentro de esta clasificación podemos enmarcar la arquitectura LAAS, AURA o SFX. Este tipo de arquitecturas suelen proporcionar buenos mecanismos para fusión de datos en niveles bajos aunque no ocurre lo mismo con la representación de los datos fusionados en niveles más



CAPÍTULO 2. FUSIÓN SENSORIAL: TIPOLOGÍA Y ARQUITECTURAS

65

elevados.

Las arquitecturas híbridas con jerarquía de estados plantean la deliberación utilizando un conocimiento tanto pasado como futuro. La reactividad se establece como un comportamiento puramente reflexivo utilizando únicamente conocimiento local. La deliberación se diseña utilizando estados temporales, no obstante esta división puede desembocar en clasificaciones no del todo lógicas. Un ejemplo de este tipo de arquitecturas lo podemos encontrar en la arquitectura 3T.

Las arquitecturas híbridas orientadas a modelo describen la deliberación como cualquier elemento relacionado con un comportamiento condicionado por una meta u objetivo. La reactividad se establece utilizando pequeñas unidades de control que operan en el presente pero que pueden utilizar conocimiento global proporcionado por un sensor virtual. El aparato deliberativo se diseña utilizando componentes de comportamiento, funciones deliberativas, modelos y estados del mundo. En este tipo de arquitecturas los mecanismos de fusión pueden quedar limitados por el sistema de representación elegido. Un ejemplo de este tipo de sistemas es la arquitectura Saphira.

Como se comentó anteriormente uno de nuestros objetivos es el proporcionar un modelo para la fusión sensorial capaz de manejar información incompleta o incierta. En el capítulo siguiente se analizará un formalismo Bayesiano para tratar con este tipo información, la programación Bayesiana, revisando sus ventajas al tratar con datos incompletos o inciertos. Además se comentarán algunas de las técnicas y aplicaciones probabilísticas más usadas en el ámbito de la robótica móvil formalizándolas utilizando únicamente este modelo y mostrando así que se trata de un formalismo general.





Universitat d'Alacant
Universidad de Alicante

“Una vez descartado lo imposible, lo que queda, por improbable que parezca, debe ser la verdad”

ARTHUR CONAN DOYLE

Capítulo 3

Programación e inferencia Bayesiana

La Programación Bayesiana (PB) es un formalismo probabilístico, basado en la inferencia Bayesiana y en el aprendizaje, para tratar la incertidumbre. Este formalismo se ha utilizado con éxito en la especificación y resolución de problemas complejos dentro del campo de la robótica móvil.

En este capítulo se introducirán los conceptos y métodos básicos de la programación Bayesiana sobre los que se fundamenta parte del trabajo presentado en esta tesis. A continuación se revisarán algunos de los métodos que permiten abordar computacionalmente los problemas de inferencia Bayesiana requeridos para la ejecución de un programa Bayesiano. También se describirán algunas de las aplicaciones, técnicas y modelos para tratar con incertidumbre, que se pueden especificar utilizando este formalismo y que demuestran que la PB es un formalismo general. Para finalizar se verá una de las aplicaciones de la programación Bayesiana más importantes en el campo de la robótica, los mapas Bayesianos. Los mapas Bayesianos proporcionan una aproximación Bayesiana a las tareas de construcción de mapas, que será revisada posteriormente.



3.1. Introducción a la PB

En el paradigma predominante de la robótica actual es el programador el que tiene una determinada concepción del entorno [LBDM04]. Esta concepción se suele describir en términos geométricos especificando la forma de los objetos y del mundo. El programador puede describir el mundo en términos analíticos porque las leyes físicas que dominan el mundo son conocidas. El entorno puede describirse también simbólicamente porque tanto los objetos como sus características pueden ser enumerados. De esta manera, con esta representación abstracta, se programa al robot; estos programas utilizarán nociones geométricas, analíticas y simbólicas. Podríamos decir que el programador *impone* su concepción del entorno al robot. Obviamente las dificultades de esta aproximación aparecen cuando el robot necesita unir esos conceptos abstractos con las lecturas obtenidas por los sensores y con las señales enviadas a sus actuadores: el problema central de estas dificultades es la no completitud de los modelos.

Cualquier modelo de un fenómeno real es incompleto. Siempre existen variables ocultas, que no se tienen en cuenta en el modelo, que influyen al fenómeno. Más aún, la percepción y el control robótico son intrínsecamente inciertos. El razonamiento con información incompleta continúa siendo, pues, un reto para los sistemas actuales. Un sistema robótico ha de afrontar el problema de utilizar una serie de percepciones incompletas y trabajar con ellas. Para la navegación robótica, por ejemplo, se han alcanzado los resultados más satisfactorios basándose en el cálculo probabilístico [DBM04b].

La programación Bayesiana (PB) [BG03], [LBDM04] es un formalismo probabilístico, propuesto recientemente, que se basa en la inferencia Bayesiana y el aprendizaje, para tratar con la incertidumbre. Este formalismo es uno de los más generales [DBM03a], siendo capaz de aglutinar diversos modelos tales como los filtros de Kalman, los modelos ocultos de Markov, las redes Bayesianas, las redes Bayesianas Dinámicas. . .

Las ventajas de este formalismo se pueden ver en [LBDM04] y se resumen a continuación:

- **Proporciona habilidades para tratar con información incompleta e incierta.** La base de la PB es tratar con el problema de programar un robot en un entorno real. En [LBDM04] se considera que esta dificultad es una consecuencia de la no completitud de los modelos.



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 69

De esta manera una de las ventajas de la aproximación propuesta es la habilidad de tener en cuenta la no completitud y la incertidumbre. Además la programación Bayesiana proporciona mecanismos para convertir la no completitud de la información en incertidumbre (utilizando aprendizaje), habilidad para razonar con la incertidumbre y habilidad para decidir teniendo en cuenta la incertidumbre.

- **Base teórica simple y concisa.** El formalismo de la PB está fundamentado en bases teóricas simples. Las preguntas esenciales se deben especificar de manera clara y formal pudiendo responderse mediante razonamiento matemático. Con las bases de la lógica y la inferencia Bayesiana los resultados experimentales pueden ser analizados y entendidos en detalle.
- **Proporciona un método de programación sistemático y simple.** La PB proporciona un método simple, sistemático y genérico para la especificación y resolución de problemas.
- **Homogeneización de las representaciones y los procesos de resolución.** Este método se basa en una única estructura de datos, llamada descripción, asociada a dos reglas de inferencia. Esta homogeneidad proporciona un desarrollo simple y genérico.
- **Obliga a especificar todas las hipótesis.** Como se ha comentado anteriormente la PB se basa únicamente en descripciones probabilísticas. Como una descripción es el único método para encapsular información relevante se obliga al programador a expresar exhaustivamente el conocimiento del que dispone para realizar una tarea.
- **Alta capacidad de expresión.** Como alta capacidad de expresión se entiende la capacidad de especificación y de realización de preguntas. Las descripciones probabilísticas ofrecen una gran capacidad de expresión para especificar modelos y cuestiones relativas a ellos.

Con la inferencia Bayesiana la PB tiene en cuenta explícitamente la incertidumbre y la no completitud de los modelos elegidos por el programador. Se ha demostrado en [LBDM04] que se pueden elaborar programas complejos, utilizando este marco de desarrollo, mediante la combinación de programas



simples. Pruebas experimentales sobre robots reales han demostrado la efectividad y la robustez de los programas construidos [KPBM03],[CB01].

El ámbito de aplicación de la programación Bayesiana es extenso. Aunque se planteó como un marco de desarrollo para la programación robótica y por tanto la mayoría de trabajos versan sobre este tema [KPBM03], [DBM04a], [CB01], [DO00], [CFBE02], [DBM04b] su uso se ha extendido a otros ámbitos, desde la formalización de modelos de vida artificial [APRS04], hasta la programación de video juegos [LHABL04].

3.1.1. Conceptos Básicos

A continuación se introducen los conceptos y la notación básica que se utilizará para la especificación de problemas utilizando el formalismo de la programación Bayesiana. Es importante familiarizarse con ambos ya que se utilizarán en los capítulos siguientes.

Definiciones y notación

En esta sección se introducirán los conceptos, postulados, definiciones, notación y reglas que son necesarios para definir un programa Bayesiano.

Proposiciones lógicas. En la PB se hace uso de proposiciones lógicas denotadas por nombres en minúsculas. Una proposición a se puede combinar para obtener más proposiciones lógicas utilizando los siguientes operadores: (\wedge, \vee, \neg) correspondiendo a la conjunción, disyunción y negación respectivamente.

Variables discretas. Las variables discretas se representan con nombres que empiezan por una letra mayúscula. Por definición una variable discreta X es un conjunto de proposiciones x_i donde dichas proposiciones son exclusivas mutuamente ($\forall_{i,j}, i \neq j, x_i \wedge x_j = \text{falso}$) y exhaustivas (como mínimo una proposición x_i es cierta). x_i representa que la variable X vale el i -ésimo valor. $|X|$ denota el cardinal del conjunto X (el número de proposiciones x_i).

La conjunción de dos variables X y Y se denota por $X \otimes Y$ y se define como el conjunto de $|X| \otimes |Y|$ proposiciones $x_i \wedge y_i$. $X \otimes Y$ es un conjunto exclusivo y mutuamente exhaustivo de proposiciones lógicas (por tanto es una nueva variable). Así pues la conjunción de n variables



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 71

es una nueva variable. La disyunción de dos variables, definida como un conjunto de proposiciones $x_i \vee y_i$ no forma una nueva variable al no ser dichas proposiciones mutuamente exclusivas.

Para trabajar con incertidumbre se asignarán probabilidades a las proposiciones. Se considera que para asignarle una probabilidad a una proposición a es necesario tener algún conocimiento previo, referenciado por la proposición π . Consecuentemente la probabilidad de una proposición a está siempre condicionada como mínimo por π . Para cada π diferente $P(\cdot|\pi)$ es una aplicación que asigna a cada proposición a un único valor real $P(a|\pi)$ en el intervalo $[0, 1]$.

Ciertamente también se trabajará con las probabilidades de la conjunción, disyunción y negación de proposiciones denotadas por $P(a \wedge b|\pi)$, $P(a \vee b|\pi)$ y $P(\neg a|\pi)$. También será interesante ver la probabilidad de que una proposición a este condicionada por conocimiento previo π y otra proposición b , es decir, $P(a|b \wedge \pi)$.

Por simplicidad y claridad cada vez que la variable X aparezca en una fórmula probabilística $\Phi(X)$ se debe entender como $\forall x_i \in X, \Phi(x_i)$.

Postulados de inferencia y reglas

Para el razonamiento probabilístico se definen únicamente dos reglas básicas a partir de las cuales se deducen todas las demás. Estas dos reglas utilizando probabilidades discretas son suficientes para solucionar cualquier problema de inferencia [Rob65], [Rob79].

- Regla de la conjunción. Dicha regla permite obtener la probabilidad de una conjunción de proposiciones

$$\begin{aligned} P(a \wedge b|\pi) &= P(a|\pi) \times P(b|a \wedge \pi) \\ &= P(b|\pi) \times P(a|b \wedge \pi) \end{aligned} \quad (3.1)$$

- Reglas de normalización. Especifica que la suma de las probabilidades de a y $\neg a$ es 1.

$$P(a|\pi) + P(\neg a|\pi) = 1 \quad (3.2)$$

A partir de la aplicación de las dos reglas anteriores se deducen las siguientes reglas referentes tanto a proposiciones como a variables:

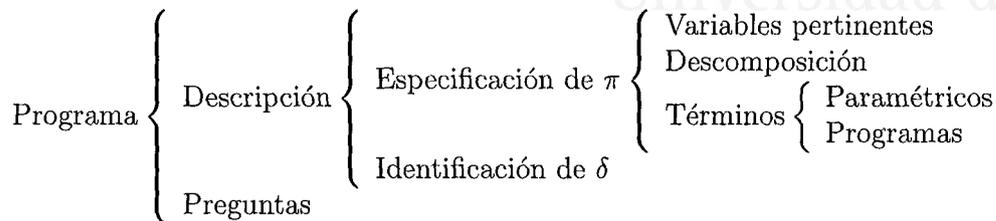


Figura 3.1: Estructura de un programa Bayesiano

- Regla de la disyunción para proposiciones.

$$P(a \vee b|\pi) = P(a|\pi) + P(b|\pi) - P(a \wedge b|\pi) \quad (3.3)$$

- Regla de la conjunción de variables o producto.

$$\begin{aligned} P(X \otimes Y|\pi) &= P(X|\pi) \times P(Y|X \wedge \pi) \\ &= P(Y|\pi) \times P(X|Y \wedge \pi) \end{aligned} \quad (3.4)$$

- Regla de normalización para variables.

$$\sum_X P(X|\pi) = 1 \quad (3.5)$$

- Regla de marginalización para variables.

$$\sum_X P(X \otimes Y|\pi) = P(Y|\pi) \quad (3.6)$$

3.1.2. Definición de un programa Bayesiano

Utilizando los postulados y reglas anteriores es posible definir un formalismo genérico que especifique modelos probabilísticos. Se define un programa Bayesiano como un medio de especificar una familia de distribuciones de probabilidad. Los elementos constituyentes de un programa Bayesiano (ver ecuación 3.1) son los siguientes:

Descripción. El propósito de la descripción es especificar un método efectivo para calcular una unión de distribuciones de un conjunto de variables



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 73

$\{X_1, X_2, \dots, X_n\}$ dado un conjunto de datos experimentales δ y un cierto conocimiento previo π . Esta unión de distribuciones se denota por $P(X_1 \otimes X_2 \otimes \dots \otimes X_n | \delta \otimes \pi)$

Conocimiento previo. Para especificar el conocimiento previo π se ha de tener en cuenta:

- Definir una serie de variables relevantes $\{X_1, X_2, \dots, X_n\}$ donde la unión de distribuciones esté definida.
- Descomponer la unión de distribuciones. Dada una partición de $\{X_1, X_2, \dots, X_n\}$ en k subconjuntos se definen k variables L^1, \dots, L^k donde cada una corresponde a uno de dichos subconjuntos. Cada variable L^i se obtiene como la conjunción de variables $\{X_{i_1}, X_{i_2}, \dots\}$ que pertenecen al subconjunto i . De esta manera:

$$P(X_1 \otimes X_2 \otimes \dots \otimes X_n | \delta \otimes \pi) = P(L_1 | \delta \otimes \pi) \times P(L_2 | L_1 \otimes \delta \otimes \pi) \times \dots \times P(L_k | L_{k-1} \otimes L_{k-2} \otimes \dots \otimes L_1 \otimes \delta \otimes \pi) \quad (3.7)$$

La hipótesis de independencia condicional proporciona un mecanismo de simplificación para (3.7). Una hipótesis de independencia condicional para la variable L_i se define escogiendo algunas variables X_i de entre las variables que aparecen en la conjunción $L_{i-1} \otimes L_{i-2} \otimes \dots \otimes L_1$, llamando R_i a la conjunción de dichas variables y disponiendo:

$$P(L_i | L_{i-1} \otimes L_{i-2} \otimes \dots \otimes L_1 \otimes \delta \otimes \pi) = P(L_i | R_i \otimes \delta \otimes \pi) \quad (3.8)$$

Obtenemos que:

$$P(X_1 \otimes X_2 \otimes \dots \otimes X_n | \delta \otimes \pi) = P(L_1 | R_1 \otimes \delta \otimes \pi) \times \dots \times P(L_k | R_k \otimes \delta \otimes \pi) \quad (3.9)$$

Dicha simplificación de una unión de distribuciones como un producto de distribuciones simples se llama descomposición.

- Definir los términos. Cada distribución $P(L_i | R_i \otimes \delta \otimes \pi)$ que aparece en el producto de (3.9) se asocia con una forma paramétrica (una función $f_\mu(L_i)$) u otro programa Bayesiano. En general μ es un vector de parámetros que puede depender de R_i , de δ o de ambos. El aprendizaje se da cuando algunos de estos parámetros se obtienen utilizando el conjunto de datos δ .



Preguntas. Dada una descripción (por ejemplo $P(X_1 \otimes X_2 \otimes \dots \otimes X_n | \delta \otimes \pi)$), una pregunta se obtiene dividiendo $\{X_1, X_2, \dots, X_n\}$ en tres conjuntos: Las variables buscadas, las variables conocidas y las variables desconocidas.

Se definen por tanto las variables *Buscado*, *Conocido* y *Desconocido* como la conjunción de las variables que pertenecen a dichos conjuntos. Se define una pregunta como la distribución:

$$P(\text{Buscado} | \text{Conocido} \otimes \delta \otimes \pi) \quad (3.10)$$

3.1.3. Ejecución de un programa Bayesiano

Ejecutar un programa Bayesiano supone tener dos capacidades básicas, inferencia Bayesiana y toma de decisiones.

Inferencia Bayesiana

Dada una unión de distribuciones $P(X_1 \otimes X_2 \otimes \dots \otimes X_n | \delta \otimes \pi)$, siempre es posible obtener cualquier pregunta posible mediante la siguiente regla de inferencia general, donde ($B = \text{Buscado}$, $C = \text{Conocido}$, $D = \text{Desconocido}$):

$$\begin{aligned} P(B | C \otimes \delta \otimes \pi) &= \sum_D P(B \otimes D | C \otimes \delta \otimes \pi) \\ &= \frac{\sum_D P(B \otimes D \otimes C | \delta \otimes \pi)}{P(C | \delta \otimes \pi)} \\ &= \frac{\sum_D P(B \otimes D \otimes C | \delta \otimes \pi)}{\sum_D \sum_B P(B \otimes D \otimes C | \delta \otimes \pi)} \\ &= \frac{1}{\sum} \times \sum_D P(B \otimes D \otimes C | \delta \otimes \pi) \end{aligned} \quad (3.11)$$

Donde la primera igualdad se obtiene aplicando la regla de marginalización (ecuación 3.6) y la segunda se obtiene a partir de la regla del producto (ecuación 3.4). En la tercera ecuación de (3.11) el denominador es un término de normalización por lo que por convención se reemplaza por $\frac{1}{\sum}$.

La inferencia Bayesiana es un problema complejo. El problema general de obtener la inferencia exacta es un problema NP-duro (NP-hard) [Coo90]



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 75

al igual que el problema general de aproximar dicha inferencia [DL93]. Se han utilizado numerosas técnicas y relajaciones a las inferencias exactas para poder obtener tiempos de computación admisibles. En la sección 3.2 se revisarán algunas de estas técnicas.

Toma de decisiones

Para una distribución dada se pueden utilizar diferentes políticas de actuación. Se pueden buscar los mejores valores (los de mayor probabilidad) u obtener valores aleatoriamente a partir de la distribución. En la sección 3.2 se revisarán algunas de las técnicas de toma de decisiones u obtención de valores que se pueden aplicar sobre una distribución dada.

3.2. Inferencia Bayesiana

Como se ha comentado anteriormente la inferencia Bayesiana es un problema complejo. No obstante se han utilizado numerosas técnicas y relajaciones al proceso de inferencia exacta para poder obtener tiempos de computación admisibles. A continuación se realizará una revisión de algunas de estas heurísticas y técnicas.

Tal y como se comentó en la sección 3.1.3, dada una distribución $P(X_1 \otimes X_2 \otimes \dots \otimes X_n)$ siempre se puede contestar a la ecuación de inferencia 3.11. Si definimos la descomposición de esta distribución como¹:

$$P(X_1 \otimes X_2 \otimes \dots \otimes X_n) = P(L_1) \times P(L_2|R_2) \times \dots \times P(L_k|R_K)$$

Siendo R_i la conjunción de la variables: $R_i = \{L_{i-1} \otimes L_{i-2} \otimes \dots \otimes L_2 \otimes L_1\}$

Podemos sustituir en la ecuación 3.11 la unión de variables por su descomposición obteniendo que:

$$\begin{aligned} P(B|C) &= \frac{1}{\Sigma} \times \sum_D P(B \otimes D \otimes C) = \\ &= \frac{1}{\Sigma} \times \sum_D \left(P(L_1) \times \prod_{i=2}^k P(L_i|R_i) \right) \end{aligned} \quad (3.12)$$

¹A partir de esta sección se omitirá, para facilitar su lectura, el conocimiento previo y los datos ($\delta \otimes \pi$) de la parte derecha de las fórmulas



Existen principalmente dos problemas a la hora de obtener tiempos de computación admisibles en este proceso de inferencia Bayesiano [BG03]: encontrar las modas de una distribución y aplicar la marginalización de variables en un espacio dimensional grande.

3.2.1. Simplificación

Antes de enfrentarse a los dos problemas anteriores puede ser de gran ayuda realizar algunas simplificaciones simbólicas que permitan reducir el número de sumas necesarias para calcular la distribución $P(\text{Buscado}|\text{Conocido})$.

Simplificaciones independientes de la pregunta a realizar

Una de las simplificaciones más utilizadas en el proceso de inferencia Bayesiano es el que realizan las redes Bayesianas. El proceso de inferencia de una red Bayesiana consiste en obtener el valor de $P(X_i|\text{Conocido})$. Los valores de las diferentes variables que aparecen en *Conocido* se llaman *evidencias*. Una red Bayesiana se define utilizando la siguiente descomposición:

$$P(X_1 \otimes \dots \otimes X_N) = \prod_{i=1}^N P(X_i|R_i)$$

Esta descomposición incluye las dependencias lógicas entre las variables X_i . No obstante esta descomposición desde un punto de vista computacional puede no ser demasiado interesante ya que posiblemente desemboque en un cálculo intensivo para obtener la pregunta $P(X_i|\text{Conocido})$.

Simplificar la computación de las redes Bayesianas supone encontrar otra descomposición $\prod_{j=1}^M P(\tilde{L}_j|\tilde{R}_j)$ que reduzca los cálculos pero que sea equivalente.



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 77

$$\begin{aligned}
 P(X_1 \otimes \dots \otimes X_N) &= \prod_{i=1}^N P(X_i | R_i) \\
 &= \prod_{j=1}^M P(\tilde{L}_j | \tilde{R}_j) \\
 &= \prod_{j=1}^M \frac{P(\tilde{L}_j \otimes \tilde{R}_j)}{P(\tilde{R}_j)} \\
 &= \prod_{j=1}^M \frac{P(\tilde{L}_j \otimes \tilde{R}_j)}{\sum_{\tilde{R}_j} P(\tilde{L}_j \otimes \tilde{R}_j)}
 \end{aligned}$$

El algoritmo JLO [JLO90] busca una descomposición como la anterior y desarrolla los cálculos necesarios para calcular $P(X_i | \text{Conocido})$. Se trata de una generalización del algoritmo de paso de mensajes de Pearl [Pea82], utilizado para actualizar la probabilidades de una red Bayesiana con forma de poliárbol. Es importante resaltar que el algoritmo JLO es muy eficiente para calcular preguntas de la forma $P(X_i | \text{Conocido})$ pero puede ser ineficiente para calcular la solución de otras preguntas, como por ejemplo las de forma $P(X_i \otimes \dots \otimes X_k | \text{Conocido})$ donde se requiere la realización de sumas complementarias.

Simplificaciones dependientes de la pregunta realizada

Aunque se pueden llegar a simplificaciones teniendo en cuenta únicamente la descomposición de la unión de variables se pueden obtener mejores resultados utilizando la pregunta a realizar a la distribución. Un grupo de preguntas se puede aplicar en cadena, una detrás de otra, para obtener un mayor beneficio. Además este proceso no está limitado a un tipo de descomposición determinado, pudiéndose aplicar a cualquier descomposición de la unión de variables inicial. A continuación se enumeran tres de los métodos aplicables al proceso de simplificación de una distribución independientemente de la pregunta a realizar:

- **Simplificación sencilla.** Dada una determinada descomposición de la distribución pueden ocurrir tres cosas:
 - Cuando un término es uniforme puede simplificarse englobándose dentro de una constante $\frac{1}{\Sigma}$.
 - Cuando un término es una distribución donde todas las variables son conocidas (*Conocido*) entonces puede simplificarse.



- Cuando un término es una distribución donde todas las variables son *Buscado* o *Conocido* puede factorizarse fuera del sumatorio.
- **Eliminación de las distribuciones que sumen la unidad.** Puede ser conveniente el reordenar los términos de una suma para encontrar términos cuya suma sea 1.
- **Ley distributiva general.** Se puede realizar una reordenación en los términos de una suma para minimizar el número de operaciones a realizar. El algoritmo correspondiente a esta simplificación se puede encontrar en detalle en [AJ00].

3.2.2. Búsqueda de las modas de una distribución

Si deseamos conocer la distribución de una variable *Buscado* debemos tener en cuenta que esta puede estar formada por la unión de múltiples variables o incluso ser continua. De esta manera es prácticamente imposible obtener de manera exhaustiva la probabilidad $P(\text{Buscado}|\text{Conocido})$. Se puede construir entonces una representación aproximada de la distribución u obtener valores directamente de la misma. En ambos casos la dificultad estriba en encontrar las modas donde la mayoría de la densidad de la distribución está concentrada. Esta tarea puede ser muy complicada ya que la mayoría de la probabilidad puede estar concentrada en pequeños subespacios del espacio de búsqueda.

Construir representaciones explícitas de las distribuciones

- **Representación utilizando un conjunto de puntos.** Si deseamos construir una representación explícita de una distribución n-dimensional $P(X)$ la representación más simple que podemos utilizar es la codificación de $P(X)$ como una muestra formada por un conjunto de N puntos, es decir: $P(X) = \{x_1, \dots, x_N\}$. Este tipo de representación aún siendo simple puede resultar funcional si la distribución se utiliza en un proceso de muestreo directo o indirecto.

Se llama muestreo directo si el problema consiste en obtener un conjunto de M puntos de una distribución. Obtener muestras directamente de $P(X)$ consiste en extraer un número aleatorio i entre 1 y N devolviendo el correspondiente punto del conjunto $\{x_1, \dots, x_N\}$.



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 79

Hablamos de muestreo indirecto cuando el proceso de muestreo consiste en estimar la distribución $P(Y)$ como una integral en el espacio de X tal que $P(Y) = \int_X P(X) \times P(Y|X)$. Esta integral se puede estimar utilizando el método de integración Monte Carlo.

$$P(Y) \approx \frac{1}{N} \times \sum_{i=1}^N P(Y|[X = x_i])$$

O también codificando $P(X)$ como pares, ponderando x_i por w_i como $P(X) = \{(x_1, w_1), \dots, (x_N, w_N)\}$ y realizando una estimación mediante filtros de partículas:

$$P(Y) \approx \frac{\sum_{i=1}^N (w_i \times P(Y|[X = x_i]))}{\sum_{i=1}^N w_i}$$

No obstante este tipo de representaciones sufren de varias limitaciones. Por una parte es complicado representar espacios multidimensionales relativamente grandes con un número razonable de puntos. Por otra parte es complicado colocar los puntos donde las modas tienen un mayor significado. Por último estos métodos son incapaces de generalizarse a la vecindad de los puntos de la muestra y por tanto $P(x)$ sólo puede ser evaluada para estos puntos.

- **MRBT.** Un Árbol Binario de Multiresolución (MRBT) es una representación explícita con capacidad de generalización. La idea principal de un MRBT es que las regiones de probabilidad elevada deben ser representadas utilizando alta resolución y por contra las regiones con baja probabilidad se pueden representar utilizando resoluciones menores. Se pueden encontrar más detalles en [BB03].

Obtención de muestras directa

Obtener muestras de una distribución probabilística continua siendo un problema complejo principalmente para las distribuciones n-dimensionales.



- **Obtención de muestras de una distribución Normal.** Si se dispone de un generador uniforme de números aleatorios es posible, en algunos casos, utilizar una función de transformación para obtener medidas de una determinada distribución. La función de transformación Box-Muller es una de las más importantes [BM58]. Esta transformación permite generar un conjunto de números aleatorios normales utilizando otro conjunto de números aleatorios generados uniformemente. Otra posibilidad es diagonalizar la matriz de varianza Σ , obteniendo sus vectores propios utilizándolos para generar el nuevo conjunto de números aleatorios.

- **Funciones de distribución.** Cuando se dispone de la función de distribución $F(X)$ correspondiente a una función $f(X)$ tal que $F(X) = \int_{-\infty}^X f(i)di$, el problema de muestreo se reduce a invertir la función de distribución $F(X)$. Obtener una muestra de $f(x)$ consiste en conseguir un número aleatorio $r = [0, 1]$ y utilizarlo para calcular $x = F^{-1}(r)$.
 No obstante generalmente no se dispone de la función de distribución y por tanto su inversa no está disponible. El cálculo explícito de $F(X)$ y su inversa aproximada es posible aunque sólo para espacios dimensionales no demasiado extensos.

- **Métodos Monte Carlo.** Los métodos de Monte Carlo agrupan distintos algoritmos de muestreo en espacios dimensionales grandes. A continuación se presentarán algunos de los métodos más populares utilizados en la actualidad.
 - *Muestreo por importancia.* Dada una distribución $P(X)$ para la cual no disponemos de ningún método de muestreo directo pero si podemos evaluarla para cada punto x_i del espacio, si suponemos que existe una distribución $Q(X)$ más simple que $P(X)$ sobre la cual podemos evaluar cualquier punto x_i entonces podemos aplicar la técnica de muestreo por importancia. Esta técnica de muestreo consiste en generar N pares $\{(x_1, \alpha_1), \dots, (x_N, \alpha_N)\}$ donde los puntos x_i se extraen de $Q(X)$ siendo:

$$\alpha_i = \frac{P(x_i)}{Q(x_i)}$$



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 81

- *Muestreo por descarte.* De la misma manera que en el apartado anterior, suponiendo las mismas características de $P(X)$ y de $Q(X)$, si se cumple que $\exists c$ tal que $\forall x, [c \times Q(x) > P(x)]$ se puede obtener un punto de $P(X)$ obteniendo uno de $Q(X)$ y aceptándolo con probabilidad:

$$\frac{1}{\Sigma} \times \frac{c \times Q(x_i)}{P(x_i)}$$

Siendo $\frac{1}{\Sigma}$ un término normalizador.

- *Muestreo Metrópolis.* Los métodos de muestreo anteriores funcionan correctamente siempre que la distribución $Q(X)$ sea una buena aproximación de $P(X)$. No obstante para espacios dimensionales complejos es difícil encontrar alguna distribución que cumpla esta característica.

El algoritmo de Metrópolis [MRR⁺53] utiliza un proceso Markoviano generando una secuencia de estados x^t , donde un nuevo estado x^t depende de su estado anterior x^{t-1} . En vez de utilizar una distribución $Q(X)$ se utiliza una distribución $Q(X_i, X^t)$ que depende del estado actual x^t . Esta distribución no tiene por que ser compleja.

De esta manera, suponiendo que el estado actual es x^t un valor x_i se genera a partir de $Q(X_i, X^t)$. Para aceptar x_i se ha de calcular:

$$a = \frac{P(x_i) \times Q(x_i, x^t)}{P(x^t) \times Q(x^t, x_i)}$$

Donde si $a > 1$ se acepta x_i siendo $x^{t+1} = x_i$ en otro caso se acepta con probabilidad a siendo $x^{t+1} = x^t$.

No obstante uno de los problemas de este tipo de métodos es el tiempo requerido para obtener un equilibrio en la red de estados, además también puede llegar a ser complejo el determinar cuándo se ha alcanzado este equilibrio.

- *Muestreo Gibbs.* El muestreo Gibbs es un método para extraer datos de una distribución como mínimo bidimensional. Se puede ver como un método Metrópolis en el cual $Q(X)$ se define en términos de la distribución condicional de $P(X)$. De esta manera se asume



que siendo $P(X)$ demasiado compleja para obtener muestras la distribución $P(X_i|X_1 \otimes \dots \otimes X_{i-1} \otimes X_{i+1} \otimes \dots \otimes X_N)$ es tratable.

Métodos variacionales

Cuando se convierte un problema de inferencia probabilística en un problema de optimización se habla de un método variacional. De esta manera se consiguen reutilizar las herramientas de optimización para resolver este tipo de problemas de inferencia. La idea principal reemplazar la distribución $P(X)$ por una aproximación $Q(X)$ y calcular la divergencia Kullback-Leibler entre ambas distribuciones. Así se define la energía de una configuración X como:

$$E(X) = -\log(P(X)) - \log(Z)$$

y la distancia Kullback-Leibler, o energía variacional libre como:

$$F(Q, P) = \sum_X \left(Q(X) \times \log \frac{Q(X)}{P(X)} \right) - \log(Z)$$

Siendo esta distancia mínima cuando $P(X) = Q(X)$. Para más información se puede consultar [JGJS99], [JW02], [YFW02].

3.2.3. Marginalización de variables

Obtener el valor de $P(\text{Buscado}|\text{Conocido})$ para un determinado valor de *Conocido* es un problema complicado, que supone marginalizar la unión de variables en el espacio *Desconocido*. Al igual que con la variable *Buscado*, la variable *Desconocido* puede estar formada por la conjunción de múltiples variables o incluso ser continua.

Dada la ecuación 3.12, el cálculo de la marginalización de $P(\text{Buscado}|\text{Conocido})$ requiere encontrar las modas de la distribución 3.13 en el espacio de búsqueda definido por la variable *Desconocido*, donde la mayoría de la densidad probabilística está concentrada y contribuye a la suma.

$$P(L_1) \times \prod_{i=2}^k P(L_i|R_i) \tag{3.13}$$



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 83

Integración analítica

Aunque el cálculo integral es la base de la inferencia Bayesiana no existe ningún método analítico que permita evaluar todas las integrales con formas complejas en espacios multidimensionales. Los métodos de aproximación numérica (como el trapezoidal o el de Simpson) pueden utilizarse en espacios reducidos pero no están adaptados para espacios multidimensionales grandes.

No obstante en la inferencia probabilística uno de los casos más conocidos donde sí se puede aplicar la integración analítica es cuando el integrando es un producto de distribuciones normales generalizadas y el modelo es lineal o puede ser linealizado. En este caso debemos calcular:

$$I(Y) = \int_{-\infty}^{\infty} P(X) \times P(Y|X) \times dX^D$$

Donde $P(X) = G(X, \mu_X, \sigma_X)$ y $P(Y|X) = G(Y, A \cdot X, \sigma_Y)$, obteniendo de esta manera la siguiente solución analítica:

$$I(Y) = G(Y, (A \cdot \mu_X), (A \cdot \sigma_X \cdot A^T + \sigma_Y))$$

Donde A es una matriz constante. Esta solución analítica se usa ampliamente en los filtros de Kalman (ver [Kal60]).

Métodos de Monte Carlo para la integración numérica

La finalidad de los métodos de Monte Carlo es aproximar eficientemente la integral n-dimensional:

$$I = \int P(X) \times Q(X) \times dX^D$$

Si asumimos que no se puede visitar cada localización x_i en el espacio de integración entonces la solución más sencilla consiste en obtener muestras del espacio de integración X y estimarlo mediante:

$$\hat{I} = \frac{1}{N} \times \sum_{i=1}^N P(x_i) \times Q(x_i)$$

Las distribuciones multidimensionales normalmente se concentran en una pequeña región del espacio de integración, conocida como conjunto típico. El



número de puntos a extraer del espacio de integración X debe ser suficiente para cubrir la región que llamaremos T , que contiene la mayoría de la masa de $P(x)$.

En vez de explorar el espacio de estados uniformemente, los métodos de Monte Carlo utilizan la información proporcionada por $P(X)$ para realizar una exploración más eficiente, aproximando la integral I mediante la estimación de la esperanza de la función $Q(X)$ bajo la distribución $P(X)$:

$$I = \int P(X) \times Q(X) \times dX^D = \langle Q(X) \rangle \quad (3.14)$$

De esta manera la exactitud de la estimación Monte Carlo no depende de la dimensionalidad del espacio de integración. Para más información ver [Nea93]

Integración simple (perfecta) de Monte Carlo

Si suponemos que somos capaces de obtener eficientemente un conjunto de muestras $\{x_1, \dots, x_N\}$ de la distribución $P(X)$, estas se pueden utilizar para obtener el estimador:

$$\hat{I} = \frac{1}{N} \times \sum_{i=1}^N Q(x_i)$$

La capacidad de obtener muestras eficientemente sólo se da cuando $P(X)$ es una distribución simple con un método de extracción directo o un producto de distribuciones donde se pueden obtener muestras directamente utilizando el algoritmo de Gibbs.

Extracción por importancia en la integración Monte Carlo

Si no somos capaces de extraer valores a partir de la distribución $P(X)$ pero sí de una distribución más simple $S(X)$ llamada distribución de muestreo se pueden extraer N puntos a partir de $S(X)$. Si esos puntos se hubieran generado a partir de $P(X)$ entonces se podría estimar I utilizando la ecuación 3.14. No obstante, como estos puntos se obtienen a partir de $S(X)$ podemos encontrarnos con que si el valor extraído de $S(X)$ es mayor que el correspondiente a $P(X)$ entonces éste se sobre-representará, ocurriendo a la inversa en caso contrario. Para solucionar este problema se introducen unos pesos w_i para añadir importancia a cada punto del estimador, de tal manera que:



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 85

Universitat d'Alacant
Universidad de Alicante

$$w_i = \frac{P(x_i)}{S(x_i)},$$
$$\hat{I} = \frac{1}{\left(\sum_{i=1}^N w_i\right)} \times \sum_{i=1}^N (w_i \times Q(x_i))$$

Este método de integración se usa especialmente en los filtros de partículas (ver [Thr02]).

3.3. Especificación de modelos probabilísticos utilizando PB

La programación Bayesiana es un formalismo general capaz de aglutinar algunos de los modelos probabilísticos más utilizados en la actualidad. A continuación se presentan algunos de estos modelos formalizados utilizando la PB. Se puede encontrar más información sobre la capacidad de representación de la PB y los modelos probabilísticos más utilizados en el campo de la robótica en [DBM03a], [BG03].

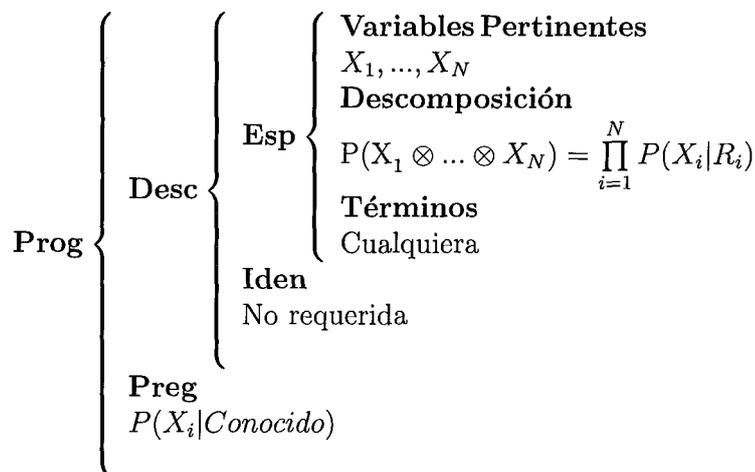
3.3.1. Redes Bayesianas

Uno de los modelos probabilísticos de propósito general más utilizados son las redes Bayesianas. Estas fueron introducidas por [Pea88] como un método para tratar con información probabilística e incierta. Son el resultado de la unión de la teoría probabilística y de la teoría de grafos. A los nodos del grafo se les asocian variables de la misma manera que a sus aristas se les asocian dependencias condicionales. En términos de la programación Bayesiana se especificarían utilizando el siguiente programa:



ESPECIFICACIÓN DE MODELOS PROBABILÍSTICOS UTILIZANDO PB

86



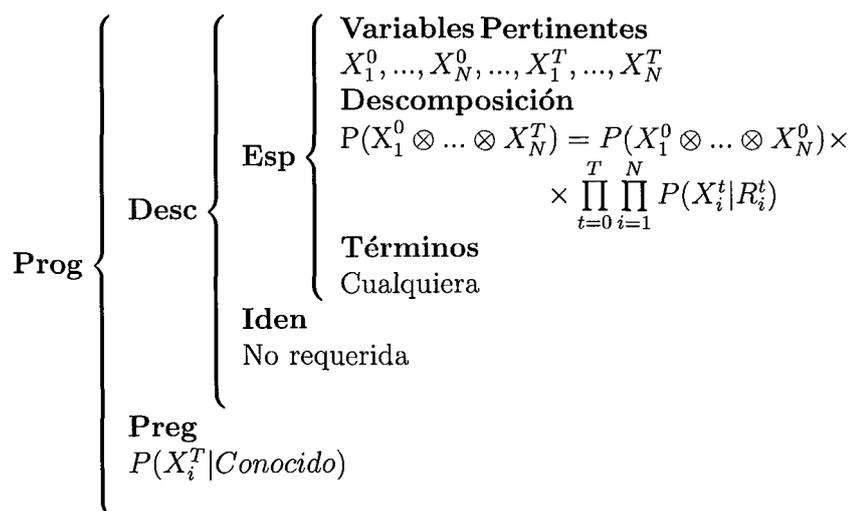
Donde las variables pertinentes no están limitadas a ninguna forma. Por otra parte la descomposición de estas variables es específica siendo un producto de distribuciones con una única variable X_i condicionada por la conjunción de sus variables padres R_i . Las formas paramétricas no se limitan aunque usualmente se utilizan tablas. Para la pregunta $P(X_i | \text{Conocido})$ se han desarrollado técnicas de inferencia eficientes aunque para preguntas más generales pueden aparecer problemas.

Redes Bayesianas dinámicas

Las redes Bayesianas clásicas se extendieron para tratar con una nueva variable, el tiempo, y ser capaces de modelar procesos estocásticos. En las redes Bayesianas dinámicas [DK89], dado un grafo representando la estructura del conocimiento en el instante t , se define la red resultante como la repetición de la estructura desde el tiempo inicial al tiempo final. Se supone que la estructura que representa el conocimiento es invariante en el tiempo y se considera que éste es discreto. Una red Bayesiana dinámica se puede definir utilizando el siguiente programa Bayesiano:



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 87



Siendo R_i^t una conjunción de variables obtenidas del conjunto $\{X_1^t, \dots, X_{i-1}^t\} \cup \{X_1^{t-1}, \dots, X_N^{t-1}\}$. La variable X_i^t depende únicamente de sus padres en el tiempo t y algunas variables del estado previo.

Una red Bayesiana dinámica puede ser expandida en el tiempo y considerarla como una red Bayesiana normal. De esta manera se pueden aplicar las mismas técnicas de inferencia aplicables a una red Bayesiana normal.

3.3.2. Estimación Bayesiana recursiva

La estimación Bayesiana recursiva es la denominación genérica de diferentes modelos probabilísticos de series temporales. Estos modelos se pueden definir utilizando el siguiente programa Bayesiano:



**ESPECIFICACIÓN DE MODELOS PROBABILÍSTICOS
UTILIZANDO PB**

88

$$\text{Prog} \left\{ \begin{array}{l} \text{Desc} \left\{ \begin{array}{l} \text{Esp} \left\{ \begin{array}{l} \text{Variables Pertinentes} \\ S_0, \dots, S_N, \dots, O_0, \dots, O_N \\ \text{Descomposición} \\ P(S_0 \otimes \dots \otimes S_N \otimes O_0 \otimes \dots \otimes O_N) = \\ P(S_0) \times P(O_0|S_0) \times \prod_{i=1}^N (P(S_i|S_{i-1}) \times P(O_i|S_i)) \\ \text{Términos} \\ P(S_0), P(S_i|S_{i-1}), P(O_i|S_i) \end{array} \right. \\ \text{Iden} \\ \text{No requerida} \end{array} \right. \\ \text{Preg} \\ P(S_{t+k}|O_0 \otimes \dots \otimes O_t) \quad \text{donde} \quad \left\{ \begin{array}{l} (k = 0) \text{ Filtrado} \\ (k > 0) \text{ Predicción} \\ (k < 0) \text{ Suavizado} \end{array} \right. \end{array} \right.$$

Siendo $\{S_0, \dots, S_N\}$ series temporales de variables de estado y $\{O_0, \dots, O_N\}$ variables temporales de observación. La descomposición $P(S_i|S_{i-1})$ es conocida como *modelo del sistema* o *modelo de transición*. $P(O_i|S_i)$ se conoce como *modelo de observación* y expresa lo que puede ser observado en el instante i cuando el sistema está en el estado S_i .

Normalmente a este tipo de modelos se les pregunta sobre cual es la probabilidad de distribución de un estado en $t + 1$ conociendo las observaciones del instante 0 a t , es decir $P(S^{t+k}|O^0 \otimes \dots \otimes O^t)$. El caso más común es el filtrado, cuando se pregunta por $k = 0$, es decir, se busca el estado presente conociendo las observaciones pasadas.

Los filtros Bayesianos tienen una importante propiedad recursiva que los hace especialmente interesantes. $P(S_t|O_0 \otimes \dots \otimes O_t)$ se puede obtener a partir de $P(S_{t-1}|O_0 \otimes \dots \otimes O_{t-1})$:

$$P(S_t|O_0 \otimes \dots \otimes O_t) = P(O_t|S_t) \times \sum_{S_{t-1}} (P(S_t|S_{t-1}) \times P(S_{t-1}|O_0 \otimes \dots \otimes O_{t-1}))$$

La predicción y el suavizado no se pueden simplificar tanto como el filtrado y requieren por tanto el realizar grandes sumas que pueden representar un problema computacional.



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 89

Modelos Ocultos de Markov

Los modelos ocultos de Markov son un modelo muy popular de filtro Bayesiano. Un modelo oculto de Markov se puede definir utilizando el siguiente programa Bayesiano:

$$\text{Prog} \left\{ \begin{array}{l} \text{Desc} \left\{ \begin{array}{l} \text{Esp} \left\{ \begin{array}{l} \text{Variables Pertinentes} \\ S_0, \dots, S_t, O_0, \dots, O_t \\ \text{Descomposición} \\ P(S_0 \otimes \dots \otimes S_t \otimes O_0 \otimes \dots \otimes O_t) = \\ P(S_0) \times P(O_0|S_0) \times \prod_{i=2}^t (P(S_i|S_{i-1}) \times P(O_i|S_i)) \\ \text{Términos} \\ P(S_0) \equiv \text{Matriz} \\ P(S_i|S_{i-1}) \equiv \text{Matriz} \\ P(O_i|S_i) \equiv \text{Matriz} \end{array} \right. \\ \text{Iden} \\ \text{No requerida} \end{array} \right. \\ \text{Preg} \\ P(S_1 \otimes S_2 \otimes \dots \otimes S_{t-1} | S_t \otimes O_0 \otimes \dots \otimes O_t) \end{array} \right.$$

Donde se consideran todas las variables discretas. Tanto el modelo de transición como el modelo de observación se especifican utilizando matrices de probabilidad. La pregunta más común que se suele realizar a un modelo oculto de Markov tiene la forma $P(S_1 \otimes S_2 \otimes \dots \otimes S_{t-1} | S_t \otimes O_0 \otimes \dots \otimes O_t)$, es decir, se desea obtener la serie de estados más probable que haya llevado al estado actual conociendo las observaciones pasadas. Esta pregunta se puede obtener de manera óptima utilizando el algoritmo de Viterbi [For73].

Filtros de partículas

Una implementación particular de los filtros Bayesianos son los filtros de partículas. En un filtro de partículas la distribución $P(S_{t-1} | O_0 \otimes \dots \otimes O_{t-1})$ se aproxima por un conjunto de N partículas con pesos proporcionales a su probabilidad. Esta aproximación se trata de un proceso dinámico donde las partículas se mueven de acuerdo con el modelo de transición $P(S_t | S_{t-1})$ y se disponen sus pesos dependiendo del modelo de observación $P(O_t | S_t)$.

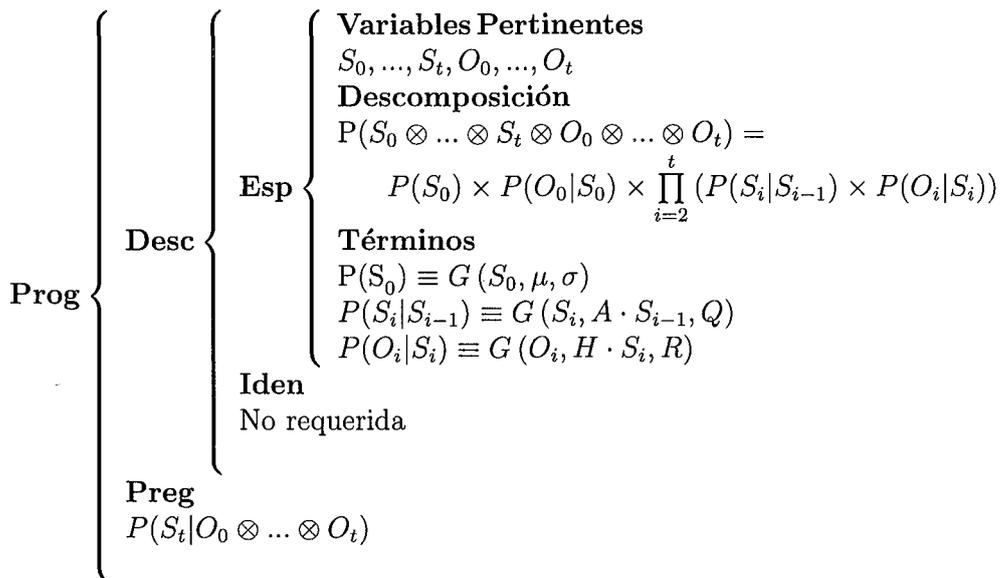


ESPECIFICACIÓN DE MODELOS PROBABILÍSTICOS UTILIZANDO PB

90

Filtros de Kalman

Los filtros de Kalman son una especialización de los filtros de partículas. Un filtro de Kalman se puede especificar utilizando el siguiente programa Bayesiano:



Donde el modelo de transición $P(S_i|S_{i-1})$ y el modelo de observación $P(O_i|S_i)$ se especifican utilizando distribuciones normales con medias determinadas por funciones lineales de la variable condicionada. Gracias a esta hipótesis es posible solucionar analíticamente la pregunta $P(S_t|O_0 \otimes \dots \otimes O_t)$, lo que proporciona un mecanismo eficiente para su obtención. Cabe notar que todas las variables utilizadas en un filtro de Kalman son continuas.

En el caso de que no exista ninguna transición u observación lineal se pueden utilizar series de Taylor de primer orden para considerar estos modelos como localmente lineales. Esta generalización es conocida como filtros de Kalman extendidos.

3.3.3. Mixture models

Los modelos de adición o mixture models, intentan aproximar una distribución de un conjunto de variables añadiendo un conjunto de distribuciones



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 91

simples. Los modelos más populares están formados por una unión de Gaussianas:

$$\left. \begin{array}{l} \text{Prog} \\ \text{Desc} \\ \text{Esp} \\ \text{Iden} \\ \text{Preg} \end{array} \right\} \left\{ \begin{array}{l} \text{Variables Pertinentes} \\ X_1, \dots, X_N, \mu_1, \dots, \mu_M, \sigma_1, \dots, \sigma_M, H \\ \text{Descomposición} \\ P(X_1 \otimes \dots \otimes X_N \otimes \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M \otimes H) = \\ P(X_1 \otimes \dots \otimes X_N \otimes \mu_1 \otimes \dots \otimes \mu_M) \times P(H) \times \\ \times P(X_1 \otimes \dots \otimes X_N | \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M \otimes H) \\ \text{Términos} \\ P(\mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M) \equiv \text{Uniforme} \\ P(H) \equiv \text{Tabla} \\ P(X_1 \otimes \dots \otimes X_N | \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M \otimes [H = i]) \\ = P(X_1 \otimes \dots \otimes X_N | \mu_i \otimes \sigma_i) \\ \text{Iden} \\ P(H) \\ \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M \\ \text{Preg} \\ P(\text{Buscado} | \text{Conocido} \otimes \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M) \\ = \sum_H P(X_1 \otimes \dots \otimes X_N | \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M \otimes H) \\ = \sum_{i=1}^M (P([H = i]) \times P(\text{Buscado} | \text{Conocido} \otimes \mu_i \otimes \sigma_i)) \end{array} \right.$$

Donde $\mu_1, \dots, \mu_M, \sigma_1, \dots, \sigma_M$ corresponde a las medias y desviaciones de las M distribuciones. H es una variable discreta que obtiene M valores y que se utiliza como variable de selección. Conociendo el valor de H se supone que la unión de distribuciones se reduce a uno de sus componentes en la distribución:

$$\begin{aligned} P(X_1 \otimes \dots \otimes X_N | \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M \otimes [H = i]) \\ = P(X_1 \otimes \dots \otimes X_N | \mu_i \otimes \sigma_i) \end{aligned}$$

La variable H puede ser considerada independiente del resto (como el caso anterior) o dependiente, como en el caso de los *expert mixture models* descritos por Jordan en [JR94], [MJ96].

La identificación es un paso muy importante donde tanto los valores de los parámetros como los de la distribución $P(H)$ se buscan para encontrar el

ESPECIFICACIÓN DE MODELOS PROBABILÍSTICOS UTILIZANDO PB

92

mejor ajuste posible. Este proceso se realiza normalmente utilizando el algoritmo EM. Para más información sobre este algoritmo consultar [DLR77], [MK97].

Las preguntas realizadas a la unión de distribuciones son de la forma $P(\text{Buscado}|\text{Conocido} \otimes \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M)$ donde *Buscado* y *Conocido* son conjunciones de algunas de las variables X_1, \dots, X_N . Aunque los parámetros μ y σ son conocidos no ocurre lo mismo con la variable H que siempre permanece oculta. De esta manera solucionar la pregunta anterior requiere realizar una suma sobre los posibles valores de H para obtener la unión de distribuciones tal que:

$$\begin{aligned} P(\text{Buscado}|\text{Conocido} \otimes \mu_1 \otimes \dots \otimes \mu_M \otimes \sigma_1 \otimes \dots \otimes \sigma_M) \\ = \sum_{i=1}^M (P([H = i]) \times P(\text{Buscado}|\text{Conocido} \otimes \mu_i \otimes \sigma_i)) \end{aligned}$$

3.3.4. Aproximaciones de máxima entropía

Las aproximaciones de máxima entropía juegan un papel muy importante en las aplicaciones con orientación física. En [Jay95] podemos encontrar una buena presentación de este tipo de aproximaciones. Un modelo de máxima entropía puede ser descrito utilizando la PB de la siguiente manera:

$$\begin{array}{l} \text{Prog} \left\{ \begin{array}{l} \text{Desc} \left\{ \begin{array}{l} \text{Variables Pertinentes} \\ X_1, \dots, X_N \\ \text{Descomposición} \\ P(X_1 \otimes \dots \otimes X_N) = \\ = \frac{1}{\Sigma} \times e^{-\sum_{i=1}^M (\lambda_i \times f_i(X_1 \otimes \dots \otimes X_N))} \\ = \frac{1}{\Sigma} \times \prod_{i=1}^M e^{-(\lambda_i \times f_i(X_1 \otimes \dots \otimes X_N))} \\ \text{Términos} \\ f_1, \dots, f_m \end{array} \right. \\ \text{Iden} \\ \lambda_1 \otimes \dots \otimes \lambda_M \end{array} \right. \\ \text{Preg} \\ P(\text{Buscado}|\text{Conocido}) \end{array} \right. \end{array}$$



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 93

Donde las variable X_1, \dots, X_N no están limitadas. La descomposición se realiza mediante un producto de funciones exponenciales, $e^{-(\lambda_i \times f_i(X_1 \otimes \dots \otimes X_N))}$ donde cada función f_i se conoce como función observable. Una función observable puede ser cualquier función real definida en el espacio X_1, \dots, X_N , tal que su valor esperado pueda ser obtenido de la siguiente manera:

$$\langle f_i(X_1 \otimes \dots \otimes X_N) \rangle = \sum_{X_1 \otimes \dots \otimes X_N} (P(X_1 \otimes \dots \otimes X_N) \times f_i(X_1 \otimes \dots \otimes X_N))$$

Los límites de un problema dado suelen expresarse con M valores reales F_i llamados nivel límite. La identificación trata de, una vez conocidos los valores límite F_i , encontrar multiplicadores de Lagrange λ_i que maximicen la entropía de la distribución $P(X_1 \otimes \dots \otimes X_N)$.

3.4. Mapas Bayesianos

Uno de los problemas cruciales en la robótica es el conseguir que un robot sea capaz de modelar su entorno. La mayoría de las aproximaciones para realizar esta área se basan en la localización Markoviana o variantes de los filtros de Kalman. No obstante en este tipo de modelos las variables de acción se utilizan únicamente como variables de entrada, ya que el modelo no está pensado para obtener las distribuciones de probabilidad sobre las acciones. En esta sección se presenta el modelo de los mapas Bayesianos, basado en la programación Bayesiana como una aproximación a las tareas de construcción de mapas robóticos.

3.4.1. Definición

Un mapa Bayesiano (MB) es una descripción, en el sentido de la programación Bayesiana robótica, que se basa en una distribución de una unión de variables $P(R \otimes L_t \otimes L'_t \otimes A)$, donde:

- R es una variable de percepción donde el robot lee los valores de la variable utilizando sus sensores físicos.
- L_t es una variable de localización en el instante t



- L'_t se define igual que la variable anterior pero en el instante t' (sin perder generalidad se puede asumir que $t' > t$)
- A es una variable de acción (se proporcionan las acciones a realizar utilizando esta variable).

Para que un MB se pueda utilizar en la práctica hace falta algo más que una descripción probabilística, hace falta que permita la creación de comportamientos. Se llama comportamiento elemental a cualquier pregunta de la forma $P(A_i|X)$ donde A_i es una parte de A y X es una parte de las variables del mapa (excluyendo a A_i). Se puede considerar un comportamiento como *no elemental*, por ejemplo si su realización implica alternancia temporal de varios comportamientos o se requiere la combinación de comportamientos con otros conocimientos.

Una *guía* para asegurarse de que un mapa pueda generar comportamientos, se basa en que un mapa conteste de manera pertinente a las tres preguntas:

- Localización: $P(L_t|R)$
- Predicción: $P(L'_t|A \otimes L_t)$
- Control: $P(A|L_t \otimes L'_t)$

Por pertinente se entiende que estas tres distribuciones asociadas a estas tres preguntas sean informativas, es decir, que su entropía esté suficientemente lejos de su máximo. Aunque esta guía no se especifique formalmente las tres capacidades de predicción, localización y control parecen ser capacidades necesarias, muy referenciadas en la literatura, que aseguran la navegabilidad. Para cada ejemplo de MB desarrollado se deberá probar que sus respuestas sean pertinentes: esto nos garantizará el buen funcionamiento del mapa en cuestión.

3.4.2. Localización, predicción y control

La distribución $P(L_t|R)$ es una de las partes constituyentes de la localización. Este término puede ser suficiente para una localización sin ambigüedad, si la representación del entorno elegida no sufre de *perceptual aliasing* (es decir, no existen dos lugares diferentes que generen las mismas percepciones).



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 95

Este tipo de localización *inmediata* toma el nombre de inferencia directa de la posición [Fil01]. En cambio, en los entornos que sufren de *perceptual aliasing*, un sólo dato sensorial no es suficiente para localizar el robot, y se hace necesario tomar en consideración el historial de la evolución del robot en el entorno, utilizando la pregunta $P(L'_t|A \otimes L_t)$. Es muy corriente encontrar esta solución en la literatura, especialmente en los formalismos de localización Markoviana y los PDMPO (Procesos de Decisión de Markov Parcialmente Observables), donde se describe el encadenamiento de las dos preguntas $P(L_t|P)$ y $P(L'_t|A \otimes L_t)$. $P(L'_t|A \otimes L_t)$ se conoce como matriz de transición o matriz estocástica. En este caso, este término es “una instancia del pasado”: L_t corresponde al lugar donde nos encontrábamos en el instante anterior, A es la acción que acaba de ser aplicada (convirtiéndose en una lectura pasiva que se puede considerar como un sensor), y por último L'_t describe los conocimientos sobre el lugar que se acaba de alcanzar.

Pero si este término se considera como una instancia “en el futuro”, puede servir para predecir el lugar de llegada L'_t , sabiendo la acción A que va a ser aplicada y el lugar actual L_t . El interés de esa anticipación o simulación de una acción es creciente en robótica, especialmente en el campo de la robótica bioinspirada. La existencia de tales capacidades en los seres vivos ha sido destacada por numerosos autores en las ciencias cognitivas [Cha02], [Ber97]. En el formalismo de los mapas Bayesianos, se ha considerado como primordial la existencia de más de un término de predicción para realizar el proceso de planificación.

La última pregunta, $P(A|L_t \otimes L'_t)$, permite, conociendo la posición actual del robot y aquella que el robot debe alcanzar, elegir la acción a aplicar, esbozando de esta manera el mecanismo de control a elegir.

3.4.3. Utilización de un mapa Bayesiano

Existen muchas maneras de utilizar las respuestas a estas tres preguntas en el marco de un programa de control robótico. Una primera manera bastante simple consiste en encadenar en el tiempo las tres preguntas, en un algoritmo determinista, como por ejemplo en el algoritmo de la figura 3.2. En este algoritmo el robot se localiza inicialmente, luego se utiliza la pregunta de predicción para dejar de esta manera al usuario la elección del lugar de llegada L'_t , después, con arreglo al lugar seleccionado como fin y del lugar reconocido por la localización, el robot elige la acción apropiada. Este programa pregunta al usuario un lugar como objetivo, y se esfuerza luego en



```

01  r ← obtenerLecturasSensores();
02  lt ← extraer (P(Lt|[R = r]))
03  para todo a ∈ A
04      proponerObjetivo (extraer (P(Lt + Dt|[A = a][Lt = lt]))
05  fpara
06  o ← PreguntarObjetivo()
07  a ← extraer (P(A|[Lt = lt][Lt + Dt] = o))

```

Figura 3.2: Algoritmo para la resolución de las preguntas de predicción, localización y control

alcanzarlo.

Es posible igualmente utilizar el control sobre la predicción del paso de tiempo anterior. El cálculo de la localización $P(L_t|R)$ a partir del mapa puede ser costoso, especialmente si hace falta invertir el término $P(R|L_t)$. En este caso, podemos obtener la posición actual calculando el paso de tiempo anterior: $t - \Delta t, l_t \leftarrow \text{extraer}(P(L_t|A_{L_t - \Delta t}))$.

En el instante t , obtenemos el valor de los sensores r , y comprobamos la probabilidad del término $P([R = r]|[L_t = l_t])$, que es la probabilidad de percibir r sobre los sensores, suponiendo que en efecto l_t sea el previsto en el paso de tiempo anterior. Si consideramos este valor numérico más alto que un umbral k , que depende de la experiencia sobre la pregunta anterior, entonces todo se desarrollaría de acuerdo con las predicciones. Podemos pues, para generar el control, suponer que el robot se encuentra en efecto en l_t , y de esta manera no sería necesario el cálculo explícito de la localización $P(L_t|R)$. Se resume esta segunda posibilidad de control robótico con el algoritmo de la figura 3.3.

Otra posibilidad de utilización de un mapa consiste en resolver una tarea disponiendo una pregunta probabilística a la descripción que define el mapa. Llamemos comportamiento elemental a una pregunta de la forma $P(A_i|X)$.

La mayoría del tiempo, para generar un comportamiento elemental, se fija una localización l' a alcanzar, y se propone la pregunta probabilística $P(A|[L_t = l'] \otimes [R = r])$.

Esta pregunta se puede resolver de dos maneras. La primera manera tiene relación con el primer algoritmo (3.2) y se basa en encadenar las preguntas de localización y de control, utilizando la siguiente derivación:



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 97

```

01  r ← obtenerLecturasSensores();
02  v ← P([R = r] | [Lt = lt])
03  si(v < k)
04      lt ← extraer (P(Lt | [R = r]))
05  fsi
06  a ← eleccionacción();
07  l't ← extraer (P(Lt + Dt | [A = a] [Lt + lt]))

```

Figura 3.3: Segundo algoritmo para la resolución de las preguntas de predicción, localización y control

$$\begin{aligned}
 P(A | [L_{t'} = l'] \otimes [R = r]) &= \frac{P(A \otimes [L_{t'} = l'] \otimes [R = r])}{P([L_{t'} = l'] \otimes [R = r])} = \\
 &= \frac{1}{\Sigma} P(A \otimes [L_{t'} = l'] \otimes [R = r]) \\
 &= \frac{1}{\Sigma} \sum_{L_t} P(A \otimes L_t \otimes [L_{t'} = l'] \otimes [R = r]) \\
 &= \frac{1}{\Sigma} \sum_{L_t} \left(\frac{P([L_{t'} = l'] \otimes [R = r]) \times P(L_t | [R = r]) \times P(A | [L_{t'} = l'] \otimes L_t)}{P(A | [L_{t'} = l'] \otimes L_t)} \right) \\
 &= \frac{1}{\Sigma} \sum_{L_t} P(L_t | [R = r]) \times P(A | [L_{t'} = l'] \otimes L_t)
 \end{aligned}$$

En este cálculo, los dos términos $P(L_t | [R = r])$ y $P(A | [L_{t'} = l'] \otimes L_t)$ se obtienen a partir de las preguntas probabilísticas del mapa. La elección de estos es importante, pues tienen una semántica evidente, son los términos de localización y de control. Por otra parte, el término de localización permite, por ejemplo, fijar gráficamente el conocimiento del robot sobre su posición. Esta inferencia puede ser vista como una versión probabilística del algoritmo 3.2, donde la extracción sobre $P(L_t | [R = r])$ es substituida por una estimación sobre el variable L_t .

La segunda manera para contestar a la pregunta global $P(A | [L_{t_0} = l_0] \otimes [R = r])$ no hace intervenir explícitamente los términos de localización y de control. Se trata sencillamente de hacer esta pregunta probabilística al mapa Bayesiano. Ahora nada fuerza al mapa a hacer aparecer en su descomposición estos términos particulares. Así la inferencia se basará únicamente en los términos que aparecen en la descomposición, y será entonces difícil encontrar en este cálculo algo parecido a las preguntas de localización, predicción, o control. Por ejemplo, si la descomposición es

$P(R \otimes L_t \otimes L_{t_0} \otimes A) = P(L_t) \times P(R | L_t) \times P(A | L_t) \times P(L_{t'} | L_t)$, hacer la pregunta $P(A | L'_t \otimes R)$, se calcula:



$$P(A|L_t \otimes R) = \frac{1}{\Sigma} \sum_{L_t} P(L_t) \times P(R|L_t) \times P(A|L_t) \times P(L_t|L_t)$$

Este método es más natural en el marco de la programación Bayesiana robótica, resolviendo una pregunta probabilística y volviendo a trabajar sistemáticamente con la descomposición escogida, no obstante de esta manera se pierde la explicación del cálculo, perdiendo la posibilidad de visualizar, por ejemplo, el conocimiento sobre la localización del robot.

3.4.4. Grafo inducido por un mapa Bayesiano

Tradicionalmente, los mapas se representan utilizando grafos. La teoría de los grafos está unida a los problemas de planificación de caminos. Los dos problemas principales dentro de este ámbito son la búsqueda del camino más corto y el viajante de comercio.

En el formalismo de los MB no aparece directamente un grafo en la definición pero podemos considerar, sin embargo, que un MB es un modelo probabilística que representa los conocimientos subjetivos que el robot tiene del mapa objetivo. Este mapa puede representarse formalmente por un grafo con arcos etiquetados $\langle X, R, F \rangle$, donde:

- X es un conjunto de lugares
- R es una relación sobre X : los arcos $\rho(X \times X)$
- F es una aplicación $F : R \rightarrow S$ que le asocia una etiqueta a cada arco siendo un símbolo del conjunto S

Se puede consultar [Sto00] para más de detalles sobre los grafo etiquetados en general. Es posible obtener de cualquier mapa Bayesiano el grafo objetivo correspondiente, sencillamente especificando:

- $X = L_t$: los valores de la variable de los lugares L_t son los nodos del grafo
- $S = A \times [0, 1]$: el arco activo del estado l_1 al estado l_2 se etiqueta con una acción que tiene adjunta una probabilidad p tal que $p = P([L'_t = l_2] | [A = a] [L_t = l_1])$.



CAPÍTULO 3. PROGRAMACIÓN E INFERENCIA BAYESIANA 99

Por ejemplo, supongamos que tenemos una variable de lugar L_t que tiene tres valores l_1, l_2, l_3 posibles, entonces el grafo tiene tres nodos, con los mismos nombres l_1, l_2, l_3 . Supongamos además que $P(L_t' | [A = a_1][L_t = l_1])$ es uniforme, entonces el grafo posee tres arcos, iniciados a partir de l_1 , que van hacia l_1, l_2 y l_3 , y que llevan cada uno la etiqueta $\langle a_1, 1/3 \rangle$.

Sin embargo, si esta variable se restringe a esta definición del grafo, induce a que la mayoría de los grafos que obtengamos de los mapas sean completos. En la práctica, como deseamos que los grafos hagan resaltar las informaciones más importantes de las distribuciones de probabilidades, sólo representaremos los arcos que corresponden a los valores de probabilidad más elevados. Esta “limitación” de la traducción de un mapa Bayesiano a un grafo es la consecuencia de la diferencia de riqueza de expresión entre los formalismos probabilísticos y los gráficos.

3.4.5. Operadores de combinación

En las secciones anteriores se ha descrito un formalismo para la definición de mapas probabilísticos. Este formalismo se complementa con dos herramientas que permiten unir mapas existentes. Estas herramientas son dos operadores de conjunto: la superposición y la abstracción de mapas.

La superposición de mapas es un operador binario que une dos mapas superponiendo los lugares de los mapas subyacentes. De esta manera el robot es capaz de localizarse en dos mapas simultáneamente, lo que enriquece su vocabulario descriptivo y su interacción con el entorno. El segundo operador, la abstracción de mapas proporciona un mecanismo para construir jerarquías de modelos. En la jerarquía resultante la localización en los modelos de alto nivel se basa en una competición probabilística de los niveles inferiores. Para más información sobre estos dos operadores se puede consultar [DBM04a], [DBM03b].

3.5. Resumen

La programación Bayesiana es un formalismo probabilístico basado en la inferencia y el aprendizaje Bayesiano. Su principal finalidad es tratar con información incompleta o incierta. La programación Bayesiana proporciona además de una base teoría simple y concisa, un método de programación sistemático, la homogeneización de las representaciones y los procesos de



resolución y una alta capacidad de expresión.

Un programa Bayesiano se define como un medio de especificar una familia de distribuciones de probabilidad. Para ejecutar un programa Bayesiano se requieren dos capacidades básicas, inferencia Bayesiana y toma de decisiones. Cabe tener en cuenta que la inferencia Bayesiana es un problema complejo. No obstante se han utilizado numerosas técnicas y relajaciones al proceso de inferencia exacta para poder obtener tiempos de computación razonables. Entre estas técnicas se encuentran los procesos de simplificación simbólica, las búsquedas de las modas de una distribución probabilística y la marginalización de variables.

La programación Bayesiana es un formalismo general que aglutina algunos de los modelos probabilísticos más utilizados actualmente, entre ellos se pueden destacar las redes Bayesianas, las redes Bayesianas dinámicas, los problemas de estimación Bayesiana recursiva, los modelos ocultos de Markov, los filtros de Kalman, los filtros de partículas, los modelos de adición (mixture models), las aproximaciones de máxima entropía. . .

Una de las aplicaciones más importantes de la programación Bayesiana son los mapas Bayesianos. La mayoría de aproximaciones capaces de modelar el entorno por el que se desplaza un robot se basan en la localización Markoviana o variantes del filtro de Kalman. En este tipo de modelos las variables de acción únicamente se utilizan como variables de entrada, por contra los mapas Bayesianos son capaces de obtener las distribuciones de probabilidad sobre las acciones proporcionando un mecanismo eficaz para el modelado del entorno de un robot.

Ahora ya disponemos de un formalismo apto para trabajar con información incompleta o incierta que además es capaz de formalizar un buen conjunto de aplicaciones y técnicas Bayesianas orientadas al campo de la robótica. A partir de este punto nos dispondremos a diseñar y especificar nuestro modelo para la fusión robótica, el modelo FSR-BAY.



Universitat d'Alacant
Universidad de Alicante

“Si has construido un castillo en el aire, no has perdido el tiempo, es allí donde debería estar. Ahora debes construir los cimientos debajo de él.”

GEORGE BERNARD SHAW

Capítulo 4

FSR-BAY: Modelo Probabilístico para la Fusión Sensorial Robótica

Partiendo de los inconvenientes de las arquitecturas de fusión clásicas comentados en el capítulo 2 se propondrá un modelo para la fusión sensorial robótica. Este modelo proporcionará un sistema formal para especificar los procesos de fusión teniendo en cuenta la incertidumbre y la no completitud de la información. Por último el modelo se complementará con dos operadores que permitirán fusionar unidades de manera complementaria o competitiva.

4.1. Introducción

Los robots autónomos ofrecen un buen campo para el estudio de las arquitecturas de fusión sensorial. La autonomía se puede evaluar por la robustez con la que un robot lleva a cabo diferentes tareas en entornos no del todo conocidos. En un sistema robótico, para alcanzar unas metas y reaccionar



a diversos eventos, es necesaria una determinada organización. Como se ha visto anteriormente la estructura de control de un robot autónomo es recomendable que contenga tanto capacidades deliberativas como capacidades reactivas, refinando las tareas a realizar de manera dinámica de acuerdo con el contexto actual.

Según [ACF⁺98] una arquitectura de control robótica ha de tener las siguientes propiedades:

- **Facilitar la programación.** Un robot útil no se puede diseñar para un único entorno con una tarea programada en detalle. Debe ser capaz de realizar múltiples funciones, descritas utilizando niveles abstractos. Estas funciones deben ser fácilmente combinables de acuerdo con la tarea a realizar.
- **Autonomía y adaptabilidad.** Un robot debe ser capaz de realizar acciones y redefinir o modificar su tarea y su propio comportamiento de acuerdo con la tarea a realizar y el contexto de ejecución percibido.
- **Reactividad.** Un robot ha de tener en cuenta los límites temporales de las tareas a realizar y hacerlos compatibles con sus metas (incluyendo su propia seguridad).
- **Comportamientos consistentes.** Las reacciones del robot a determinados eventos deben ser guiadas por los objetivos de su tarea.
- **Robustez.** La arquitectura de control ha de ser capaz de explotar la redundancia de las funciones de procesamiento. La robustez requiere por tanto un control descentralizado.
- **Extensibilidad.** La integración de nuevas funciones y la definición de nuevas tareas debe ser sencilla. Se deben proporcionar mecanismos de aprendizaje que permitan aprender distintas habilidades al agente robótico.

Obviamente también sería deseable que se tuviera en cuenta la incertidumbre de los datos con los que se trabaja. En el caso de que se utilice una representación interna del mundo, ésta debería basarse y utilizar mecanismos para trabajar tanto con la no completitud como con la incertidumbre.

Por otra parte, es importante remarcar que no existe ningún modelo común para la fusión sensorial hasta el momento, aunque hay muchos modelos



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

y arquitecturas que proponen mecanismos similares para desarrollar los procesos de fusión [Elm02].

En el capítulo anterior se revisaron algunas de las arquitecturas más utilizadas en el ámbito de la fusión sensorial, tanto para realizar tareas de fusión genérica como para especificar sistemas de fusión orientados al campo de la robótica móvil. Se destacaron principalmente tres tipos de arquitecturas: las arquitecturas centradas en información, los modelos funcionales, y las arquitecturas basadas en el conocimiento.

Las arquitecturas centradas en la información proporcionan una buena herramienta para el estudio y la comprensión de un determinado sistema pero por otra parte no guían al diseñador en la elección de métodos apropiados para realizar un diseño e implementación correctos. Este tipo de arquitecturas necesitan proporcionar mecanismos de fusión al diseñador, que le permitan especificar el problema de una manera más concreta.

Los modelos funcionales no adolecen del problema principal de las arquitecturas centradas en la información, sí que especifican la secuencia de acciones a desarrollar. No obstante pueden tener el problema de proporcionar sistemas demasiado lentos a los estímulos, además de necesitar decisiones complejas del mundo. Este tipo de modelos requieren una descripción del mundo, que muchas veces puede limitar el campo de acción de los sistemas diseñados utilizando esta base funcional.

Las arquitecturas basadas en el conocimiento reaccionan de manera apropiada en entornos dinámicos, de hecho son las arquitecturas mejor adaptadas al trabajo en dichos entornos, pero tienen serias dificultades para resolver problemas complejos, además de no favorecer la modularidad del sistema.

Por otra parte en la mayoría de las arquitecturas robóticas se utilizan de manera conjunta algunos de los modelos de fusión anteriores. Las arquitecturas híbridas organizativas suelen proporcionar buenos mecanismos para la fusión de datos en niveles bajos, no obstante no ocurre lo mismo con la representación de los datos fusionados en los niveles más elevados. Las arquitecturas híbridas con jerarquía de estados especifican la responsabilidad del aparato deliberativo utilizando estados temporales que utilizan la velocidad de ejecución como medida de prioridad. De esta manera los distintos algoritmos a utilizar se deben clasificar teniendo en cuenta sus necesidades temporales. No obstante esta división puede desembocar en clasificaciones no del todo lógicas. Las arquitecturas híbridas orientadas a modelo utilizan sensores virtuales junto con procesamiento distribuido y asíncrono de la percepción. Los mecanismos de fusión sensorial se utilizan para el filtrado de los datos, para



la combinación de comportamientos (normalmente utilizando lógica difusa o mecanismos de votación) y para el modelo de objetos de interés (donde se fusionará la información que provenga de distintos sensores). No obstante los mecanismos de fusión quedan limitados por el sistema de representación elegido.

En este capítulo se propondrá un modelo probabilístico para la fusión sensorial robótica multinivel. Este modelo se basará en la programación Bayesiana, proporcionando una herramienta para el diseño y la implementación de sistemas robóticos, dotando al programador mecanismos concretos para el desarrollo de las tareas de fusión. Se trata por tanto de un modelo con una fuerte fundamentación matemática. Más concretamente este modelo realizará una descomposición modular de la información basándose para ello en distribuciones probabilistas. Será la manera de unir dichas distribuciones, llamadas también unidades Bayesianas, lo que nos permitirá desarrollar las tareas de fusión utilizando los operadores diseñados para tal efecto. Una distribución probabilista estará formada por un conjunto de variables previamente especificadas por el programador para un problema en concreto. Las unidades guiarán al diseñador del sistema, tanto a una determinada descomposición de la información como a una programación modular y reutilizable. Se propone de esta manera un modelo que unifica la representación de la información y los procesos de fusión de bajo y alto nivel.

4.1.1. Líneas Generales

La filosofía general que se utilizará en el modelo propuesto, como se ha comentado anteriormente, se basa en trabajar con distribuciones probabilísticas, que representan procesos diferenciados, y combinarlas entre sí, de manera correcta, proporcionando procesos más complejos que los anteriores. Esta filosofía es diferente de la mayoría de aproximaciones previas en las que se intenta extraer, de un mundo intratable y muy complejo, una jerarquía de pequeños modelos.

Tanto la incertidumbre como la no completitud de la información y las capacidades de aprendizaje son temas que han sido tratados de manera secundaria por la mayoría de las arquitecturas de fusión presentadas [ACF⁺98]. Una de las principales características que se tuvieron en cuenta a la hora de elaborar este modelo fue el que trabajara de manera implícita con la incertidumbre y la no completitud de la información. La mayoría de modelos previos no contemplan ningún mecanismo para el manejo y la utilización de



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

105

datos inciertos. Un sistema real ha de tener en cuenta que para trabajar con el mundo ha de utilizar modelos que tengan en cuenta la incertidumbre de los datos con los que trabaja.

El formalismo de la programación Bayesiana se ha utilizado para la especificación y el diseño del modelo propuesto. Las ventajas obtenidas del uso de este formalismo son principalmente el diseño modular, la fundamentación matemática y el tener en cuenta de manera implícita la incertidumbre y la no completitud de la información a tratar. Además, la inferencia Bayesiana proporciona mecanismos de aprendizaje que se integran de manera directa en nuestro modelo. De esta manera se pueden realizar tareas de aprendizaje para cada una de las unidades pertenecientes al sistema. Más aún, es importante destacar que el formalismo Bayesiano utilizado obliga a la especificación de las distribuciones de probabilidad o bien al aprendizaje de las mismas, con lo cual siempre se producirá una salida incluso para situaciones no previstas con anterioridad.

Por último se propondrá un modelo que soporte la construcción de sistemas robóticos. La programación Bayesiana ya proporciona los mecanismos necesarios para la especificación y la resolución de problemas robóticos [LBDM04]. Un programa Bayesiano se puede adaptar al modelo propuesto para poder utilizar los mecanismos de fusión que se proporcionarán más adelante, obteniendo todas las ventajas que esto conlleva.

Más concretamente, el modelo propuesto se basará en los siguientes principios:

- **Modularidad.** Cuando se trabaja con agentes autónomos que interactúan en el mundo real es necesario descomponer las tareas a realizar en distintos módulos. El mundo real es muy complejo y difícil de tratar y por tanto es necesaria esta descomposición para cualquier tipo de tarea a realizar. Además, la división modular facilita la integración de nuevos módulos y la reutilización de los mismos.
- **Unificación en la representación.** Tanto para información de bajo nivel como para decisiones de alto nivel sería conveniente utilizar la misma representación. No obstante esto no implica que se utilicen distintos tipos de fusión dependiendo de la información con la que se trabaja.
- **Adaptabilidad y aprendizaje.** La programación de un robot autónomo no se puede realizar utilizando una única tarea para un entorno



determinado. Se deben definir por tanto diferentes subtareas capaces de adaptarse a distintos espacios utilizando técnicas de aprendizaje. De esta manera se podrán utilizar tareas que no estaban pensadas inicialmente para una labor determinada.

- **Manejo de la incertidumbre y robustez del sistema.** Un sistema robótico ha de ser capaz de desenvolverse en un mundo incierto proporcionando actuaciones robustas incluso en condiciones adversas. Además, ha de ser capaz de detectar fallos de actuación o percepción y corregirlos de manera satisfactoria.
- **Reactividad y comportamientos consistentes.** El robot tiene que tener en cuenta los límites temporales de las tareas a realizar para hacerlos compatibles con sus metas (incluyendo su propia seguridad). Por otra parte las reacciones del robot a determinados eventos deben ser guiadas por los objetivos de su tarea.

4.2. Modelo Propuesto

El modelo de fusión multinivel robótico propuesto (FSR-BAY: Fusión Sensorial Robótica Bayesiana) se basa en la descomposición y procesado de la información mediante el uso de una serie de unidades de procesamiento Bayesianas. Cada unidad de procesamiento representa una unión de distribuciones de variables probabilistas. Esta unión de distribuciones será la que caracterizará el funcionamiento de una unidad, y vendrá dado por el conocimiento previo del diseñador sobre el problema. Será por tanto el programador el que utilizando el conocimiento disponible sobre la tarea a realizar plantee para cada unidad una determinada estructura que ayude a encontrar los parámetros necesarios para resolver el problema.

Las entradas de las unidades de procesamiento serán variables discretas obtenidas de manera directa o definidas a partir de otras unidades. Las variables discretas se representan con nombres que empiezan por una letra mayúscula. Por definición una variable discreta X es un conjunto de proposiciones x_i donde dichas proposiciones son exclusivas mutuamente y exhaustivas. Las salidas de las unidades de procesamiento estarán formadas por todas las variables de salida especificadas para tal efecto. Cabe mencionar que las salidas de una determinada unidad no son la única información accesible, se pueden realizar multitud de preguntas a la distribución de variables conjunta



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

107

especificada previamente por el programador. Estas preguntas pueden servir como un mecanismo de visualización del funcionamiento de la unidad, como una entrada a otra unidad, o incluso como un mecanismo de refinado de la información tratada por la unidad.

4.2.1. Unidades de Procesamiento

Una unidad de procesamiento u es una descripción que define la siguiente distribución probabilística: $P(I \otimes S \otimes O \otimes |u)$, donde I es una variable de entrada que especifica la información a tratar, S es una variable de estado que representa la situación en la que se encuentra la unidad de procesamiento y O es una variable de salida que especifica la nueva información que se desea generar. Las variables I , S y O no tienen que ser atómicas pudiendo estar formadas por múltiples variables. Por simplicidad, además, se asumirán discretas.

No se limita la descomposición de esta distribución probabilística así como las formas de la misma. De esta manera la descomposición de la variable o variables de entrada I se puede definir utilizando preguntas a otras unidades de procesamiento distintas. Tampoco se especifica ningún tipo de aprendizaje concreto dejando libertad al diseñador del sistema para utilizar el método que considere más oportuno.

La variable de estado S representa la situación de la unidad de procesamiento. La finalidad principal de esta variable es dotar a la unidad de mecanismos de memoria capaces de cambiar el funcionamiento de la unidad. No obstante, en comportamientos reactivos, donde la información de entrada proporciona de manera directa la información de salida, esta variable puede no ser útil y por tanto no se definirá.

En comportamientos más elaborados el uso de una o varias variables de estado es fundamental. Estas variables describirán la situación actual de la unidad y formarán parte del proceso de transformación de las variables de entrada a las variables de salida. El estado de una unidad podrá cambiar cada vez que se le proporcione a esta nueva información de entrada.

Tal y como se ha especificado previamente u está formada por una descripción sobre un conjunto de variables. Es por tanto sencillo realizar cualquier pregunta a esta distribución. De esta manera tanto S como todo el conjunto de preguntas posibles a la unidad u forman la situación de una unidad y por tanto pueden ser utilizados como descriptores de la misma.

Una unidad de procesamiento se considera funcional cuando responde de



manera informativa (su entropía es lo suficientemente lejana a su máximo) a las siguientes preguntas aplicables a su distribución:

- Cuando se trabaje con unidades donde la utilización de un estado S no sea pertinente se requiere que como mínimo se responda de manera informativa a la pregunta $P(O|I)$ de manera que se haya especificado para cada entrada la salida que generará. De esta manera se asegura que para todos los valores de entrada se obtendrá una salida como resultado del procesamiento.
- Por el contrario, en una unidad con una o más variables de estado se requiere un número de preguntas proporcional al número de estados que disponga dicha unidad:

$$\begin{aligned}
 S &= \{S_0, S_1, \dots, S_n\}, \\
 A_i &= \{a_{1,i}, a_{2,i}\}; a_{1,i} \subseteq O, a_{2,i} \subseteq S; \\
 &P(S_0|I), P(S_1|S_0 \otimes A_0), P(S_2|S_1 \otimes A_1), \dots, P(S_n|S_{n-1} \otimes A_{n-1}), \\
 &P(O|S_n \otimes a_{2,n})
 \end{aligned}$$

Inicialmente, partiendo de las variables de entrada, se obtiene la probabilidad de que se dé un primer estado S_0 . Utilizando este estado se han de responder de manera informativa todas las preguntas necesarias hasta llegar a la distribución que permita obtener las variables de salida. En alguno de estos pasos intermedios puede ser de utilidad el contar con información de algún estado previo o incluso de la salida de la unidad para proporcionar más información. Una unidad Bayesiana, para considerarse útil, ha de responder de manera informativa a todas las preguntas definidas previamente.

De esta manera la respuesta a una de estas preguntas podrá servir para realizar una primera evaluación cualitativa del funcionamiento de una unidad de procesamiento.



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

109

$$\left. \begin{array}{l} \text{UB} \\ \left\{ \begin{array}{l} \text{Descripción} \\ \left\{ \begin{array}{l} \text{Especificación} \\ \left\{ \begin{array}{l} \text{Variables:} \\ I : \text{Información de entrada} \\ S : \text{Estado interno de la unidad} \\ O : \text{Información de salida} \\ \text{Descomposición: Cualquiera} \\ \text{Formas paramétricas: Cualquiera} \\ \text{Identificación: Cualquiera} \end{array} \right. \\ \text{Preguntas: Cualquiera} \\ \text{Se requieren informativas:} \\ \text{Si } S = \emptyset \\ \{P(O|I)\} \\ \text{Sino} \\ \left\{ \begin{array}{l} S = \{S_0, S_1, \dots, S_n\}, \\ A_i = \{a_{1,i}, a_{2,i}\}; a_{1,i} \subseteq O, a_{2,i} \subseteq S; \\ P(S_0|I), (P(S_1|S_0 \otimes A_0), P(S_2|S_1 \otimes A_1), \dots, P(S_n|S_{n-1} \otimes A_{n-1})), \\ P(O|S_n \otimes a_{2,n}) \end{array} \right. \end{array} \right\} \end{array} \right\} \end{array} \right. \quad (4.1)$$

4.2.2. Conexión Bayesiana de Unidades de Procesamiento

Tal y como se ha especificado previamente, dada una unidad u es posible realizar cualquier pregunta sobre la unión de distribuciones que la forma; como se define en la sección 3.1.3 esto es posible siempre que se conozca la unión de distribuciones conjunta.

Por otra parte, la descomposición de una unidad de procesamiento tampoco está limitada. Es el diseñador del sistema el encargado de especificar las formas de cada término de dicha distribución. De esta manera uno de estos términos puede ser una pregunta a otra unidad de procesamiento distinta de la actual, procediendo a unir una unidad u con otras unidades cercanas a sus capacidades o necesidades de procesado de la información.

La conexión entre las unidades determinará por tanto el flujo de la información y el funcionamiento conjunto del sistema. Por ejemplo una conexión



(x, y) si $x < 4$	\rightarrow	$(4, y)$
(x, y) si $y < 3$	\rightarrow	$(x, 3)$
(x, y) si $x > 0$	\rightarrow	$(x - d, y)$
(x, y) si $y > 0$	\rightarrow	$(x, y - d)$
(x, y) si $x > 0$	\rightarrow	$(0, y)$
(x, y) si $y > 0$	\rightarrow	$(x, 0)$
(x, y) si $x + y \geq 4 \wedge y > 0$	\rightarrow	$(4, y - (4 - x))$
(x, y) si $x + y \geq 3 \wedge x > 0$	\rightarrow	$(x - (3 - y), 3)$
(x, y) si $x + y \leq 4 \wedge y > 0$	\rightarrow	$(x + y, 0)$
(x, y) si $x + y \leq 3 \wedge x > 0$	\rightarrow	$(0, x + y)$

Figura 4.1: R : posibles acciones a realizar.

secuencial puede proporcionar procesos más cercanos a los sistemas jerárquicos, una conexión en estrella sistemas con una base de procesamiento centrado en la información. . .

Conexión secuencial de unidades de procesamiento

A continuación vamos a ver un ejemplo muy simple donde se definen dos unidades de procesamiento que colaborarán para resolver un problema clásico de inteligencia artificial, las jarras de agua. Aunque este problema se especifica de manera satisfactoria utilizando sistemas de producción puede ser instructivo ver como se podría resolver utilizando el modelo aquí presentado.

El problema plantea lo siguiente: se tienen dos jarras de agua, una de tres litros y otra de cuatro, las cuales no disponen de marcas de medición. Las jarras se pueden llenar con una bomba de agua y también se pueden vaciar en el suelo. ¿Cómo se puede lograr tener exactamente dos litros de agua en la jarra de cuatro litros?

La fase inicial de nuestra aproximación se encarga de especificar las variables necesarias para la resolución del problema. A priori necesitaremos una variable $S_{jarra} = (x, y)$ que especificará el número de litros que contiene la jarra 1 (x) y la jarra 2 (y). Además sería conveniente utilizar una variable que determine si se ha llegado a la solución del problema *final* y otra que contenga las posibles acciones a realizar R (ver tabla 4.1).

Una vez decididas las variables necesarias se especifican dos unidades de procesamiento. Una primera unidad será la encargada de seleccionar la regla



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

111

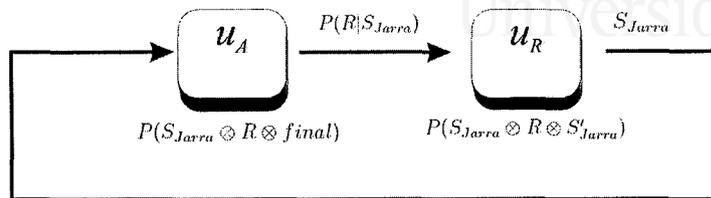


Figura 4.2: Conexión secuencial de UP

a aplicar dado un estado determinado:

$$\begin{aligned}
 U_A : (I, S, O) &= (S_{Jarra}, \emptyset, final \otimes R) \\
 P(S_{Jarra} \otimes R \otimes final) &= P(S_{Jarra}) \times P(R|S_{Jarra}) \times P(final|S_{Jarra} \otimes R) = \\
 &= P(S_{Jarra}) \times P(R|S_{Jarra}) \times P(final|S_{Jarra})
 \end{aligned}$$

La segunda se encargará de obtener, dado un estado y la regla a aplicar, el nuevo estado generado

$$\begin{aligned}
 U_R : (I, S, O) &= (S_{Jarra} \otimes R, \emptyset, S'_{Jarra}) \\
 P(S_{Jarra} \otimes R \otimes S'_{Jarra}) &= P(S_{Jarra}) \times P(R|S_{Jarra}) \times P(S'_{Jarra}|S_{Jarra} \otimes R)
 \end{aligned}$$

El término $P(S_{Jarra})$ es considerado a priori uniforme ya que todos los estados se consideran equiprobables. $P(final|S_{Jarra})$ viene especificada por el enunciado del problema. $P(R|S_{Jarra})$ se puede especificar en forma de tabla, de manera que dado un estado se considere que reglas se pueden aplicar e incluso cuales son las más probables en ser seleccionadas. Como este término también figura en la segunda unidad puede ser obtenido directamente de la primera sin necesidad de recalcular. Además, de esta manera, la segunda unidad realiza la ejecución de la regla dependiendo de la selección de la regla por parte de la primera unidad. Por último $P(S'_{Jarra}|S_{Jarra} \otimes R)$ se representará también como una tabla, donde dada una regla y un estado se especificará el nuevo estado que se desea alcanzar.

Se puede comprobar de manera sencilla que las unidades definidas previamente responden de manera informativa a las preguntas base de las unidades Bayesianas. Para la primera unidad, al no contener estado, se ha de responder a la pregunta $P(final \otimes R|S_{Jarra})$. Como se ha visto previamente $final$ y R son variables independientes de manera que $P(final \otimes R|S_{Jarra}) =$



$P(\text{final}|S_{Jarra}) \times P(R|S_{Jarra})$. Tanto el primer término como el segundo están especificados con anterioridad. Para la segunda unidad hay que responder de manera informativa a la pregunta $P(S'_{Jarra}|S_{Jarra} \otimes R)$ que también está especificada previamente.

La selección de las reglas aplicables para una posterior ejecución se puede realizar de manera aleatoria extrayendo un elemento al azar de la probabilidad $P(R|S_{Jarra})$. No obstante, para simplificar la nomenclatura y la resolución del problema se deja al programador la responsabilidad de proporcionar mecanismos para almacenar la secuencia de reglas aplicadas así como para asegurar que no se repiten estados al aplicar una determinada regla.

La unidad U_R obtendrá un nuevo estado a partir de la aplicación de R . Este estado será transmitido a la unidad U_A hasta que se determine la finalización del algoritmo mediante la pregunta $P(\text{final}|S_{Jarra})$.

Operadores de fusión

La conexión de unidades de procesamiento puede requerir un flujo de información continuo que ha de ser proporcionado por preguntas a otras unidades. Es posible que en determinadas situaciones o que para algunos tipos de información sea conveniente combinar dos o más unidades de procesamiento para formar una única unidad. La combinación de unidades dependerá del tipo de información y del método de fusión que se desee utilizar. En la siguiente sección se proponen dos operadores para la combinación de unidades de proceso en relación al tipo de configuración utilizada. Los tipos de configuración utilizados se basan en la categorización sensorial, propuesta por Durrant-Whyte (ver 2.1.3).

4.3. Operador de fusión complementaria

Se define una configuración de unidades de proceso como complementaria si las unidades no dependen directamente entre ellas pero por otra parte se pueden combinar para conseguir una imagen más completa de la información que está siendo observada. Un ejemplo podría ser un conjunto de unidades que intenten extraer distinta información de un entorno común (por ejemplo variables de localización) que serán después combinadas para conseguir una descripción más completa del fenómeno observado (un mapa del entorno).



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

113

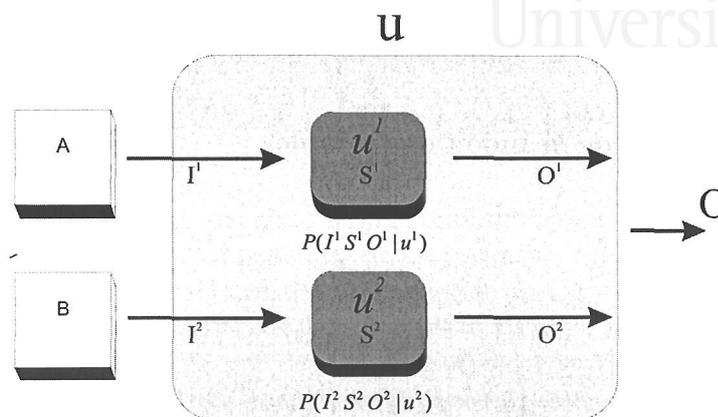


Figura 4.3: Operador de fusión complementaria

Si disponemos de dos unidades u^1 y u^2 , capaces de responder a las preguntas pertinentes se desea obtener una nueva unidad u que aglutine la información de ambas unidades. Como ya se ha comentado anteriormente la información que genera una unidad no queda únicamente limitada a la información de salida O sino que cualquier pregunta a una unidad así como su estado actual S muestran información de la misma. De esta manera, al tratarse de unidades independientes será necesario combinar todas sus variables y utilizarlas como variables de entrada en la nueva unidad. Obviamente el número de estados influirá en como se fusionan dos unidades. Se supondrá que se desean agrupar dos unidades con el mismo número de estados.

Unidades reactivas, $|S| = 0$

Definición de las variables

En el caso de la fusión de unidades reactivas, aquellas unidades que no contienen ningún estado, se definen las variables de la nueva unidad u como:

$$\begin{aligned} I &= I_1 \wedge I_2 \wedge O_1 \wedge O_2, |I| = |O_1| + |O_2| + |I_1| + |I_2| \\ O &= O_1 \cup O_2, |O| = |O_1| + |O_2| - |O_1 \cap O_2| \\ S &= \emptyset, |S| = 0 \end{aligned}$$

Como se ha comentado anteriormente, la variable I contendrá todas las variables de las subunidades a fusionar. La variable de salida de la unidad O



contendrá todas las variables distintas de salida tanto de la primera unidad como de la segunda.

Descomposición de la unión de variables

La descomposición propuesta es la siguiente:

$$\begin{aligned}
 P(I \otimes O) &= P(I_1 \otimes I_2 \otimes O_1 \otimes O_2 \otimes O) = \\
 &= P(I_1) \times P(I_2|I_1) \times P(O_1|I_1 \otimes I_2) \times P(O_2|I_1 \otimes I_2 \otimes O_1) \times \\
 &\quad \times P(O|I_1 \otimes I_2 \otimes O_1 \otimes O_2) = \\
 &= P(I_1) \times P(O_1|I_1) \times P(I_2) \times P(O_2|I_2) \times P(O|O_1 \otimes O_2) =
 \end{aligned}$$

Siendo para el caso general de fusión de n unidades:

$$P(I \otimes O) = \left(\prod_{i=1}^n P(I_i \otimes O_i) \right) \times P(O|O_1 \otimes O_2 \otimes \dots \otimes O_n)$$

De esta manera, suponiendo la descomposición de una unidad como $P(I \otimes O) = P(I) \times P(O|I)$, la fusión complementaria de dos unidades reactivas se define como el producto de sus distribuciones por $P(O|O_1 \otimes O_2)$. Para cada salida O_1 y O_2 la distribución $P(O|O_1 \otimes O_2)$ determina una nueva salida O formada por el dominio de O_1 y O_2 permitiendo la unión del conocimiento proveniente de estas dos unidades. Además esta distribución se puede diseñar de manera sencilla por el programador en forma de tabla, que proporcione para cada salida parcial de las subunidades qué salida corresponde a la unidad fusionada.

Identificación de términos

Todos los términos de la forma $P(I_i \otimes O_i)$ han sido descompuestos e identificados en sus unidades pertinentes. El término $P(O|O_1 \otimes O_2)$ es una tabla que o bien puede ser aprendida o bien especificada previamente por el programador.

Preguntas

Para que la definición de la unión complementaria de ambas unidades sea considerada como unidad ha de responder de manera informativa a la pregunta $P(O|I)$:



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

115

$$P(O|I) = \frac{P(O \otimes I)}{P(I)} = \frac{1}{\Sigma} \times \left(\prod_{i=1}^2 P(I_i \otimes O_i) \right) \times P(O|O_1 \otimes O_2)$$

Podemos considerar que $P(I)$ actúa como un término normalizador por lo que es sustituido por la constante $\frac{1}{\Sigma}$. La pregunta $P(O|I)$ se responde de manera informativa si tanto las distribuciones de ambas subunidades $P(I \otimes O)$ como la distribución $P(O|O_1 \otimes O_2)$ son informativas. Para cada subunidad, $P(I \otimes O)$ es informativa por definición. Por otra parte es el programador el que ha de asegurar que su propuesta de distribución $P(O|O_1 \otimes O_2)$ es informativa.

Unidades con un estado, $|S| = 1$

Definición de las variables

Siguiendo la misma filosofía que en el caso anterior la fusión complementaria requiere que todas las variables de las uniones a fusionar se dispongan como variables de entrada de la nueva unidad. Se definen por tanto las variables de la nueva unidad u como:

$$\begin{aligned} I &= I_1 \wedge I_2 \wedge O_1 \wedge O_2 \wedge S_1 \wedge S_2, \\ |I| &= |O_1| + |O_2| + |I_1| + |I_2| + |S_1| + |S_2| \\ O &= O_1 \cup O_2 \\ |O| &= |O_1| + |O_2| - |O_1 \cap O_2| \\ S &= \{s_1, s_2, \dots, s_n\}, s_i \in (S_1 \cup S_2) \\ |S| &\geq 2 \end{aligned}$$

La variable de estado S ha de ser especificada por el programador y debe contener un número de estados igual o mayor a dos para no perder la representatividad de las subunidades. Se pueden especificar tantos estados como unidades a fusionar de manera que cada estado haga referencia a una unidad determinada. Esto puede ser útil, por ejemplo, en situaciones en las cuales se desee agrupar diversas localizaciones robóticas en un mapa global. No obstante también es posible reducir la expresividad de las unidades descendiendo el número de estados (o aumentarla incrementando su número). Queda por tanto en manos del programador el decidir el número de estados relevantes en la unidad fusionada. No obstante cabe remarcar que todos los



estados definidos, para que sean de utilidad, han de tener relación con alguno o algunos de los estados de las subunidades a ser fusionadas.

Descomposición de la unión de variables

La descomposición propuesta es la siguiente:

$$\begin{aligned} P(I \otimes S \otimes O) &= P(I_1 \otimes I_2 \otimes S_1 \otimes S_2 \otimes O_1 \otimes O_2 \otimes S \otimes O) = \\ &= P(S) \times \left(\prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S) \right) \times P(O | S) \end{aligned}$$

De esta manera la fusión complementaria de dos unidades está formada por el producto de sus distribuciones conocido un estado junto con la probabilidad de que se dé ese estado multiplicado por la probabilidad de que dado ese estado se dé una determinada salida. De esta manera se consigue que todas las subunidades que formen la unidad determinen la salida producida por la unidad.

Identificación de términos

$P(I_i \otimes S_i \otimes O_i | S)$ está definido en sus correspondientes subunidades siempre que $S \subseteq S_i$, considerándose uniforme en otro caso. $P(S)$ se podrá considerar uniforme a priori ya que es probable que no se disponga de información inicial del espacio de estados. $P(O | S)$ se especificará en forma de tabla donde para cada estado el programador especificará que salidas son más probables.

Preguntas

Al contener ambas unidades un estado, se ha de confirmar que la unidad fusionada responda de manera informativa a dos preguntas; $P(S | I)$ y $P(O | S)$. Para la primera pregunta tenemos que:

$$\begin{aligned} P(S | I) &= \frac{1}{\Sigma} \sum_o \left(P(S) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S) \times P(O | S) \right) = \\ &= \frac{1}{\Sigma} P(S) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S) \times \sum_o P(O | S) = \\ &= \frac{1}{\Sigma_1} P(S) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S) = \\ &= \frac{1}{\Sigma_2} \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S) \end{aligned}$$



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

117

De esta manera se observa que la pregunta $P(S|I)$ queda especificada con la unión de distribuciones de $P(I \otimes S \otimes O)$ dado un estado S . Por otra parte, $P(O|S)$ se define como:

$$\begin{aligned}
 P(S|O) &= \frac{1}{\Sigma} \times \sum_I \left(P(S) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i|S) \times P(O|S) \right) = \\
 &= \frac{1}{\Sigma} \times P(S) \times \sum_I \left(\prod_{i=1}^n P(I_i \otimes S_i \otimes O_i|S) \right) \times P(O|S) \\
 &= \frac{1}{\Sigma} \times P(S) \times P(O|S) = \\
 &= \frac{1}{\Sigma_2} \times P(O|S)
 \end{aligned}$$

Donde $P(O|S)$ se ha definido previamente en la descomposición de la unión de probabilidades.

Unidades con más de un estado, $|S| > 1$

Con unidades de más de un estado es difícil especificar una única descomposición que determine el modelo de fusión más apto. No obstante se proporciona una descomposición de la unidad fusionada genérica para n estados. Un caso particular de esta descomposición, para $n = 2$ ha sido especificado y utilizado previamente para la navegación y localización robótica en [DBM03b], [DBM04a].

Definición de las variables

En este caso dispondremos como variables de entrada a todas las variables de las subunidades a fusionar:

$$\begin{aligned}
 I &= I_1 \wedge I_2 \wedge O_1 \wedge O_2 \wedge S_1 \wedge S_2, \\
 |I| &= |O_1| + |O_2| + |I_1| + |I_2| + |S_1| + |S_2| \\
 S_1 &= \{a_1, a_2, \dots, a_m\} ; S_2 = \{b_1, b_2, \dots, b_m\} \\
 |S_1| &= |S_2| = m \\
 O &= O_1 \cup O_2 \\
 |O| &= |O_1| + |O_2| - |O_1 \cap O_2| \\
 S &= \{S_a, S_b\} \\
 S_a &\subseteq S_1, S_b \subseteq S_2 \\
 |S_a| + |S_b| &\geq 4
 \end{aligned}$$



Descomposición de la unión de variables

A continuación se propone una descomposición genérica para la fusión de dos unidades de información con m estados. Obviamente esta no es la única descomposición posible pero sí es una de las que se han mostrado más prometedoras a la hora de realizar tareas de fusión.

$$\begin{aligned}
 P(I \otimes S \otimes O) &= P(I \otimes S^1 \otimes \dots \otimes S^m \otimes O) = \\
 &P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1) \times \prod_{i=1}^n P(S^2 | I_i \otimes S_i \otimes O_i \otimes S^1) \times \\
 &\times \dots \times \prod_{i=1}^n P(S^m | I_i \otimes S_i \otimes O_i \otimes S^{m-1} \otimes S^{m-2} \otimes \dots \otimes S^1) \times \\
 &\times P(O | S^1 \otimes S^2 \otimes \dots \otimes S^m)
 \end{aligned}$$

No obstante, si consideramos que únicamente el primer estado depende de las variables de entrada entonces:

$$\begin{aligned}
 P(I \otimes S \otimes O) &= P(I \otimes S^1 \otimes \dots \otimes S^m \otimes O) = \\
 &P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1) \times P(S^2 | S^1) \times P(S^3 | S^1 \otimes S^2) \times \\
 &\times \dots \times P(S^m | S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times P(O | S^1 \otimes S^2 \otimes \dots \otimes S^m)
 \end{aligned}$$

Identificación de términos

$P(S^1)$ se podrá considerar a priori uniforme ya que es probable que no se disponga de información inicial del espacio de estados. $P(S^2 | S^1) \times P(S^3 | S^1 \otimes S^2) \times \dots \times P(S^m | S^1 \otimes S^2 \otimes \dots \otimes S^{m-1})$ determinan la memoria de la unidad que será o bien especificada por el programador o bien aprendida. Por otra parte $\prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1)$ queda definido por las respuestas de las subunidades Bayesianas al estado inicial. Por último, la distribución $P(O | S^1 \otimes S^2 \otimes \dots \otimes S^m)$ que determina la salida de la unidad dependiendo del estado de su memoria, deberá ser especificada por el programador. De esta manera se consigue que todos los estados de ambas unidades contribuyan en la decisión de cual será la nueva salida de la unidad fusionada.

Preguntas

Según la definición de una unidad de proceso Bayesiana, para una unidad de n estados se han de responder de manera informativa n preguntas relativas



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA 119

a la distribución de probabilidades conjunta. A continuación se planteará la resolución de los tres tipos de preguntas diferentes que será necesario identificar para comprobar que la unión de las unidades también es una unidad Bayesiana.

- **Obtención de la información.** Inicialmente, a partir de la información de entrada I se ha de obtener el primer estado de la unidad S^1 . De esta manera, se ha de responder de manera informativa a la siguiente pregunta $P(S^1|I)$:

$$\begin{aligned}
 P(S^1|I) &= \sum_{O, S^2, \dots, S^m} \left(\begin{array}{l} P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1) \times \\ P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \\ \times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times \\ P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m) \end{array} \right) = \\
 &= P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1) \times \\
 &\times \sum_{S^2, \dots, S^m} \left(\begin{array}{l} P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \\ \times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \end{array} \right) \times \\
 &\times \sum_O P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m) = \\
 &= \frac{1}{\Sigma} \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1)
 \end{aligned} \tag{4.2}$$

Donde $\frac{1}{\Sigma}$ engloba el término $P(S^1)$. De esta manera dada una entrada para determinar el valor del estado inicial hay que determinar cual es la respuesta de las subunidades a este estado.

- **Secuencia de estados.** Por otra parte se ha de asegurar, para cada secuencia de estados establecida hasta llegar a la información de salida, la información de las preguntas pertinentes. Es decir se ha de comprobar que las siguientes distribuciones son informativas: $P(S_1|S_0 \otimes A_0) \wedge P(S_2|S_1 \otimes A_1), \dots, P(S_n|S_{n-1} \otimes A_{n-1})$, donde $A_i \subseteq \{S, O\}$.

$$\begin{aligned}
P(S_{i+1}|S_i \otimes O_i) &= \sum_I \left(\begin{array}{l} P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i|S^1) \times \\ P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \\ \times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times \\ P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m) \end{array} \right) = \\
&= P(S^1) \times \sum_I \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i|S^1) \times P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \\
&\times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m) = \\
&= \frac{1}{\Sigma} \times P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times \\
&\times P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m)
\end{aligned} \tag{4.3}$$

De esta manera la obtención de un estado no depende únicamente de las distribuciones entre estados sino también de lo que contribuye un estado en la generación de una determinada salida.

- **Obtención de la información de salida.** Por último cabe determinar la no uniformidad de la pregunta $P(O|S)$ que especifica la salida de la unidad.

$$\begin{aligned}
P(O|S) &= \sum_I \left(\begin{array}{l} P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i|S^1) \times \\ P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \\ \times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times \\ P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m) \end{array} \right) = \\
&= P(S^1) \times \sum_I \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i|S^1) \times P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \\
&\times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m) = \\
&= \frac{1}{\Sigma} \times P(S^2|S^1) \times P(S^3|S^1 \otimes S^2) \times \dots \times P(S^m|S^1 \otimes S^2 \otimes \dots \otimes S^{m-1}) \times \\
&\times P(O|S^1 \otimes S^2 \otimes \dots \otimes S^m)
\end{aligned} \tag{4.4}$$

Como era de esperar, al igual que en el caso anterior, la obtención de la información de salida no depende únicamente de lo que contribuye un estado en la generación de una determinada salida sino también de las distribuciones entre estados que parten desde la información de entrada.



4.4. Operador de fusión competitiva heterogénea

Se define una configuración de unidades de proceso como competitiva o redundante si cada unidad obtiene medidas independientes de una misma propiedad. Se distinguen dos configuraciones competitivas [VG99]: la fusión de datos provenientes de diferentes sensores o las medidas obtenidas por un único sensor en diferentes instantes de tiempo. Las configuraciones competitivas proporcionan robustez al sistema.

De la misma manera que en el apartado anterior, si disponemos de dos unidades u^1 y u^2 , capaces de responder pertinentemente a sus preguntas asociadas se desea obtener una nueva unidad u que aglutine la información de ambas unidades. El número de estados de las unidades influirá en como se fusionan dos unidades. A priori se supondrá que se desean agrupar dos unidades con el mismo número de estados.

En esta sección se presenta un operador binario capaz de aglutinar dos unidades fusionando la información de ambas. En una primera aproximación se supondrá que no somos capaces de combinar a la vez las salidas de u^1 y u^2 , en este caso hablaremos de fusión heterogénea de unidades para un número de estados dado. En la siguiente sección se considerará el caso de unidades homogéneas en las que la salida de ambas unidades se pueda combinar en un mismo instante de manera sencilla.

Fusión competitiva, unidades reactivas, $|S| = 0$

En el caso de la fusión de unidades reactivas, aquellas unidades que no contienen ningún estado, se definen las variables de la nueva unidad u como:

$$\begin{aligned} I &= I_1 \wedge I_2, |I| = |I_1| + |I_2| \\ O &= O_1 \cup O_2, |O| = |O_1| + |O_2| - |O_1 \cap O_2| \\ S &= \emptyset, |S| = 0 \end{aligned}$$

De esta manera, la variable de entrada I estará formada por las entradas de las subunidades. Como no sabemos cómo combinar las salidas de las subunidades al fusionarlas, la salida estará formada por todas las posibles salidas de estas.

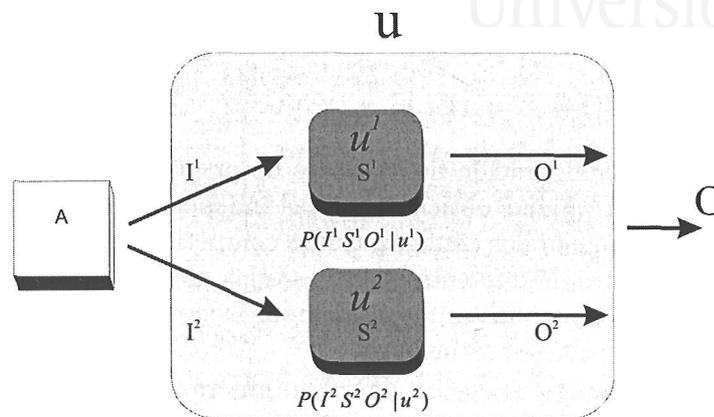


Figura 4.4: Operador de fusión complementaria

Descomposición de la unión de variables

Se propone la siguiente descomposición:

$$\begin{aligned}
 P(I \otimes O) &= P(I_1 \otimes I_2 \otimes O) = \\
 &= P(O) \times P(I_1|O) \times P(I_2|O \otimes I_1) = \\
 &= P(O) \times P(I_1|O) \times P(I_2|O)
 \end{aligned}$$

Donde la segunda igualdad se deduce de la independencia condicional entre I_1 y I_2 .

Identificación de términos

La probabilidad de una determinada salida $P(O)$ se considera a priori uniforme. Por otra parte ambas distribuciones $P(I_1|O)$ y $P(I_2|O)$ han de ser especificadas por el programador, de manera que dada una salida se determine qué entradas pudieron generarlas. Normalmente se utilizará una tabla que especifique las relaciones entre la salida y las entradas.

Preguntas

Para que la definición de la nueva unidad sea informativa se debe responder a la pregunta $P(O|I)$:

$$P(O|I) = \frac{P(O \otimes I)}{P(I)} = \frac{1}{\Sigma} \times P(O) \times P(I_1|O) \times P(I_2|O)$$



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

123

$P(I)$ actúa como un término normalizador por lo que se substituye por la constante $\frac{1}{Z}$. La pregunta $P(O|I)$ se responde correctamente conociendo la distribución de probabilidades conjunta de $P(I \otimes O)$.

Fusión competitiva, unidades con un estado, $|S| = 1$

La fusión competitiva parte de la base de que se desea trabajar con información que describe una misma característica. De esta manera la entrada a la nueva unidad Bayesiana estará formada por las entradas de las subunidades.

$$\begin{aligned} I &= I_1 \wedge I_2, |I| = |I_1| + |I_2| \\ S &= S_1 \wedge S_2, |S| = |S_1| + |S_2| \\ O &= O_1 \cup O_2, |O| = |O_1| + |O_2| - |O_1 \cap O_2| \end{aligned}$$

De la misma manera los estados de las subunidades formarán el nuevo estado de la unidad principal.

Descomposición de la unión de variables

En el caso de subunidades con un estado el número de descomposiciones posibles aumenta con creces. A continuación se describe una de las descomposiciones posibles que han proporcionado mejores resultados.

$$\begin{aligned} P(I \otimes S \otimes O) &= P(I_1 \otimes I_2 \otimes S_1 \otimes S_2 \otimes O) = \\ &= P(S_1) \times P(S_2) \times P(I_1|S_1 \otimes S_2) \times P(I_2|S_1 \otimes S_2 \otimes I_1) \times P(O|S_1 \otimes S_2) = \\ &= P(S_1) \times P(S_2) \times P(I_1|S_1) \times P(I_2|S_2) \times P(O|S_1 \otimes S_2) \end{aligned}$$

La segunda igualdad se deduce al considerar que las informaciones obtenidas por las unidades son independientes entre sí.

Identificación de términos

$P(S_1)$ y $P(S_2)$ son términos uniformes ya que a priori no sabemos la probabilidad de que se dé un estado determinado. Tanto $P(I_1|S_1)$ como $P(I_2|S_2)$ pueden considerarse como modelos del sensor. Estos modelos se pueden obtener a partir de especificaciones del fabricante o de relaciones de la información establecidas a priori. Por último la distribución $P(O|S_1 \otimes S_2)$ tiene forma de tabla y ha de ser especificada por el programador o aprendida.



Preguntas

Para que la unidad resultante de la fusión sea una unidad Bayesiana se ha de contestar de manera informativa a las siguientes preguntas: $P(S|I)$ y $P(O|S)$.

$$\begin{aligned}
 P(S|I) &= \sum_O (P(S_1) \times P(S_2) \times P(I_1|S_1) \times P(I_2|S_2) \times P(O|S_1 \otimes S_2)) = \\
 &= P(S_1) \times P(S_2) \times P(I_1|S_1) \times P(I_2|S_2) \times \sum_O P(O|S_1 \otimes S_2) \\
 &= \frac{1}{\sum_I} \times P(S_1) \times P(S_2) \times P(I_1|S_1) \times P(I_2|S_2) \\
 &= \frac{1}{\sum_I} \times P(I_1|S_1) \times P(I_2|S_2)
 \end{aligned}$$

De esta manera se define $P(S|I)$ como el producto de las distribuciones $P(I_1|S_1)$ y $P(I_2|S_2)$ pertenecientes a las dos subunidades fusionadas.

$$\begin{aligned}
 P(O|S) &= \sum_I (P(S_1) \times P(S_2) \times P(I_1|S_1) \times P(I_2|S_2) \times P(O|S_1 \otimes S_2)) = \\
 &= \frac{1}{\sum_I} \times P(O|S_1 \otimes S_2)
 \end{aligned}$$

La probabilidad $P(O|S_1 \otimes S_2)$ definida previamente será determinada por el valor de salida para cada combinación de estados.

Fusión competitiva, unidades con dos estados, $|S| = 2$

En el caso de unidades con dos estados las descomposiciones posibles son mayores y por tanto dependen más concretamente del problema a tratar. No obstante, puede resultar útil identificar los estados de manera temporal, representando el presente/pasado o presente/futuro. En nuestro caso concreto una unidad contendrá las siguientes variables:

$$\begin{aligned}
 I &= I_1 \wedge I_2, |I| = |I_1| + |I_2| \\
 S &= S_1 \wedge S_2 \wedge S_1^1 \wedge S_2^1, |S| = |S_1| + |S_2| + |S_1^1| + |S_2^1| \\
 O &= O_1 \cup O_2, |O| = |O_1| + |O_2| - |O_1 \cap O_2|
 \end{aligned}$$

Descomposición de la unión de variables

A continuación se plantean tres descomposiciones distintas de la unión de variables. Como se ha comentado anteriormente, en la fusión competitiva la descomposición depende de manera directa del problema y de la interrelación



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

125

de las unidades. No obstante estas tres descomposiciones cubren tendencias generales sobre las opciones de descomposición disponibles.

$$\begin{aligned}
 P(I \otimes S \otimes O) &= P(I_1 \otimes I_2 \otimes S_1 \otimes S_2 \otimes S_1^1 \otimes S_2^1 \otimes O) = \\
 &= P(I_1) \times P(I_2) \times P(S_1) \times P(S_2) \times P(S_1^1|I_1 \otimes S_1) \times \\
 &\quad \times P(S_2^1|I_2 \otimes S_2) \times P(O|S_1^1 \otimes S_2^1)
 \end{aligned} \tag{4.5}$$

En esta descomposición podemos observar como el camino hacia la información de salida depende de la interacción de los dos estados de las subunidades. Por una parte, dado el estado actual y la información de entrada se desea obtener cuál será el estado al que se desea llegar. Ambos estados futuros S_1^1 y S_2^1 han de ser combinados para obtener nueva información.

$$\begin{aligned}
 P(I \otimes S \otimes O) &= P(I_1 \otimes I_2 \otimes S_1 \otimes S_2 \otimes S_1^1 \otimes S_2^1 \otimes O) = \\
 &= P(S_1) \times P(S_2) \times P(I_1|S_1) \times P(I_2|S_2) \times P(O) \times \\
 &\quad \times P(S_1^1|O \otimes S_1) \times P(S_2^1|O \otimes S_2)
 \end{aligned} \tag{4.6}$$

Por otra parte, posible descomposición se puede observar en la ecuación 4.6. En este caso se utiliza el estado actual y la salida deseada para determinar cuál debe de ser el estado futuro. Además, el estado actual se determina utilizando la información de entrada disponible. Una descomposición similar ha sido utilizada, junto con el formalismo de los mapas Bayesianos, de manera satisfactoria para la generación de mapas y conductas en robots autónomos [DBM04a], [DBM04b], [DBM03b].

$$\begin{aligned}
 P(I \otimes S \otimes O) &= P(I_1 \otimes I_2 \otimes S_1 \otimes S_2 \otimes S_1^1 \otimes S_2^1 \otimes O) = \\
 &= P(O) \times P(I_1|O) \times P(I_2|O) \times P(S_1|I_1) \times P(S_2|I_2) \times \\
 &\quad \times P(S_1^1|S_1 \otimes O) \times P(S_2^1|S_2 \otimes O)
 \end{aligned} \tag{4.7}$$

Por último se presenta la descomposición 4.7. Dada una salida se puede determinar qué entrada es la que la generó con mayor probabilidad. De la misma manera dada una entrada se puede determinar cuál es el mejor estado que la representa. Por último dado el estado actual y la salida requerida se desea alcanzar un nuevo estado.

Identificación de términos y preguntas

Tanto la identificación de términos como la comprobación de que las preguntas relativas a una unidad sean informativas dependen directamen-



te de la descomposición utilizada. Debido a la gran diversidad de descomposiciones posibles no se ha determinado en este apartado ninguna identificación de términos o respuesta a las preguntas de la unidad, dejando esta demostración para cuando se utilice en concreto alguna de estas descomposiciones. No obstante es sencillo determinar que, por ejemplo, a partir de las tres descomposiciones anteriores se genera una unidad Bayesiana que responde de manera informativa a sus preguntas guía teniendo identificados todos sus términos. Estas preguntas se derivan de la definición de una unidad Bayesiana y son para el caso de dos estados las siguientes: $P(S_0|I)$, $P(S_1|S_0 \otimes A_0)$, $P(S_2|S_1 \otimes A_1)$, $P(O|S_2 \otimes S)$.

Fusión competitiva heterogénea, unidades con n estados, $|S| = n$

Para el caso general de fusión de unidades con n estados nos encontramos con que el número de descomposiciones posibles es muy grande (del orden de $n!$) si no tenemos en cuenta ninguna simplificación relativa a la descomposición. No obstante, aun teniendo en cuenta dichas simplificaciones continuamos trabajando sobre un conjunto de descomposiciones excesivamente grande.

Es el programador el que ha de elegir una descomposición y comprobar el funcionamiento de la misma para solucionar un problema concreto. Como se comentó anteriormente la descomposición de las variables refleja el conocimiento previo que se tiene sobre la tarea a solucionar. Puede ser conveniente que el programador utilice como guía alguna de las tres descomposiciones presentadas en el apartado anterior. En estas descomposiciones se asocia una semántica temporal a los estados que permiten de manera lógica describir las relaciones entre las variables de la unidad.

No obstante buscar descomposiciones de las variables de entrada con más de dos estados, que se adapten a una determinada tarea, así como proporcionar un método de selección de dichas descomposiciones para un problema dado son algunas de las líneas de investigación futuras que se proponen en esta tesis.



4.5. Operador de fusión competitiva homogénea

Como se ha comentado anteriormente una configuración de unidades de proceso se consideran competitivas si cada unidad obtiene medidas independientes de una misma propiedad. A la hora de fusionar dos unidades competitivas de manera *homogénea* supondremos que somos capaces de combinar sus salidas para generar una nueva. De esta manera se supone que se conocen las propiedades de la información de salida que permiten unirlos o bien que existe un sistema capaz de arbitrar qué salida se utilizará en la unidad.

Fusión competitiva homogénea, unidades reactivas, $|S| = 0$

En las unidades competitivas homogéneas podemos suponer a priori que ambas subunidades son independientes entre sí ya que aunque observen una misma propiedad la información de entrada puede ser distinta. De la misma manera su salida se puede combinar entre sí sin alterar el resultado de la unidad.

Definición de las variables

Se propone la siguiente descomposición de variables:

$$\begin{aligned} I &= I_1 \wedge I_2, |I| = |I_1| + |I_2| \\ O &= O_1 \wedge O_2, |O| = |O_1| + |O_2| \\ S &= \emptyset, |S| = 0 \end{aligned}$$

Descomposición de la unión de variables

Suponiendo independencia condicional entre ambas subunidades se propone la siguiente descomposición:

$$P(I \otimes O) = P(I_1 \otimes I_2 \otimes O_1 \otimes O_2) = P(I_1 \otimes O_1) \times P(I_2 \otimes O_2)$$

Identificación de términos y preguntas

Ambos términos, $P(I_1 \otimes O_1)$ y $\times P(I_2 \otimes O_2)$ se responden preguntando a las subunidades correspondientes. De la misma manera la unidad resultante del



proceso de fusión es informativa siempre y cuando las dos unidades también lo sean.

Fusión competitiva homogénea, unidades con n estados, $|S| = n$

De la misma manera, para el caso en el que dispongamos de n estados, al tratarse de unidades independientes el resultado será parecido al caso anterior.

Definición de las variables

Se propone la siguiente descomposición de variables:

$$\begin{aligned} I &= I_1 \wedge I_2, |I| = |I_1| + |I_2| \\ O &= O_1 \wedge O_2, |O| = |O_1| + |O_2| \\ S &= S_1 \wedge S_2, |S| = |S_1| + |S_2| = 2n \end{aligned}$$

Descomposición de la unión de variables

Suponiendo independencia condicional entre ambas subunidades se propone la siguiente descomposición:

$$\begin{aligned} P(I \otimes S \otimes O) &= P(I_1 \otimes I_2 \otimes S_1^1 \otimes S_1^2 \otimes \dots \otimes S_1^n \otimes S_2^1 \otimes S_2^2 \otimes \dots \otimes S_2^n \otimes O_1 \otimes O_2) = \\ &= \prod_i P(I_i \otimes S_i^1 \otimes S_i^2 \otimes \dots \otimes S_i^n \otimes O_i) \end{aligned}$$

Identificación de términos y preguntas

Al tratarse de unidades independientes los términos de la unidad fusionada dependen directamente de las subunidades que la forman. Por otra parte las preguntas que ha de responder una unidad de manera informativa se han de resolver de manera independiente por cada unidad. De esta manera si las subunidades responden de manera informativa a las preguntas guía también responderá de manera informativa la unidad fusionada.



CAPÍTULO 4. FSR-BAY: MODELO PROBABILÍSTICO PARA LA FUSIÓN SENSORIAL ROBÓTICA

129

4.6. Sobre la independencia condicional

Tanto en el operador de fusión competitiva como en el operador de fusión complementaria se han realizado algunas suposiciones de independencia condicional en el proceso de descomposición de variables. Aunque las suposiciones de independencia condicional se han propuesto de manera genérica puede ser que en alguna tarea concreta sea procedente revisar dichas independencias, enriqueciendo por ejemplo una unidad mediante experimentación.

Un proceso de adquisición de datos anulando algunas de las independencias condicionales realizadas en la descomposición genérica puede llevar a modelos más complejos que por otra parte den más opción a procesos de aprendizaje.

En el capítulo 6 se verá cómo revisando la independencia condicional del operador de fusión competitiva seremos capaces de *enseñar* al robot a utilizar otro tipo de sensores sobre los cuales no tiene información a priori sobre su funcionamiento.

4.7. Resumen

Una arquitectura de fusión orientada a la robótica es deseable que disponga de herramientas para tratar la incertidumbre y la no completitud de los datos con los que trabaja. En el caso de que se utilice una representación interna del mundo, ésta debería basarse y utilizar los mecanismos para trabajar con este tipo de datos. Además cabe resaltar que tanto la incertidumbre como la no completitud de la información así como las capacidades de aprendizaje son temas que han sido tratados de manera secundaria por la mayoría de las arquitecturas de fusión actuales [ACF⁺98].

En este capítulo se ha aportado un modelo de fusión robótico basado en la descomposición y el procesamiento de la información mediante el uso de unidades de procesamiento Bayesianas. Cada unidad de procesamiento representa una unión de distribuciones de variables probabilísticas. Esta unión de distribuciones será la que caracterizará el funcionamiento de una unidad.

Para la conexión de unidades de procesamiento Bayesiano se puede necesitar un flujo de información continuo. Este flujo ha de proporcionarse mediante la realización de preguntas a otras unidades. No obstante, en determinadas situaciones o para algunos tipos de información será conveniente combinar dos o más unidades de procesamiento para formar una única unidad.



Obviamente la combinación de unidades dependerá del tipo de información y del método de fusión a utilizar. Se han proporcionado dos operadores de combinación de unidades, el operador de fusión complementaria y el operador de fusión competitiva, relacionados con el tipo de configuración sensorial utilizada. El operador de fusión complementaria fusiona unidades que no dependen directamente entre ellas pero se pueden combinar para conseguir una imagen más completa del fenómeno observado. El operador de fusión competitiva fusiona unidades que obtienen medidas independientes de una misma propiedad.

El modelo FSR-BAY engloba tanto a las unidades Bayesianas como a los operadores de fusión. Este modelo tiene cuatro ventajas principales respecto a los sistemas de fusión clásicos. Por una parte utiliza una base rigurosa fundamentada en la inferencia y aprendizaje Bayesiano. Además este modelo es capaz de trabajar con la incertidumbre y con la no completitud de la información de manera explícita. Provee también mecanismos de aprendizaje que lo hacen más versátil y adaptable. Por último cabe remarcar que el uso de descripciones probabilísticas como información básica se traduce en una representatividad uniforme para la fusión tanto de bajo como de alto nivel.

En el siguiente capítulo se aportará un ejemplo de fusión complementaria, utilizando el modelo FSR-BAY, para la localización de un robot autónomo.



Universitat d'Alacant
University of Alicante

“Aquella teoria que no encuentre aplicaci3n pr3ctica en la vida es una acrobacia del pensamiento”

SWAMI VIVEKANANDA

Capítulo 5

Caso de estudio: Fusión Bayesiana Complementaria

Como se ha comentado anteriormente cuando disponemos de informaci3n proveniente de varios sensores podemos fusionarla consiguiendo: un aumento de la robustez de los datos, extender la cobertura espacial y temporal, incrementar la confianza, reducir la ambigüedad y la incertidumbre, más robustez contra interferencias y mejorar la resoluci3n. Las Unidades Bayesianas son un modelo de fusión, basado en la programaci3n Bayesiana, orientado al área de la rob3tica móvil. Una unidad Bayesiana permite definir cómo se procesa la informaci3n en una determinada tarea. Agrupando unidades Bayesianas mediante operadores de fusión se consigue establecer una nueva unidad que fusiona a las anteriores.

A continuaci3n se aportará una aplicaci3n pr3ctica del operador de fusión complementaria. Fusionaremos informaci3n proveniente de distintos sensores del mismo tipo. Para el proceso de fusión se proporcionará una especificaci3n del operador de combinaci3n correspondiente. También se obtendrán los distintos submapas que describen el entorno del robot. Se demostrará que un mapa Bayesiano es un tipo especial de unidad Bayesiana y por tanto podremos utilizar los operadores de fusión definidos previamente. De esta manera se fusionarán los mapas anteriores para obtener un mapa completo (la unidad de localizaci3n)



que permitirá, entre otras tareas, la localización del robot. Además se complementará esta unidad de localización con una nueva unidad para proporcionar más robustez al sistema. Por último se verá cómo es posible, utilizando el operador de fusión competitiva, detectar si un sensor funciona de manera incorrecta y por tanto debemos descartar sus lecturas.

5.1. Introducción

Los primeros robots industriales se diseñaron únicamente para trabajar en entornos especialmente preparados, donde cada componente se encontraba situado en una posición determinada. De esta manera un robot era capaz de reconocer en cada momento donde se encontraba un objeto dado.

Si deseamos dotar a un robot de autonomía se requiere que tenga capacidades tanto para identificar variaciones en el entorno como para reaccionar respecto a éste sin requerir intervención humana. Los cambios en el entorno pueden aumentar muy rápidamente si el robot además es móvil. Las acciones que un robot puede realizar dependen del lugar donde se encuentra y por tanto dependen de la incertidumbre de localización del robot. El mapa del entorno se suele elaborar durante la navegación del robot; utilizando este mapa, el robot ha de ser capaz de planificar una trayectoria y evitar obstáculos durante su travesía.

Este problema es de difícil solución, ya a mediados de los 80 varios investigadores establecieron como deseable que un robot en movimiento por un entorno, y haciendo uso de sus sensores, fuera capaz de elaborar un mapa que luego utilizara para planificar sus actos. Incluso se replantearon si era realmente necesaria la existencia de un mapa. Así aparecieron los robots basados en comportamientos. Los robots basados en comportamientos almacenan un pequeño conjunto de variables, que constituyen su estado. Son capaces de evitar obstáculos, alcanzar objetivos, seguir muros, . . . Sin la necesidad de un mapa, tan sólo reaccionando ante la entrada de sus sensores. Entre la línea de investigación a favor de los robots basados en los comportamientos y la de los robots basados en modelos apareció una tercera filosofía híbrida que consistía en la utilización de un modelo para predecir el estado del entorno, que



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

133

fuera eficaz para el desarrollo de planes, pero con un comportamiento reactivo mínimo, como si fuera un conjunto de reflejos rápidos, necesarios para la adaptación a un entorno cambiante. El valor del mapa estaría directamente relacionado con el grado en que el robot puede predecir el entorno.

Uno de los sensores que más se ha utilizado históricamente para la elaboración de mapas del entorno son los sensores de ultrasonidos. Estos han sido ampliamente utilizados por los investigadores por sus características frente a otros sistemas de obtención de medidas, como por ejemplo el láser o los sistemas de visión. Sus principales ventajas son el bajo precio, bajo consumo comparado con los sistemas de visión, sencillez de utilización, rapidez y bajo coste algorítmico, además no son sensibles al humo ni a la luminosidad ambiente y su peso es bajo. No obstante no todo son ventajas, entre sus principales inconvenientes destacan la pobre direccionalidad del haz, lo que hace difícil la tarea de discriminar entre obstáculos próximos, la dificultad de discernir entre ecos de sensores cercanos, un comportamiento especular y la alta sensibilidad a la temperatura y humedad del medio por el que se propaga el haz.

Para salvar algunos de los problemas que sufren los sensores de ultrasonidos se propuso la utilización de matrices de sensores disparados separadamente en el tiempo utilizando una combinación entre el tiempo de emisión/recepción y la amplitud de la señal [KB90], [JK95]. De esta manera se proporcionaban una serie de medidas que fusionándolas permitían aliviar algunos de los problemas que aparecían al utilizar un único sensor. Las técnicas de fusión más utilizadas para el mapeado robótico se basan en modelos de localización Markoviana o variantes de los filtros de Kalman [DBM03a]. Otras técnicas de fusión son los mapas de rejillas, los mapas topológicos y los mapas de métrica completa.

Cuando se realiza un proceso de fusión utilizando un modelo de localización Markoviana un robot puede observar sus estados utilizando sus sensores pero también puede influenciarlos enviando comandos a sus actuadores. No obstante en estos modelos las variables de acción sólo se utilizan como variables de entrada ya que el modelo no tiene en cuenta el cálculo de la probabilidad de la distribución de las acciones. Los Mapas Bayesianos se presentan como una generalización de los modelos de Localización Markoviana. Este modelo utiliza un cálculo explícito de la distribución de las variables de estado de manera óptima y permite tener en cuenta la distribución probabilística de las acciones.

En este capítulo se aportará una aplicación de fusión complementaria uti-

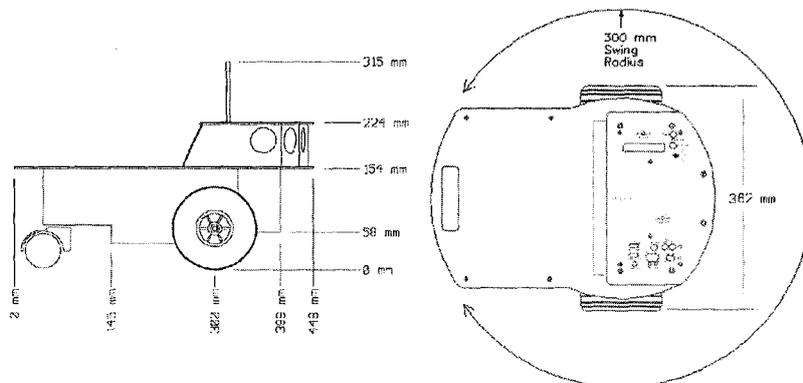


Figura 5.1: Dimensiones y forma del robot Pioneer I

lizando únicamente sensores de ultrasonidos. Se proporcionarán varios mapas Bayesianos para la fusión de la información y se verá como un mapa Bayesiano es un tipo especial de unidad Bayesiana. Se aplicará el operador de fusión complementaria para combinar diversos mapas Bayesianos y se formará un nuevo mapa que contenga la información de los submapas que lo forman. El mapa resultante de aplicar el operador de fusión complementaria, llamado unidad de localización, se combinará con una unidad Bayesiana encargada de controlar en última instancia los movimientos del robot. Por último se verá cómo el agente aquí presentado es capaz de detectar, utilizando únicamente las lecturas de los sensores de ultrasonido, cualquier fallo en su sistema sensorial, pudiendo continuar con su tarea descartando las lecturas proporcionadas por sensores erróneos.

5.2. Plataforma robótica utilizada

Para el desarrollo de las tareas de fusión comentadas anteriormente es necesario disponer de una plataforma de desarrollo donde poder probar de manera realista los modelos propuestos. Para la realización de las pruebas de este capítulo se ha utilizado tanto un robot Pioneer I como un modelo del mismo proporcionado por la empresa ActivMedia junto con el software de control Saphira [KMRS97].

El robot Pioneer I, desarrollado por ActivMedia, es un robot de dos

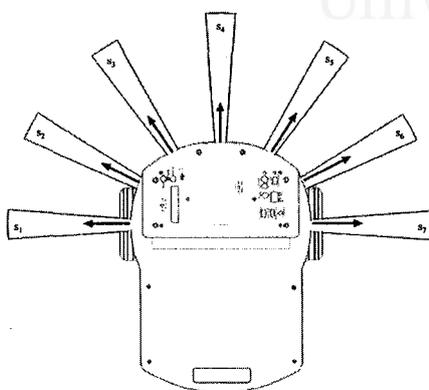


Figura 5.2: Anillo de sónares del robot Pioneer I

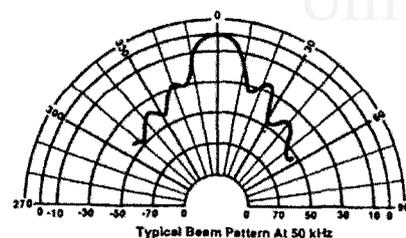
ruedas con motor diferencial, de tamaño y peso reducido. Este robot incluye sensores odométricos y de velocidad además de un anillo de siete sensores de ultrasonidos. Los sensores de ultrasonidos pertenecen a la serie MINI-S PB de SensComp. Se trata de transductores electroestáticos de 50 KHz con circuitería electrónica SMT. La distancia de trabajo para la cual están configurados es de 0.15 a 3m, con una exactitud de un 1% con un ratio de repetición de 10Hz.

Este robot dispone de tres modos de operación: un modo de autoverificación, un modo de funcionamiento autónomo (programando la memoria PROM para acceder al microcontrolador MC68HC11) o un modo servidor, que es el que utilizaremos para el desarrollo de las pruebas conectando un PC al robot. Cuando disponemos al robot en este último modo es el microcontrolador del robot, basado en el sistema operativo PSOS, el que maneja las funciones de bajo nivel del robot, incluyendo el disparo de los sónares de ultrasonidos, la activación de los motores, recoger la información de los sensores odométricos, etc. . .

El software cliente, Saphira, realiza las tareas de alto nivel, manejando las rutinas de computación intensiva como la interpretación de los sensores, la realización de tareas básicas de navegación, etc. . . Más concretamente se utilizará el software Saphira como puente con Matlab para proporcionar a este último entorno la posibilidad de manejar al robot directamente.

Matlab¹ es la abreviatura de Matrix Laboratory (laboratorio de matrices).

¹<http://www.mathworks.com>



Note: dB normalized to on-axis response.
Note: Curves are representative only. Individual responses may differ.

Figura 5.3: Patrón del haz de los sensores de ultrasonidos MINI-S PB.

Es un programa de matemáticas creado por The MathWorks en 1984. Está disponible para las plataformas Unix, Windows y MAC. Se pueden ampliar sus capacidades con Toolboxes, algunas de ellas están destinadas al procesamiento digital de señal, adquisición de datos, economía, inteligencia artificial, lógica difusa... En este caso se ha desarrollado una Toolbox para poder acceder de manera remota a cualquier robot de ActivMedia que se pueda interconectar utilizando Saphira. Matlab usa un lenguaje de programación creado en 1970 para proporcionar un acceso sencillo al software de matrices LINPACK y EISPACK sin tener que usar Fortran. También tiene su propio compilador.

Cabe remarcar que la empresa ActivMedia también proporciona un entorno de simulación que de manera transparente permite utilizar mundos virtuales para poder evaluar las conductas del robot previamente a su implantación real. Se trata de un simulador realista, que tiene en cuenta el modelo odométrico del robot, sus errores y el ruido existente en la obtención de las medidas de los sensores de ultrasonidos [KMRS97]. Para este capítulo en concreto se ha combinado el uso del simulador robótico con el robot real.

5.3. Unidad de localización

La definición de un Mapa Bayesiano (MB), depende del entorno sobre el cual el robot vaya a interactuar y de la tarea a realizar. Esta dependencia no tiene por qué ser fuerte, ya que a un mapa se le puede dotar de mecanismos de aprendizaje y por tanto puede adaptarse a entornos desconocidos, pero siempre dentro del conocimiento previo sobre el entorno, es decir de la descomposición y formas propuestas.

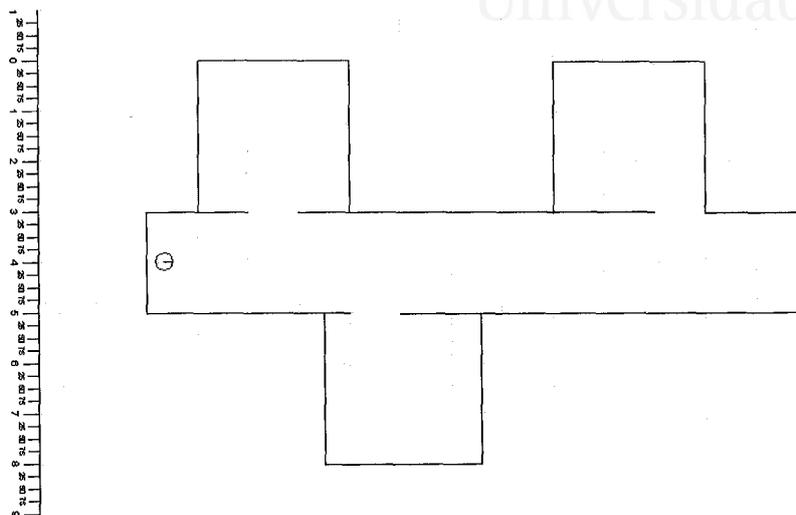


Figura 5.4: Entorno donde se realizarán las pruebas. Representa una parte del pasillo del Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante.

En nuestro caso se ha supuesto que el robot va a moverse en el entorno de la figura 5.4 realizando una tarea de navegación. El agente robótico servirá de conexión entre las distintas habitaciones pudiéndose desplazar de una hasta otra. Para formalizar la tarea es conveniente dividirla. De esta manera se definen cuatro subtareas:

- El robot está en el pasillo (y tiene que avanzar a través de él).
- El robot reconoce una puerta derecha (y puede ignorarla o disponerse para atravesarla).
- El robot detecta el final del pasillo (y debe girar 180°).
- Y finalmente el robot detecta únicamente una pared (y debe evitarla).

Una vez que se han especificado las subtareas hay que definir un mapa Bayesiano para cada situación. Para este entorno se crearán cuatro MB encargados de localizar al robot. Como se comentó en la sección 3.4, un MB es una descripción que incluye cuatro variables, una de percepción, una de acción y dos de localización. Esta descripción se basa en el formalismo de la



programación Bayesiana y por tanto para su definición es necesario especificar para cada MB las partes de un programa Bayesiano, enumeradas en la figura 3.1.

Por último cabe recordar que en el desarrollo de esta tarea se utilizará un modelo del robot Pioneer I, que dispone de siete sensores de ultrasonidos, definidos por la variable de percepción $R = R_x = \{R_1, R_2, \dots, R_7\}$ y dos actuadores controlados por la variable $A = \{V_{rot}, V_{tras}\}$, representando la velocidad rotacional y traslacional del robot respectivamente.

5.3.1. Mapa Bayesiano del Pasillo, c_p

Definición

El primer mapa propuesto c_p describe la navegación a través de un pasillo genérico. Para este mapa se define una variable de localización L como $L = \{\theta, d_x\}$ donde θ representa el ángulo entre el robot y el pasillo y d_x la distancia a su centro (ver figura 5.5a)

Es conveniente discretizar tanto la variable θ como la variable d_x . La discretización permite obtener un modelo más compacto aunque ha de ser lo suficientemente exacto para poder llevar a cabo las tareas requeridas por el mapa. En este caso la variable θ se ha discretizado en cuatro regiones $|\theta| = 4$, que dividen el sector $(0..π)$ en cuatro partes iguales, obteniendo así incrementos de $\frac{π}{4}$ radianes. La variable d_x se ha discretizado en 5 regiones $|d_x| = 5$ que dividen al ancho del pasillo en partes iguales.

Aunque a priori la variable L' ha de compartir el mismo dominio que la variable L puede ser conveniente ampliar su dominio. En algunas situaciones será necesario que una determinada acción desemboque en una nueva localización en otro mapa distinto al actual. De esta manera dada la localización actual L puede ser conveniente el alcanzar una localización futura L' relativa al mapa actual pero con mayor probabilidad de localización en un mapa distinto (y por tanto no sería productivo incluir dicha localización en la variable L). Los valores de L' se determinarán pues por el dominio de L junto con los valores necesarios para desarrollar los comportamientos del mapa actual.

Además de la variable de localización L en el instante t y la variable de localización L' en el instante $t + 1$, el mapa estará formado por la variable de percepción R y la variable de acción A definidas previamente. Utilizando las variables anteriores se propone la siguiente descomposición de la unión de distribuciones que representa el mapa:



**CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA
COMPLEMENTARIA**

139

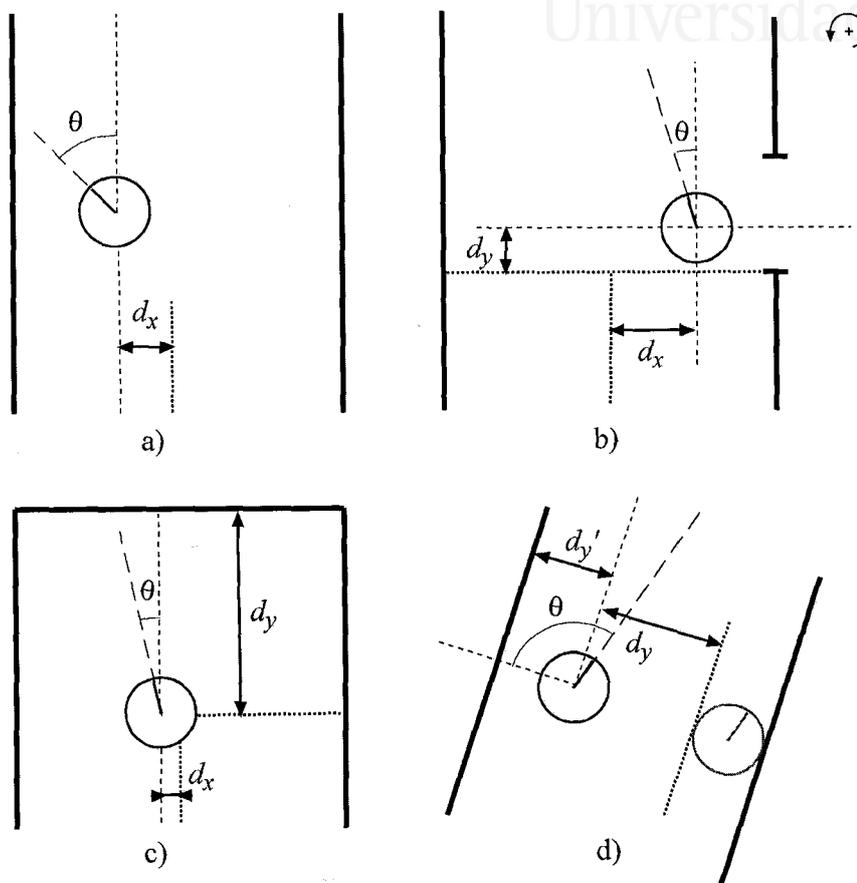


Figura 5.5: Mapas Bayesianos propuestos. a) Mapa del pasillo (c_p), d_x define la distancia del robot al centro del pasillo, θ representa el ángulo entre el robot y el pasillo. b) Mapa de la puerta (c_u), d_x y d_y definen la distancia del robot al centro del pasillo y al inicio de la puerta respectivamente, θ representa el ángulo entre el robot y el centro del pasillo. c) mapa de fin de pasillo (c_e), d_x define la distancia del robot al centro del pasillo, d_y la distancia del robot a la pared final y θ el ángulo del robot respecto a la esquina. d) Mapa de la pared (c_a), d'_y representa la distancia del robot a la pared detectada. Por otra parte θ representa el ángulo del robot respecto a la pared. Se considera positivo un ángulo si se especifica en sentido antihorario.



$$\begin{aligned}
& P(R_x \otimes \theta \otimes d \otimes \theta' \otimes d' \otimes Vrot \otimes Vtras | c_p) = \\
& = P(R_x | c_p) \times P(\theta \otimes d | R_x \otimes c_p) \times P(\theta' \otimes d' | R_x \otimes \theta \otimes d \otimes c_p) \times \\
& \quad \times P(Vrot | R_x \otimes \theta \otimes d \otimes \theta' \otimes d' \otimes c_p) \times \\
& \quad \times P(Vtras | R_x \otimes \theta \otimes d \otimes \theta' \otimes d' \otimes Vrot \otimes c_p) \\
& = P(R_x | c_p) \times P(\theta \otimes d | R_x \otimes c_p) \times P(\theta' \otimes d' | \theta \otimes d \otimes c_p) \times \\
& \quad \times P(Vrot | \theta \otimes d \otimes \theta' \otimes d' \otimes c_p) \times P(Vtras | \theta \otimes d \otimes \theta' \otimes d' \otimes c_p)
\end{aligned}$$

Una vez especificada la unión de distribuciones es necesario identificar los términos que la forman. $P(R_x | c_p)$ se puede considerar como una distribución uniforme suponiendo que inicialmente ninguna lectura tiene una probabilidad de surgir mayor que el resto. $P(\theta' \otimes d' | \theta \otimes d \otimes c_p)$ también se puede considerar uniforme suponiendo que inicialmente se puede ejecutar cualquier acción en una posición L y por tanto existe la misma probabilidad de llegar a cualquier posición L' .

Por otra parte $P(\theta \otimes d | R_x \otimes c_p)$ describe, utilizando una lectura, el ángulo y la distancia del robot en el pasillo. No obstante, sería mucho más fácil en vez de proporcionar un modelo directo, calcular el modelo inverso $P(R_x | \theta \otimes d \otimes c_p)$ ya que para éste podríamos obtener la distribución utilizando directamente el robot (tomando medidas en el entorno real). No obstante podemos deducir una distribución a partir de la otra ya que:

$$\begin{aligned}
P(R_x | L) &= \frac{1}{\Sigma} \times \sum_{L'A} \left(P(R_x | c_p) \times P(L | R_x \otimes c_p) \times P(L' | R_x \otimes L \otimes c_p) \times \right. \\
& \quad \left. \times P(A | R_x \otimes L \otimes L' \otimes c_p) \right) \\
&= \frac{1}{\Sigma} \times P(R_x | c_p) \times P(L | R_x \otimes c_p) \times \sum_{L'} P(L' | R_x \otimes L \otimes c_p) \times \\
& \quad \times \sum_A P(A | R_x \otimes L \otimes L') \\
&= \frac{1}{\Sigma} \times P(R_x | c_p) \times P(L | R_x \otimes c_p) \\
&= \frac{1}{\Sigma'} \times P(L | R_x \otimes c_p)
\end{aligned} \tag{5.1}$$

De esta manera el robot se puede disponer en distintos ángulos θ y en diferentes distancias d_x para obtener el valor de las lecturas R_x . Cada sensor R_i se representará utilizando una distribución Gaussiana que mostrará la media y la varianza para cada ángulo y distancia.

$P(Vrot | \theta \otimes d_x \otimes \theta' \otimes d'_x \otimes c_p)$ y $P(Vtras | \theta \otimes d_x \otimes \theta' \otimes d'_x \otimes c_p)$ muestran la velocidad (rotacional o traslacional) necesarias para dado el ángulo y la distancia actual llegar a un ángulo y una distancia final. Estos términos se



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

141

pueden especificar de manera sencilla utilizando una tabla proporcionada por el programador.

Por otra parte, para que un mapa Bayesiano se considere útil ha de generar comportamientos [DO00]. Con la configuración utilizada el mapa del pasillo es capaz de realizar entre otros los siguientes comportamientos:

- *Cp.SeguirPasilloCentrado*, que correspondería a la pregunta $P(V_{rot} \otimes V_{tras} | R_x [L' = \langle 0, -170 \rangle])$. Es decir, se desea obtener la velocidad rotacional y la velocidad traslacional dada una lectura R_x y la localización L' que deseamos obtener.
- *Cp.SeguirPasilloIzq*, que correspondería a la pregunta $P(V_{rot} \otimes V_{tras} | R_x [L' = \langle 0, -500 \rangle])$.
- *Cp.SeguirPasilloDer*, que correspondería a la pregunta $P(V_{rot} \otimes V_{tras} | R_x [L' = \langle 0, 500 \rangle])$.
- *Cp.CambiarSentido*, que correspondería a la pregunta $P(V_{rot} \otimes V_{tras} | R_x [L' = \langle -180, 2 \rangle])$.

Utilizando el mismo razonamiento que en esta sección se propone el mapa Bayesiano de la figura 5.6.

Identificación de los términos

Tal y como se ha comentado anteriormente es necesario realizar un proceso de aprendizaje para obtener información sobre la distribución $P(R_x | \theta \otimes d \otimes c_p)$. Situando al robot en diferentes ángulos θ y diferentes distancias d_x se grabarán los valores emitidos por los sones de ultrasonidos R_i . En cada posición y ángulo se obtendrán 10 medidas que serán fusionadas para cada sensor consiguiendo de esta manera más robustez en las lecturas individuales. Para cada sensor individual se obtendrá una Gaussiana representada por su media y varianza (μ, σ^2) . En la figura 5.7 se muestra la posición del robot para la obtención de las medidas de las figuras 5.8 y 5.9 donde se muestra el detalle de 4 de las 12 medidas ($|L| = |\theta| \times |d_x| = 12$) necesarias para completar el entrenamiento del mapa c_p . En la figura 5.10 se muestra el valor de la distribución $P(R_x | \theta \otimes d \otimes c_p)$ completa.

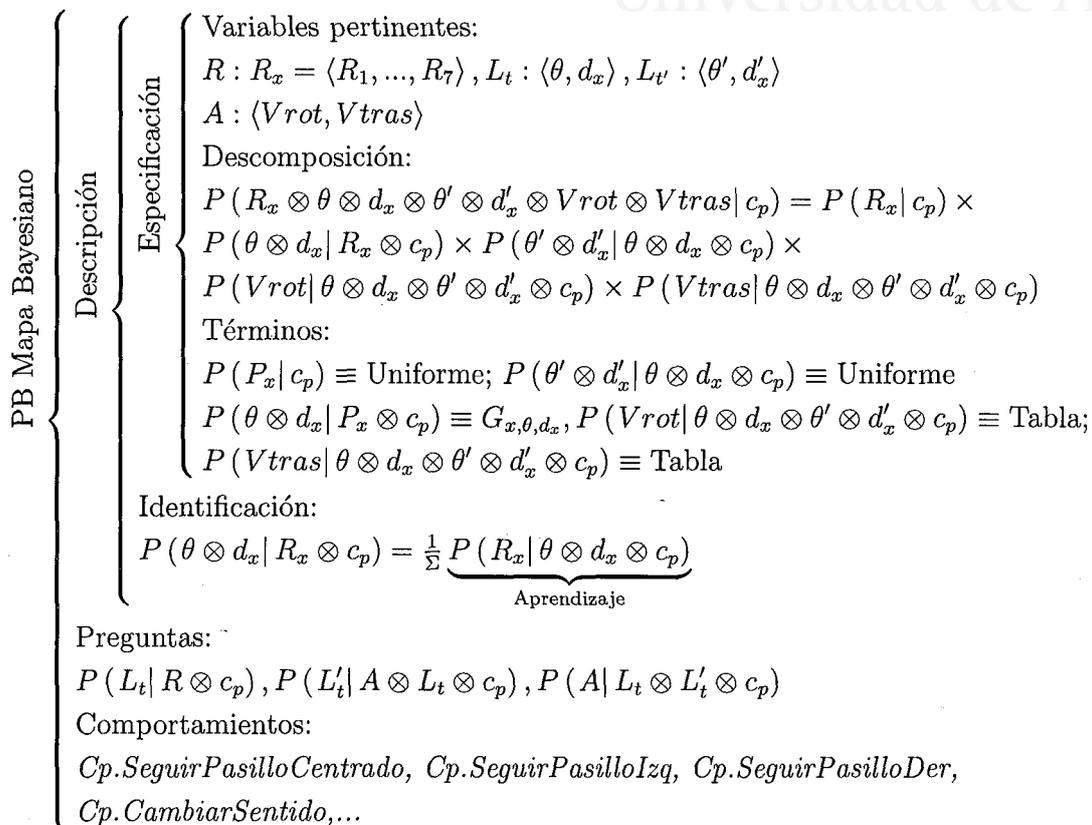


Figura 5.6: Formalización del mapa Bayesiano del pasillo utilizando el formalismo de la programación Bayesiana.

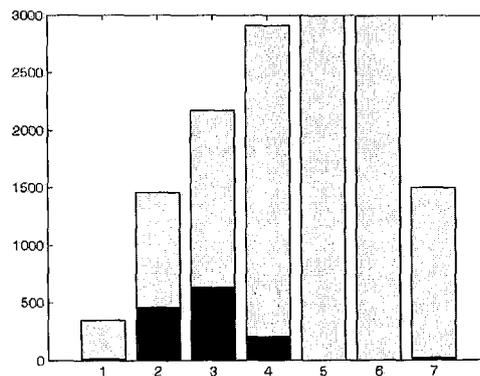
Caso 10	\odot	(500,0,25)
Caso 9	\ominus	(170,0,-25)
Caso 5	$\omin�$	(-170,0,0)
Caso 1	$\omin�$	(-500,0,25)

Figura 5.7: Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa pasillo (c_p) .

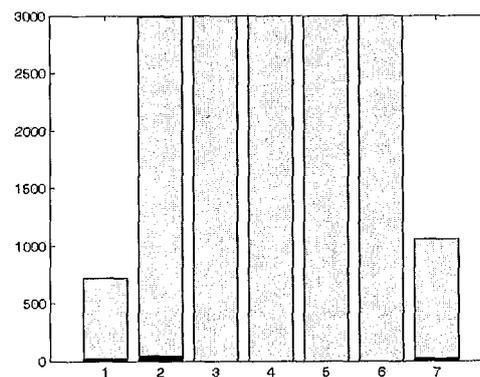


CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

143

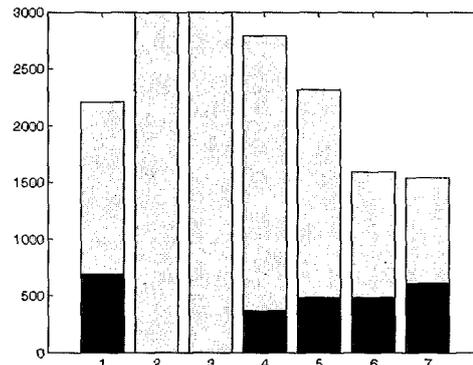


a)

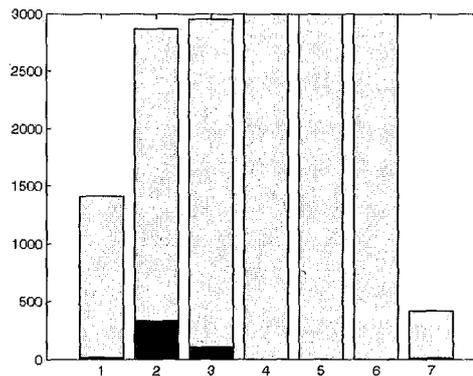


b)

Figura 5.8: Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (-500, 0, 25)$. b) Medidas en la posición $(x, y, \theta) = (-170, 0, 0)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.



a)



b)

Figura 5.9: Valores de los sensores de ultrasonidos ($R_1...R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (170, 0, -25)$. b) Medidas en la posición $(x, y, \theta) = (500, 0, 25)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.



**CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA
COMPLEMENTARIA**

145

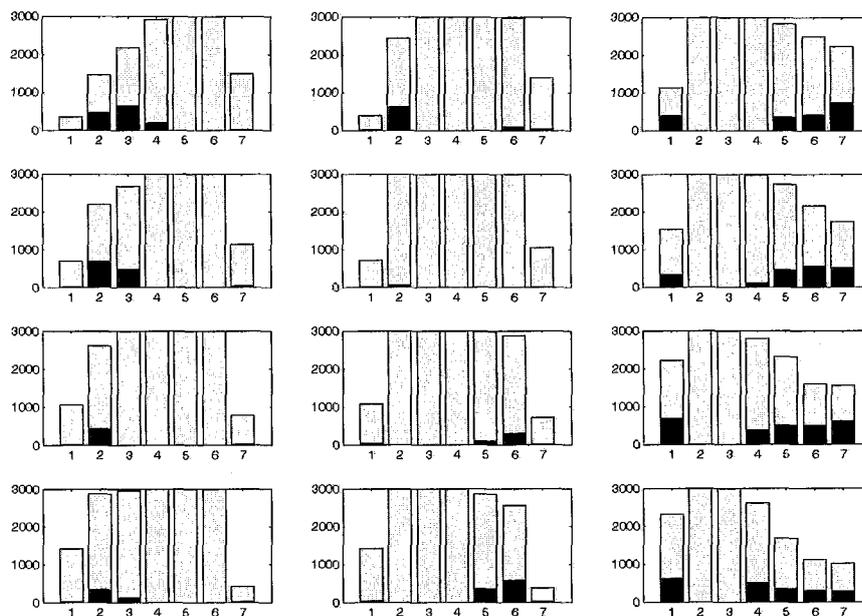


Figura 5.10: Distribución $P(R_x|\theta \otimes d \otimes c_p)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-25, 0, 25\}$ para las posiciones $d_x = \{-500, -170, 170, 500\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros.



5.3.2. Mapa Bayesiano de la Puerta, c_u

Definición

En este mapa se describe la navegación a través de un pasillo que dispone de una puerta situada a la derecha del robot. Para este mapa se define una variable de localización L como $L = \{\theta, d_x, d_y\}$ donde θ representa el ángulo entre el robot y la puerta (siendo $\theta = 0$ cuando el robot forma $\frac{\pi}{2}$ radianes con la puerta), d_x la distancia a su centro y d_y la distancia al inicio de la puerta (ver figura 5.5b).

Al igual que en el caso anterior es conveniente discretizar la variable L . En este caso la variable θ se ha simplificado en dos regiones $|\theta| = 2$, concretamente 5° a la derecha de la puerta o 5° a su izquierda. La variable d_x se ha discretizado en 2 regiones $|d_x| = 2$, y por último la variable d_y se ha discretizado también en dos regiones $|d_y| = 2$.

El mapa Bayesiano completo estará formado por la variable de percepción R , las variables de localización L y L' y la variable de acción A definidas previamente. Utilizando las variables anteriores se propone la siguiente descomposición de la unión de distribuciones que representa el mapa:

$$\begin{aligned}
 & P(R_x \otimes \theta \otimes d_x \otimes d_y \otimes \theta' \otimes d'_x \otimes d'_y \otimes Vrot \otimes Vtrans | c_u) = \\
 & = P(R_x | c_u) \times P(\theta \otimes d_x \otimes d_y | R_x \otimes c_u) \times P(\theta' \otimes d'_x \otimes d'_y | R_x \otimes \theta \otimes d_x \otimes d_y \otimes c_u) \times \\
 & \quad \times P(Vrot | R_x \otimes \theta \otimes d_x \otimes d_y \otimes \theta' \otimes d'_x \otimes d'_y \otimes c_u) \times \\
 & \quad \times P(Vtrans | R_x \otimes \theta \otimes d_x \otimes d_y \otimes \theta' \otimes d'_x \otimes d'_y \otimes Vrot \otimes c_u) \\
 & = P(R_x | c_u) \times P(\theta \otimes d_x \otimes d_y | R_x \otimes c_u) \times P(\theta' \otimes d'_x \otimes d'_y | \theta \otimes d_x \otimes d_y \otimes c_u) \times \\
 & \quad \times P(Vrot | \theta \otimes d_x \otimes d_y \otimes \theta' \otimes d'_x \otimes d'_y \otimes c_u) \times \\
 & \quad \times P(Vtrans | \theta \otimes d_x \otimes d_y \otimes \theta' \otimes d'_x \otimes d'_y \otimes c_u)
 \end{aligned}$$

Al igual que en el mapa anterior, una vez especificada la unión de distribuciones es necesario identificar los términos que la forman. $P(R_x | c_u)$ se puede considerar como una distribución uniforme suponiendo que inicialmente ninguna lectura tiene una probabilidad de surgir mayor que el resto. $P(\theta' \otimes d'_x \otimes d'_y | \theta \otimes d_x \otimes d_y \otimes c_u)$ también se puede considerar uniforme suponiendo que inicialmente se puede ejecutar cualquier acción en una posición L y por tanto existe la misma probabilidad de llegar a cualquier posición L' .

$P(\theta \otimes d_x \otimes d_y | R_x \otimes c_u)$ se describe utilizando una lectura, además del ángulo y la distancia del robot en el pasillo. No obstante, igual que en el mapa anterior, sería mucho más fácil en vez de proporcionar un modelo directo



calcular el modelo inverso $P(R_x|\theta \otimes d_x \otimes d_y \otimes c_u)$ ya que para éste podríamos obtener la distribución utilizando directamente el robot (tomando medidas en el entorno real). Podemos obtener una distribución a partir de la otra tal y como se mostró en la ecuación 5.1.

$P(Vrot|\theta \otimes d_x \otimes d_y \otimes \theta' \otimes d'_x \otimes d'_y \otimes c_u)$ y $P(Vtras|\theta \otimes d_x \otimes d_y \otimes \theta' \otimes d'_x \otimes d'_y \otimes c_u)$ muestran la velocidad (rotacional o traslacional) necesarias para dado el ángulo y la distancia actual llegar a un ángulo y una distancia relativa a la puerta. Estos términos se pueden especificar de manera sencilla, como en el caso anterior, utilizando una tabla proporcionada por el programador.

Como se comentó anteriormente para que un mapa Bayesiano se considere útil ha de generar comportamientos. Con la configuración utilizada en este mapa se pueden realizar entre otros los siguientes comportamientos:

- *Cu. IgnorarPuerta*, que correspondería a la pregunta $P(Vrot \otimes Vtras|R_x[L' = \langle 0, -280, 530 \rangle])$.
- *Cu. PrepararseAtravesar*, que correspondería a la pregunta $P(Vrot \otimes Vtras|R_x[L' = \langle -85, 280, 270 \rangle])$.
- *Cu. CambiarSentido*, que correspondería a la pregunta $P(Vrot \otimes Vtras|R_x[L' = \langle -180, -280, 270 \rangle])$.

Con el mismo razonamiento se propone el mapa Bayesiano de la figura 5.11.

Identificación de los términos

Al igual que en el caso anterior es necesario realizar un proceso de aprendizaje para obtener información sobre la distribución $P(R_x|\theta \otimes d_x \otimes d_y \otimes c_u)$. Situando al robot en diferentes ángulos θ y diferentes distancias d_x, d_y se grabarán los valores emitidos por los sónares de ultrasonidos R_i . En cada posición y ángulo se obtendrán 10 medidas que serán fusionadas para cada sensor consiguiendo de esta manera más robustez en las lecturas individuales. Para cada sensor individual se obtendrá una Gaussiana representada por su media y varianza (μ, σ^2) . En la figura 5.12 se observa la posición del robot para la obtención de las medidas de la figura 5.13, donde se muestra el detalle de 2 de las 12 medidas ($|L| = |\theta| \times |d_x| = 12$) necesarias para completar el entrenamiento del mapa c_u . En la figura 5.14 se muestra el valor de la distribución $P(R_x|\theta \otimes d_x \otimes d_y \otimes c_u)$ completa.

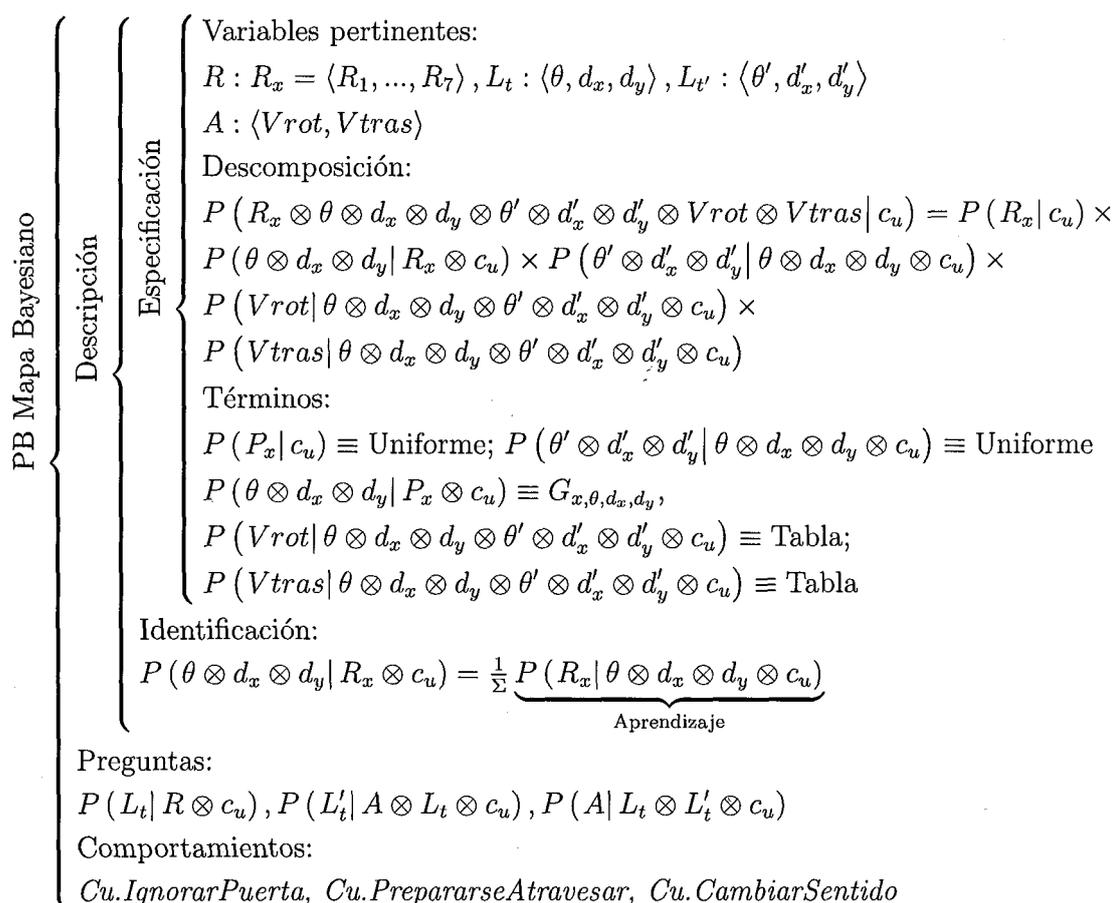


Figura 5.11: Formalización del mapa Bayesiano de la puerta utilizando el formalismo de la programación Bayesiana.



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

149

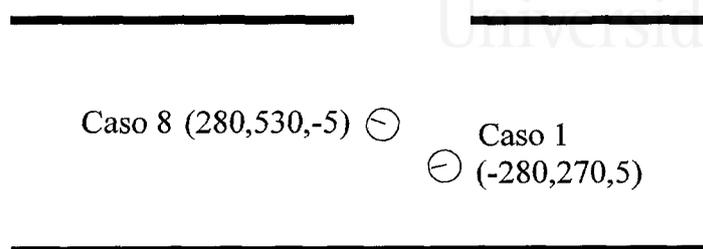


Figura 5.12: Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa puerta (c_u).

5.3.3. Mapa Bayesiano de fin de pasillo, c_e

Definición

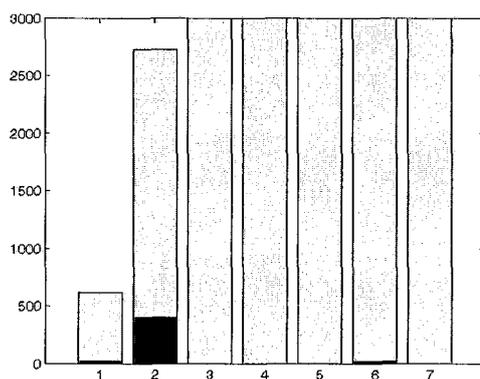
Como su nombre indica el mapa de fin de pasillo c_e describe la navegación a través de una zona terminal del pasillo, donde el robot no puede continuar. Para este mapa se define una variable de localización L como $L = \{\theta, d_x, d_y\}$ donde θ representa el ángulo entre el robot y el pasillo d_x la distancia a su centro y d_y la distancia al final del pasillo (ver figura 5.5c).

Como en los casos anteriores se discretizará tanto la variable θ como las variables d_x y d_y . En este caso la variable $|\theta| = 3$ se ha simplificado en tres regiones. La variable $|d_x| = 4$ se ha discretizado en 4 regiones que dividen el ancho del pasillo en partes iguales. La variable $|d_y| = 3$ se ha discretizado en 3 dividiendo la distancia del robot a la pared final. Esta división no es uniforme sino que se trata de una división cuadrática obteniendo más porciones de las partes cercanas al final de la pared.

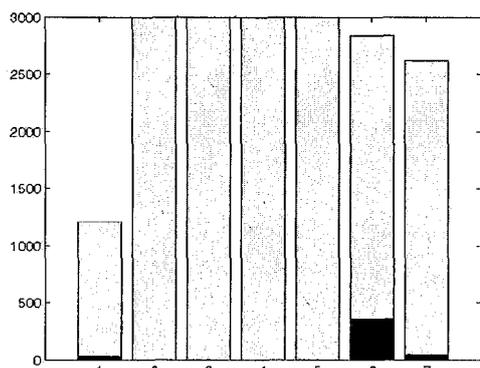
Además de la variable de localización L en el instante t y la variable de localización L' en el instante $t + 1$, el mapa estará formado por la variable de percepción R y la variable de acción A definidas previamente.

Para que un mapa Bayesiano se considere útil ha de generar comportamientos, así pues se proponen los siguientes:

- *Cp.AvanzarHastaLimite*, que correspondería a la pregunta $P(V_{rot} \otimes V_{tras} | R_x[L' = (0, -200, 2625)])$.
- *Cp.Evitar*, que correspondería a la pregunta $P(V_{rot} \otimes V_{tras} | R_x[L' = (180, -200, 1050)])$.



a)



b)

Figura 5.13: Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (-280, 270, 5)$. b) Medidas en la posición $(x, y, \theta) = (280, 530, 5)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.



**CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA
COMPLEMENTARIA**

151

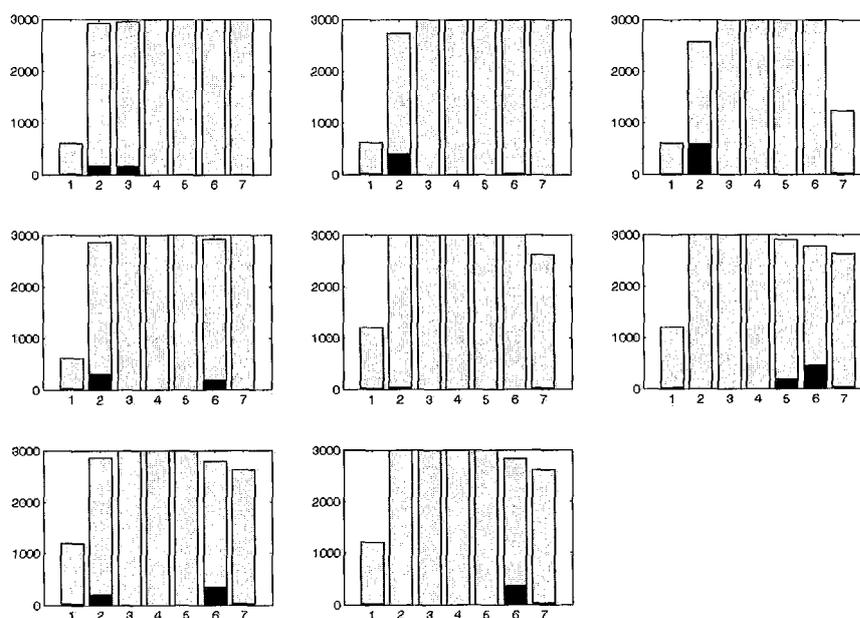


Figura 5.14: Distribución $P(R_x | \theta \otimes d_x \otimes d_y \otimes c_u)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-5, 5\}$ para las posiciones $d_x = \{-280, 280\}$ y $d_y = \{270, 530\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros.



- *Cp.EvitarDer*, que correspondería a la pregunta $P(Vrot \otimes Vtras | R_x[L' = \langle 180, 600, 1050 \rangle])$.
- *Cp.EvitarIzq*, que correspondería a la pregunta $P(Vrot \otimes Vtras | R_x[L' = \langle 180, -600, 1050 \rangle])$.

Procediendo de la misma manera que en la sección 5.3.2 se propone el mapa Bayesiano de la figura 5.15.

Identificación de los términos

Para identificar la distribución $P(R_x | \theta \otimes d_x \otimes d_y \otimes c_u)$ situamos al robot en diferentes ángulos θ y diferentes distancias d_x, d_y , registrando las lecturas de los sónares de ultrasonidos R_i . Al igual que en el resto de mapas en cada posición y ángulo se obtendrán 10 medidas para cada sónar de ultrasonidos. Cada sensor se representará por una Gaussiana con su media y varianza (μ, σ^2) . En las figuras 5.16, 5.17, 5.18 se muestra la posición en que se obtuvieron las medidas y el detalle de 4 de las 36 medidas ($|L| = |\theta| \times |d_x| \times |d_y| = 36$) necesarias para completar el entrenamiento del mapa c_e . En la figura 5.19 se muestra el valor de la distribución $P(R_x | \theta \otimes d_x \otimes d_y \otimes c_e)$ completa.

5.3.4. Mapa Bayesiano de la pared, c_a

Definición

El mapa Bayesiano de la pared c_a describe la navegación cuando el único objeto localizable es una pared. Para este mapa se define una variable de localización L como $L = \{\theta, d_y\}$ donde θ representa el ángulo entre el robot y el pasillo y $d_y = Ancho_{pasillo} - Diámetro_{Robot} - d'_y$ donde d'_y es la distancia a la pared (ver figura 5.5d).

Se discretizará tanto la variable θ como la variable d_y siendo $|\theta| = 3$ y $|d_y| = 3$.

Por otra parte para que un mapa Bayesiano se considere útil ha de generar comportamientos. Con la configuración utilizada el mapa Bayesiano de la pared es capaz de realizar entre otros los siguientes comportamientos:

- *Ca.SeguirPared*, que correspondería a la pregunta $P(Vrot \otimes Vtras | R_x[L' = \langle -90, 850 \rangle])$.

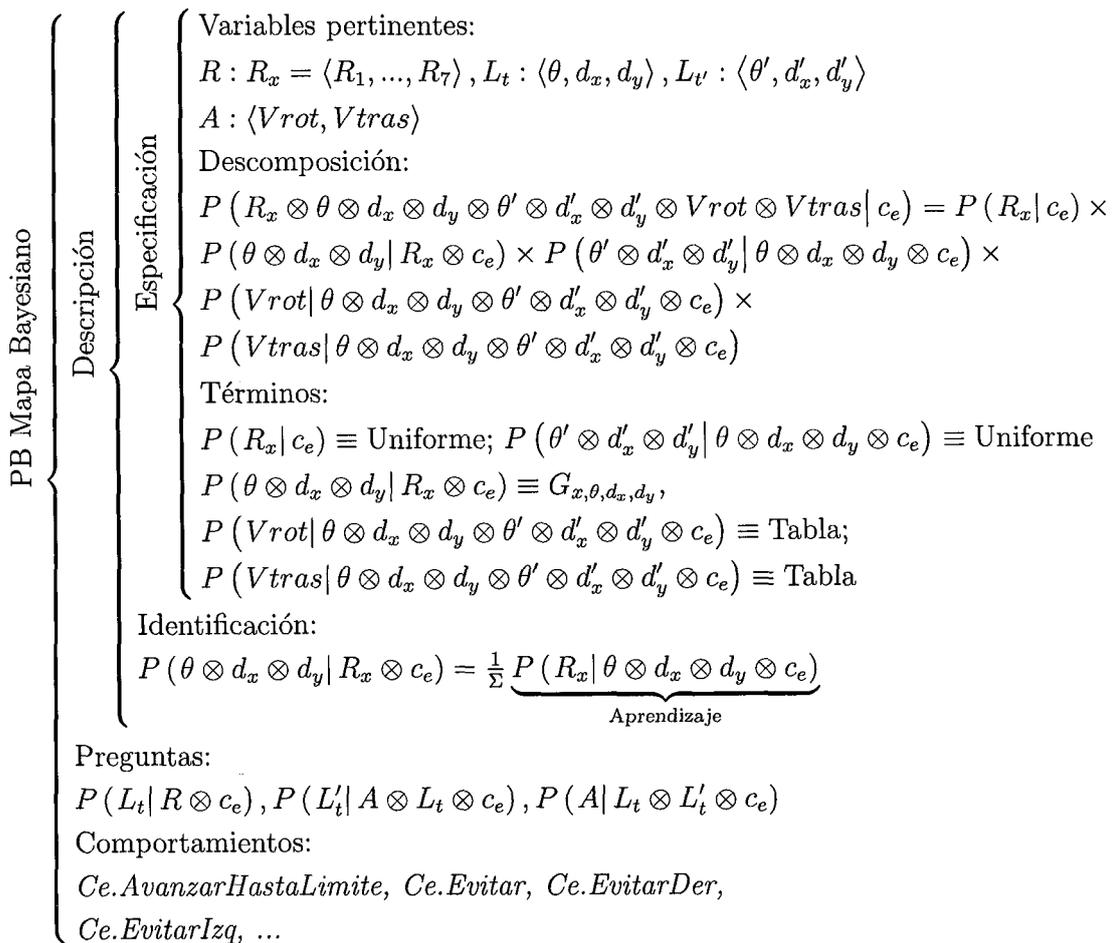


Figura 5.15: Formalización del mapa Bayesiano de final de pasillo utilizando el formalismo de la programación Bayesiana.

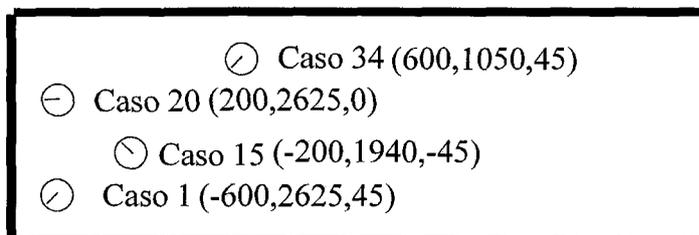


Figura 5.16: Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa puerta (c_u).

- *Ca.SeguirParedIzq*, que correspondería a la pregunta $P(Vrot \otimes Vtras | R_x[L' = \langle -90, 425 \rangle])$.
- *Ca.SeguirParedDer*, que correspondería a la pregunta $P(Vrot \otimes Vtras | R_x[L' = \langle -90, 1275 \rangle])$.

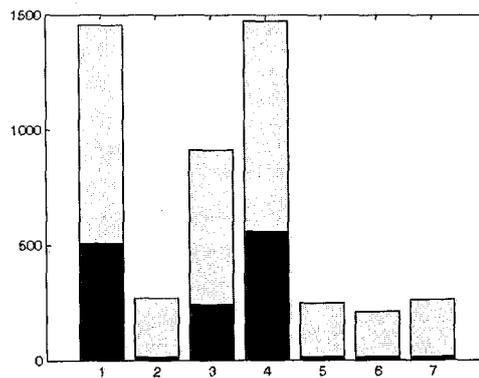
Procediendo de la misma manera que en la sección 5.3.1 se propone el mapa Bayesiano de la figura 5.20.

Identificación de los términos

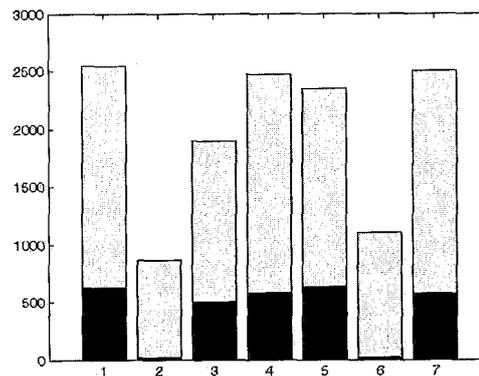
Para identificar la distribución $P(R_x | \theta \otimes d_y \otimes c_a)$ situamos al robot en diferentes ángulos θ y diferentes distancias d_x . En la figura 5.21 se muestra la posición donde se obtuvieron las medidas mostradas en detalle en la figura 5.22. En concreto podemos observar 2 de las 9 medidas ($|L| = |\theta| \times |d_y| = 9$) necesarias para completar el entrenamiento del mapa c_a . Por otra parte, en la figura 5.23, se muestra el valor de la distribución $P(R_x | \theta \otimes d_y \otimes c_a)$ completa.

5.3.5. Fusión de Mapas Bayesianos

Como se comentó en la sección 3.4 un mapa Bayesiano está formado por cuatro variables: una variable de percepción R , dos variables de localización L, L' y una de control A . Teniendo en cuenta el modelo de las unidades Bayesianas se puede considerar que la variable de percepción es una variable de entrada, que las variables de localización son variables de estado y la variable de control es una variable de salida que enviará comandos al robot. De esta manera podemos especificar un mapa Bayesiano como una unidad Bayesiana, tal que: $u : \{I = R, S_1 = L, S_2 = L', O = A\}$. Esta unidad

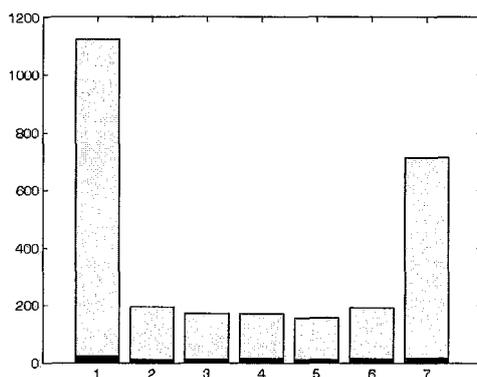


a)

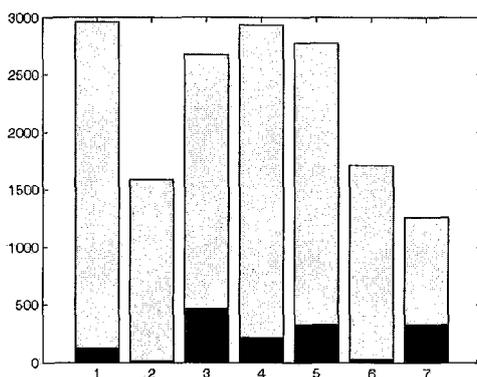


b)

Figura 5.17: Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (-600, 2625, 45)$. b) Medidas en la posición $(x, y, \theta) = (-200, 1940, -45)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.



a)



b)

Figura 5.18: Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (200, 2625, 0)$. b) Medidas en la posición $(x, y, \theta) = (600, 1050, 45)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

157

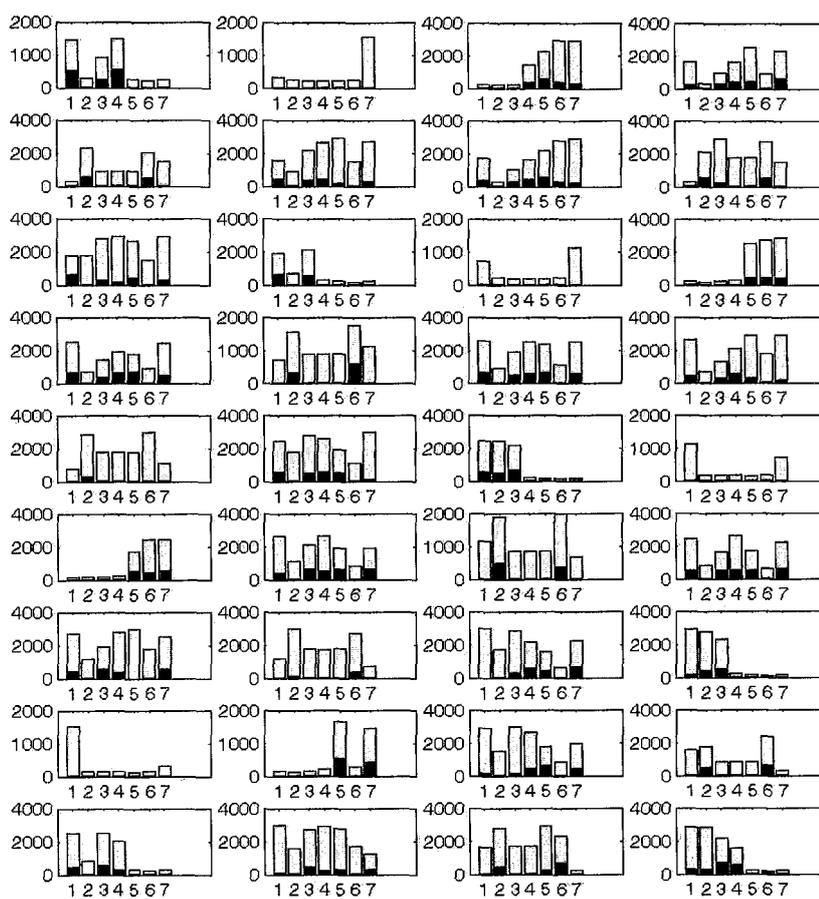


Figura 5.19: Distribución $P(R_x | \theta \otimes d_x \otimes d_y \otimes c_e)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-45, 0, 45\}$ para las posiciones $d_x = \{-600, -200, 200, 600\}$ y $d_y = \{1050, 1940, 2625\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros.

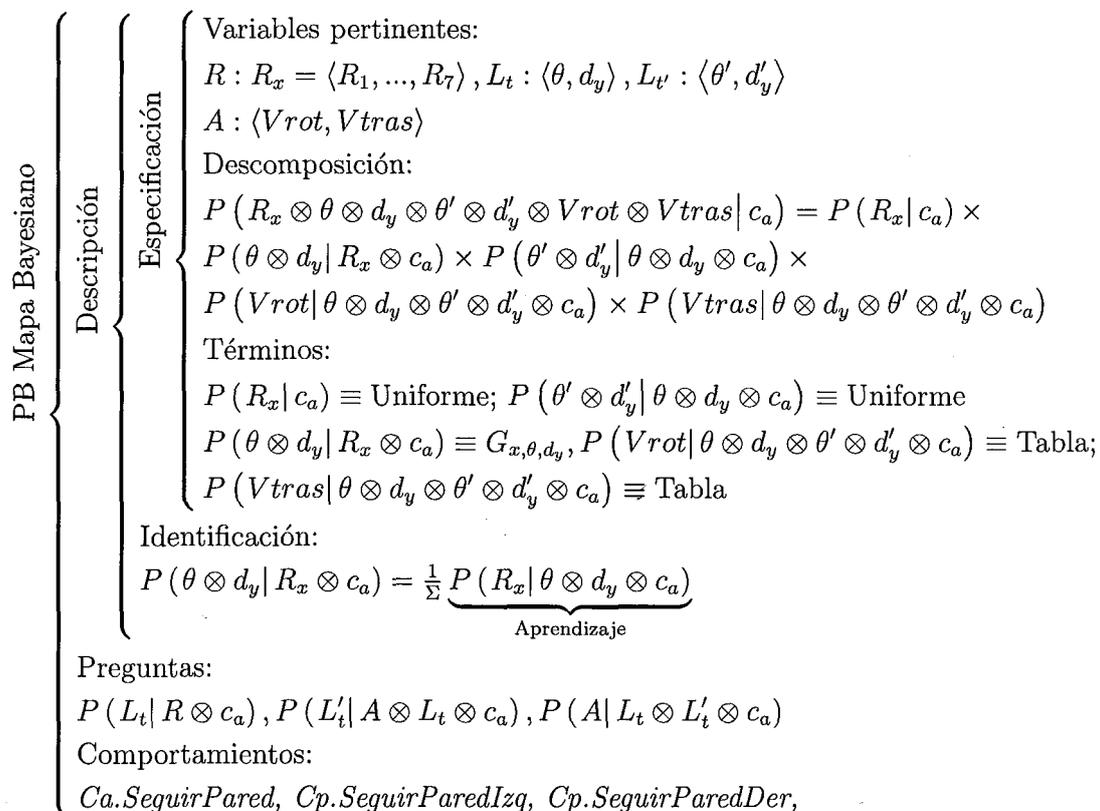


Figura 5.20: Formalización del mapa Bayesiano de la puerta utilizando el formalismo de la programación Bayesiana.

① Caso 7 (0,1275,20)

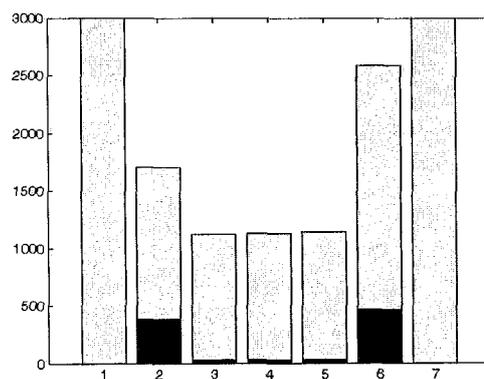
① Caso 2 (0,425,0)

Figura 5.21: Posición y ángulo del robot (x, y, θ) para la obtención de medidas R en la fase de entrenamiento del mapa pared (c_a) .

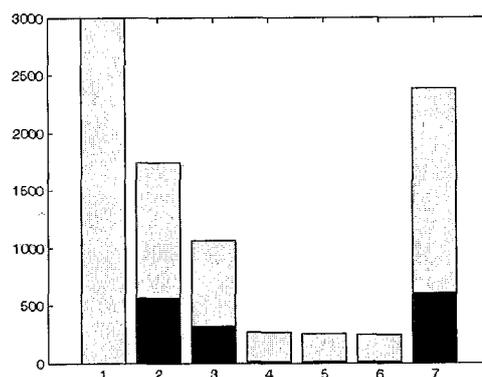


**CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA
COMPLEMENTARIA**

159



a)



b)

Figura 5.22: Valores de los sensores de ultrasonidos ($R_1 \dots R_7$). La barra oscura representa la desviación típica de las lecturas, la clara la media. a) Medidas en la posición $(x, y, \theta) = (0, 1275, 20)$. b) Medidas en la posición $(x, y, \theta) = (0, 425, 0)$. La distancia al obstáculo retornada por R_x se especifica en milímetros.

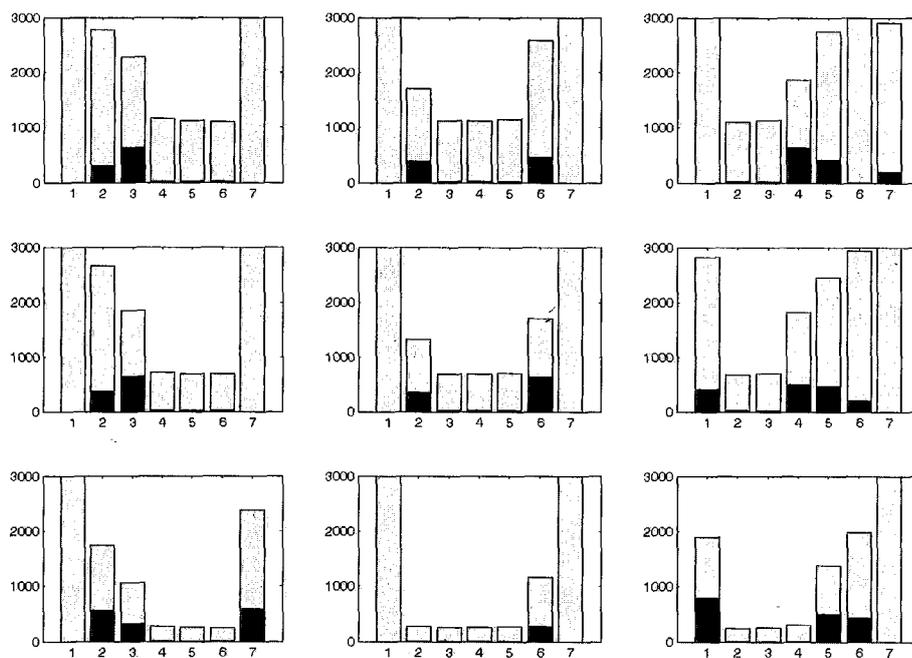


Figura 5.23: Distribución $P(R_x|\theta \otimes d_x \otimes d_y \otimes c_a)$. El sentido de la figura es de izquierda a derecha. Se empieza variando el ángulo $\theta = \{-20, 0, 20\}$ para las posiciones $d_y = \{425, 850, 1275\}$. La barra oscura representa la desviación típica de las lecturas, la clara la media. La distancia al obstáculo retornada por R_x se especifica en milímetros.



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

161

Bayesiana tendrá que responder de manera informativa a las tres preguntas guía propuestas en la sección 3.4. La pregunta de localización $P(L_t|R) = P(S_1|I)$ es informativa como se puede ver en la ecuación 4.2. De la misma manera tanto la pregunta de predicción $P(L'_t|A \otimes L_t) = P(S_2|O \otimes S_1)$ como la de control $P(A|L_t \otimes L'_t) = P(O|S_1 \otimes S_2)$ responden de manera informativa, como se demuestra en las ecuaciones 4.3, 4.4.

De esta manera podemos considerar a un mapa Bayesiano como una unidad Bayesiana sin pérdida de generalidad pudiendo aplicar los operadores definidos en el capítulo 4.3 para la fusión de unidades. Es deseable combinar los mapas descritos anteriormente para disponer de un mapa global que integre toda esta información. El operador de fusión complementaria permite fusionar unidades que no tienen dependencia entre ellas pero que se pueden combinar para obtener una imagen más completa del fenómeno a observar.

Tal y como se comentó en la sección 4.3 una unidad Bayesiana con más de un estado, resultante de fusionar dos subunidades, ha de estar compuesta de las siguientes variables:

$$\begin{aligned}
 I &= I_1 \wedge I_2 \wedge O_1 \wedge O_2 \wedge S_1 \wedge S_2, \\
 |I| &= |O_1| + |O_2| + |I_1| + |I_2| + |S_1| + |S_2| \\
 S_1 &= \{a_1, a_2\} ; S_2 = \{b_1, b_2\} \\
 |S_1| &= |S_2| = m \\
 O &= O_1 \cup O_2 \\
 |O| &= |O_1| + |O_2| - |O_1 \cap O_2| \\
 S &= \{S_1, S_2\} \\
 |S_1| + |S_2| &= 4
 \end{aligned} \tag{5.2}$$

Proponiendo la siguiente descomposición:

$$\begin{aligned}
 P(I \otimes S \otimes O) &= P(I \otimes S^1 \otimes \dots \otimes S^m \otimes O) = \\
 &P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1) \times \prod_{i=1}^n P(S^2 | I_i \otimes S_i \otimes O_i \otimes S^1) \times \\
 &\times \dots \times \prod_{i=1}^n P(S^m | I_i \otimes S_i \otimes O_i \otimes S^{m-1} \otimes S^{m-2} \otimes \dots \otimes S^1) \times \\
 &\times P(O | S^1 \otimes S^2 \otimes \dots \otimes S^m)
 \end{aligned}$$

Donde para el caso de unidades con dos estados, donde ambos son independientes entre sí, como es el caso de los mapas Bayesianos, se define la descomposición resultante del proceso de fusión como:



$$\begin{aligned}
 P(I \otimes S \otimes O) &= P(I \otimes S^1 \otimes \dots \otimes S^m \otimes O) = \\
 &P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1) \times P(S^2 | S^1) \times P(O | S^1 \otimes S^2) = \quad (5.3) \\
 &= P(S^1) \times \prod_{i=1}^n P(I_i \otimes S_i \otimes O_i | S^1) \times P(S^2) \times P(O | S^1 \otimes S^2)
 \end{aligned}$$

Una versión del operador de fusión complementaria para unidades con dos estados ha sido utilizado con éxito al aplicarse junto con los mapas Bayesianos. El operador de conjunto presentado por Diard, en [DBM04a], es un caso particular del operador de fusión complementaria que permite realizar la combinación de dos mapas Bayesianos. Cabe remarcar que este operador únicamente se especifica para su uso junto con mapas Bayesianos, quedando limitada su aplicación exclusivamente a las tareas de localización y mapeado que este tipo de mapas proporcionan.

Una vez completado el proceso de especificación de los mapas Bayesianos y definido el operador de fusión complementario podemos fusionar varios mapas para conseguir uno más general. El mapa resultante del proceso de fusión especificado en las ecuaciones 5.2, 5.3 será capaz de localizarse en las cuatro posiciones de los submapas que lo forman, siendo $I = \{I_p, I_a, I_u, I_e, S_p, \dots, S_e, O_p, \dots, O_e\}$, $S = \{c_p, c_a, c_u, c_e\}$ y O una nueva variable que trabaja en el dominio de las variables de los mapas a fusionar. En la descomposición presentada en la ecuación 5.3 las probabilidades de encontrarse inicialmente en una determinada localización $P(L)$ y $P(L')$ se suponen uniformes. Por otra parte todos los términos $P(R_i \otimes L_i \otimes L'_i \otimes A_i | L)$ se definen respecto al valor de la variable de localización del mapa resultante del proceso de fusión. Si la posición L no es reconocida por un submapa entonces suponemos que la distribución anterior es uniforme, en otro caso la distribución $P(R_i \otimes L_i \otimes L'_i \otimes A_i | L)$ estará definida de la misma manera que en el submapa al que pertenecía. Por último el término $P(A | L \otimes L')$, que podemos interpretar como dada la localización actual qué acción debemos realizar para obtener una nueva localización, se puede definir en forma de tabla o aprender experimentalmente.

Utilización de la distribución $P(R|L)$

Para utilizar la distribución $P(R|L)$ de manera sencilla es necesario determinar el parecido de un conjunto de lecturas R (las lecturas del robot



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

163

en un momento dado) a otro conjunto R' (el conjunto de lecturas que ha sido aprendido en $P(R'|L_i)$). De esta manera dada una lectura será sencillo obtener la distribución de la variable L_i para determinar la localización del robot.

Como se comentó anteriormente cada R_i se obtuvo calculando la media y varianza de diez lecturas del sónar de ultrasonidos i . Así las medidas del sónar de ultrasonidos en una posición dada tienen asignadas una forma normal que las caracteriza. Por tanto, si deseamos comparar un conjunto de siete medidas R (correspondientes a los siete sónares de ultrasonidos del robot Pioneer I) deberemos comparar siete distribuciones Gaussianas y obtener la probabilidad de que un conjunto de lecturas R represente a otro R' .

A priori las distancias desarrolladas para la comparación de histogramas no son aplicables directamente en este caso debido a que cada sónar se caracteriza por dos estadísticos distintos que hay que tener en cuenta de manera conjunta. Por ejemplo si comparásemos únicamente la media de las distribuciones nos encontraríamos con que en el caso de la figura 5.24 ambas distribuciones tendrían la misma distancia aunque es obvio que esto no debería ser así.

Una solución factible es calcular la probabilidad de error de clasificación entre dos clases, la lectura actual y la lectura aprendida, que será equivalente a calcular la probabilidad de que ambas distribuciones generen los mismos valores y por tanto sean la misma. Siendo $\{\Omega_1, \Omega_2\}$ las dos particiones en que se divide el espacio y $\bar{\Omega}_i$ el complemento de la región i , que representa cualquier región excepto Ω_i , se define la probabilidad de error entre dos clases como:

$$P(\text{error}) = \sum_{i=1}^2 \int_{\bar{\Omega}_i} P(x|w_i)P(w_i)dx$$

Siendo w_i el sensor origen de las lecturas. De esta manera para el ejemplo anterior obtendríamos que en el primer caso no existe probabilidad de error (la probabilidad de que ambas Gaussianas representen los mismos datos es próxima a 0) y sin embargo en el segundo caso el error es mayor al solaparse ambas distribuciones (ver figura 5.25). Para más información consultar [Wee99].

Antes de aplicar el proceso de fusión se debe definir cómo se combinarán las probabilidades de los múltiples sensores. Se ha realizado la siguiente propuesta:

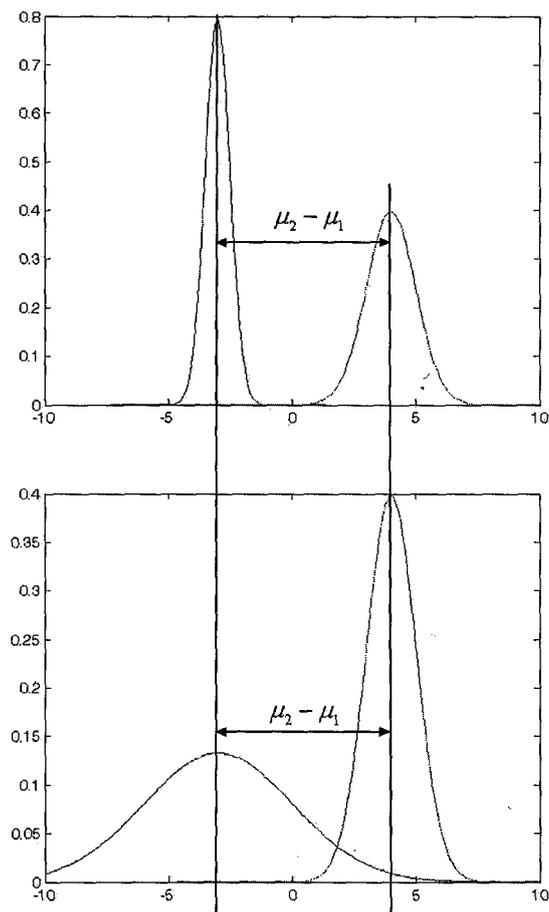


Figura 5.24: Distancia entre dos distribuciones Gaussianas sin tener en cuenta su desviación típica. En la segunda gráfica se solapan ambas distribuciones pero no se tiene en cuenta en la medida de distancia.



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

165

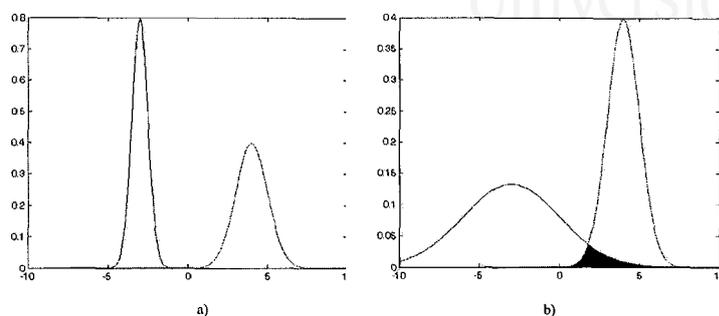


Figura 5.25: Probabilidad de que una distribución sea igual a otra. a) Prácticamente no existe probabilidad de que un valor generado por la primera distribución también sea generado por la segunda. b) La probabilidad de que un valor se genere por ambas distribuciones se representa en forma de área sombreada

$$P(Rx|L) = \prod_i P(Rx_i|L) \quad (5.4)$$

Es decir, se asume que los elementos sensoriales son independientes (se supone que conocida la causa, las consecuencias son independientes). Aunque se trata de una hipótesis fuerte proporciona una mejora de la calidad de la señal, un funcionamiento más robusto a fallos de uno de los sensores y eficiencia en la computación, como se puede ver en [LBDM04].

Resultados del proceso de fusión

Como se ha comentado anteriormente, aplicando el operador de fusión complementaria se obtiene un nuevo mapa que aglutina el conocimiento de los submapas que lo forman. No obstante en este nuevo mapa será conveniente obtener las tres preguntas guía (localización, predicción y control) requeridas para el manejo del mapa. De esta manera es fácil comprobar que:



$$\begin{aligned}
P(L_t|R) &= \sum_{L'_t, A} P(R \otimes L_t \otimes L'_t \otimes A) = \\
&= \sum_{L'_t, A} \left(P(L_t) \times \prod_{i=1}^n P(R_i, L_{t,i}, L'_{t,i}, A_i | L_t) \times P(L'_t) \times P(A | L_t \otimes L'_t) \right) \\
&= P(L_t) \times \prod_{i=1}^n P(R_i, L_{t,i}, L'_{t,i}, A_i | L_t) \times \sum_{L'_t} P(L'_t) \times \sum_A P(A | L_t \otimes L'_t) \\
&= \frac{1}{\Sigma} \times \prod_{i=1}^n P(R_i, L_{t,i}, L'_{t,i}, A_i | L_t)
\end{aligned}$$

$$\begin{aligned}
P(L'_t | A \otimes L_t) &= \sum_R P(R \otimes L_t \otimes L'_t \otimes A) = \\
&= \sum_R \left(P(L_t) \times \prod_{i=1}^n P(R_i, L_{t,i}, L'_{t,i}, A_i | L_t) \times P(L'_t) \times P(A \otimes L_t \otimes L'_t) \right) \\
&= P(L_t) \times \sum_R \prod_{i=1}^n P(R_i, L_{t,i}, L'_{t,i}, A_i | L_t) \times P(L'_t) \times P(A \otimes L_t \otimes L'_t) \\
&= \frac{1}{\Sigma'} \times P(A \otimes L_t \otimes L'_t)
\end{aligned}$$

$$\begin{aligned}
P(A | L_t \otimes L'_t) &= \sum_R P(R \otimes L_t \otimes L'_t \otimes A) = \\
&= \sum_R \left(P(L_t) \times \prod_{i=1}^n P(R_i, L_{t,i}, L'_{t,i}, A_i | L_t) \times P(L'_t) \times P(A \otimes L_t \otimes L'_t) \right) \\
&= P(L_t) \times \sum_R \prod_{i=1}^n P(R_i, L_{t,i}, L'_{t,i}, A_i | L_t) \times P(L'_t) \times P(A \otimes L_t \otimes L'_t) \\
&= \frac{1}{\Sigma''} \times P(A \otimes L_t \otimes L'_t)
\end{aligned}$$

Así podemos utilizar el mapa resultante de la fusión de la misma manera que utilizábamos los submapas, realizando preguntas a su descomposición de variables. Por ejemplo, vamos a localizarnos en el entorno de la figura 5.4 utilizando el mapa global. Este mapa dispone de una variable de estado L con cuatro posibles estados (pasillo, pared, puerta, fin de pasillo) que corresponden a cada uno de los mapas fusionados. Como se ha visto anteriormente contestar a la pregunta $P(L_t|R)$ implica el preguntar a la descomposición de variables de cada uno de los submapas.

Puede ser conveniente después de la fase de entrenamiento, en la cual se obtienen los valores de las distribuciones $P(R|L)$ para todos los submapas, comprobar el parecido de las lecturas obtenidas para todos ellos. Puede darse



el caso de que una discretización incorrecta de las variables genere una misma lectura y por tanto la localización se realice con la misma intensidad en dos mapas simultáneamente. Por ejemplo en el mapa del pasillo, si se gira demasiado el robot, es posible que deje de observar las dos paredes que lo caracterizan y vea únicamente una (confundiendo la localización con una pared).

Como ejemplo de utilización del mapa fusionado a continuación se muestra el valor más probable de L_t para 60 posiciones del robot en tres ángulos distintos.

Como se observa en las figuras 5.26, 5.27 y 5.28 el robot se localiza correctamente en la mayoría de los casos incluso para situaciones no previstas en los submapas a fusionar.

5.3.6. Desarrollo de una conducta simple

Una de las finalidades de un mapa Bayesiano es la de proporcionar comportamientos. En los apartados anteriores donde se definían los submapas (c_p, c_u, c_a, c_e) se comentaba que las distribuciones $P(Vrot|L \otimes L')$ y $P(Vtras|L \otimes L')$ se podían especificar por el programador en forma de tabla o bien ser aprendidas. A continuación se mostrará un ejemplo muy simple de una conducta de recorrido definiendo ambas distribuciones para los cuatro submapas que van a ser fusionados.

Submapa del pasillo

En esta conducta se requerirá que el robot siga el pasillo intentando centrar su ángulo y su posición respecto al mismo. De esta manera la distribución $P(Vrot|L \otimes L')$ se podría definir fácilmente para el caso en que dada una localización L únicamente se pretenda centrar al robot en el pasillo. Así podemos fijar la variable L' y especificar la tabla de la figura 5.29.

No obstante no siempre es necesario especificar la tabla completa, se puede diseñar un programa que dadas L y L' determine el valor de $Vrot$ sustituyendo a la distribución $P(Vrot|L \otimes L')$. De esta manera la tabla anterior puede ser equivalente a un programa del tipo:

- 01 $AnguloGiroD = \lfloor d_x/5 \rfloor$
- 02 $AnguloGiroA = -\theta$
- 03 $Vrot = AnguloGiroD + AnguloGiroA$

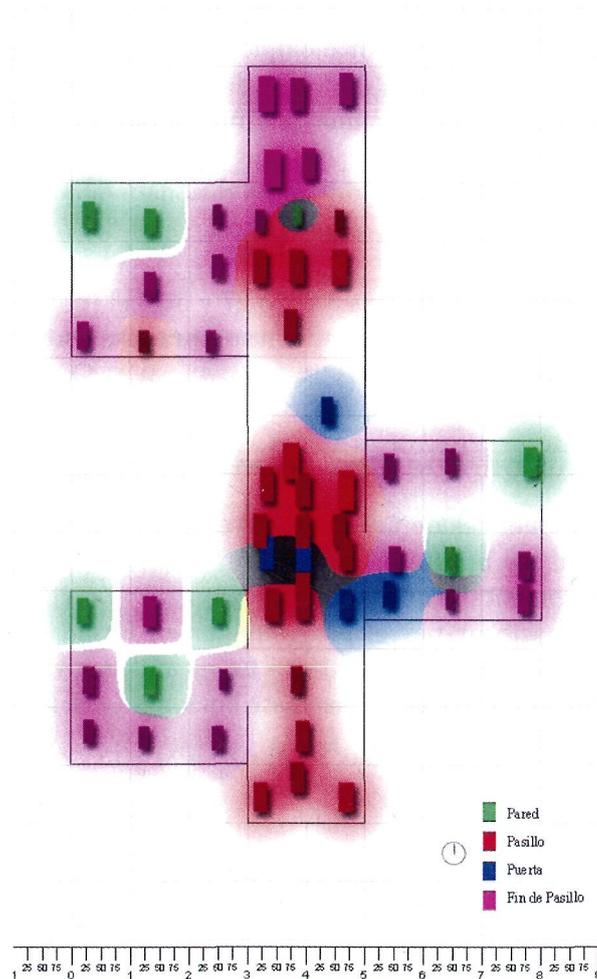


Figura 5.26: Localización del mapa global para 60 medidas (indicadas por el centro de los rectángulos de la figura). La saturación del color y el tamaño de los rectángulos indican la probabilidad de la localización (mayor tamaño mayor probabilidad de localización). El resultado de la localización es pasillo (rojo), pared (verde), puerta (azul), fin de pasillo (magenta). Las localizaciones equiprobables se envuelven con áreas del mismo color. El robot siempre tiene una posición de 0° respecto al mundo (representada en la figura).

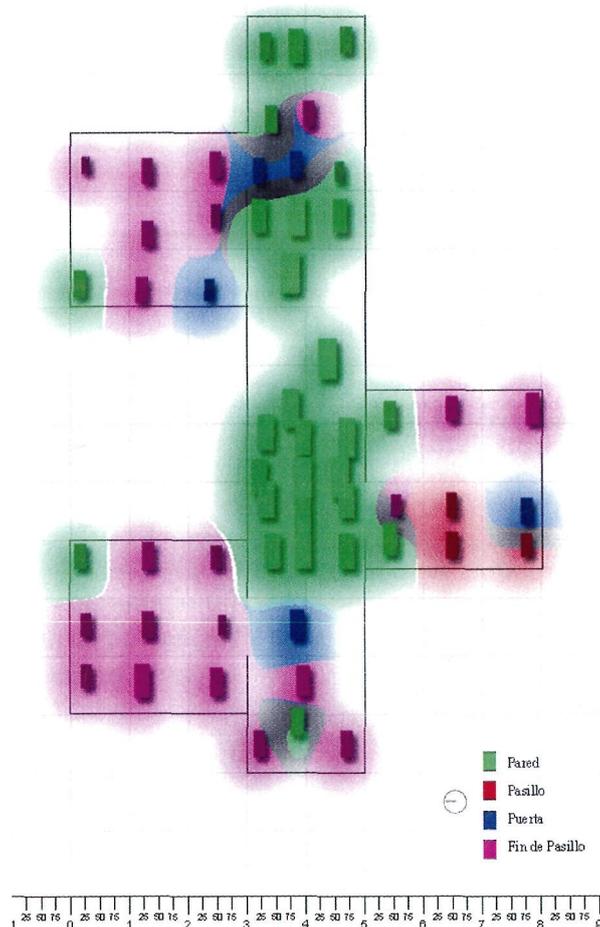


Figura 5.27: Localización del mapa global para 60 medidas (indicadas por el centro de los rectángulos de la figura). La saturación del color y el tamaño de los rectángulos indican la probabilidad de la localización (mayor tamaño mayor probabilidad de localización). El resultado de la localización es pasillo (rojo), pared (verde), puerta (azul), fin de pasillo (magenta). Las localizaciones equiprobables se envuelven con áreas del mismo color. El robot siempre tiene una posición de -90° respecto al mundo (representada en la figura).

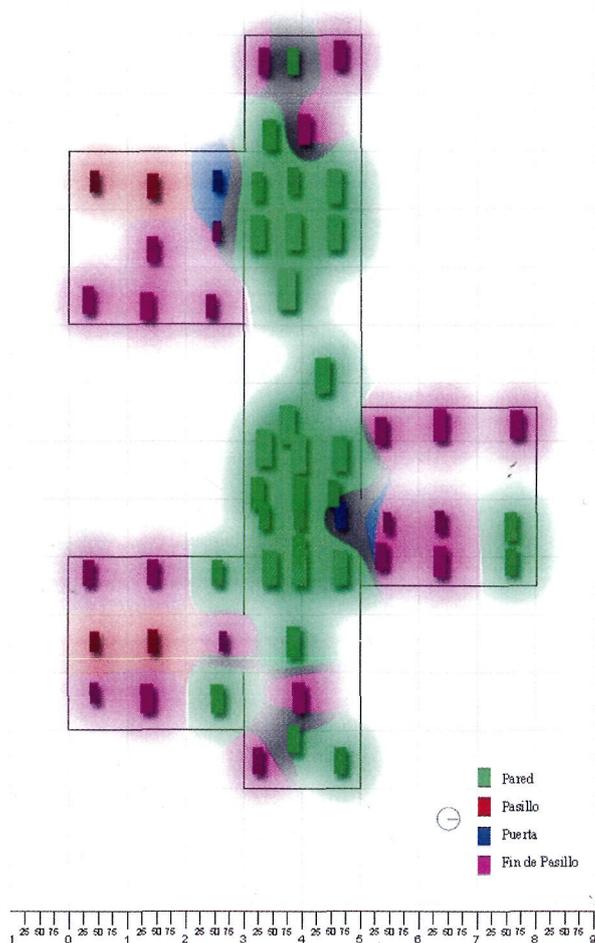


Figura 5.28: Localización del mapa global para 60 medidas (indicadas por el centro de los rectángulos de la figura). La saturación del color y el tamaño de los rectángulos indican la probabilidad de la localización (mayor tamaño mayor probabilidad de localización). El resultado de la localización es pasillo (rojo), pared (verde), puerta (azul), fin de pasillo (magenta). Las localizaciones equiprobables se envuelven con áreas del mismo color. El robot siempre tiene una posición de 90° respecto al mundo (representada en la figura).



**CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA
COMPLEMENTARIA**

171

θ	d_x	V_{rot}
-25	-500	35
0	-500	10
25	-500	-15
-25	-170	30
0	-170	5
25	-170	-20
-25	170	20
0	170	-5
25	170	-30
-25	500	15
0	500	-10
25	500	-35

Figura 5.29: Tabla que especifica la distribución $P(V_{rot}|L \otimes [L' = \langle 0, -170 \rangle])$, (para mejorar la legibilidad se representa únicamente el valor más probable para la variable V_{rot}).

Utilizando el programa anterior conseguimos simplificar sobre manera la especificación de la distribución $P(V_{rot}|L \otimes L')$. Por otra parte la distribución $P(V_{tras}|L \otimes L')$ en este caso se puede asumir uniforme y fijar por tanto la velocidad traslacional a velocidad constante (en nuestro caso se ha fijado a $0.3m/s$).

Submapas de la puerta, pared y fin de pasillo

Al igual que en el caso anterior los submapas restantes necesitan de la definición de $P(V_{rot}|L \otimes L')$ y $P(V_{tras}|L \otimes L')$ para generar comportamientos. El mapa de la puerta se comportará exactamente igual que el mapa del pasillo ignorando las puertas que detecte el robot a mano derecha. El mapa de la pared por otra parte intentará evitar la pared poniéndose paralelo a ella. Ambos utilizarán una velocidad traslacional uniforme de $0.3m/s$. Por otra parte, el mapa de fin de pasillo girará constantemente hacia la derecha hasta evitar la pared frontal que limita su paso. La velocidad traslacional será relativa al ángulo de giro efectuado (a mayor giro menor velocidad).



Ejecución del mapa Bayesiano resultante del proceso de fusión

Una vez especificados completamente los submapas y completado el proceso de fusión podemos utilizar el mapa resultante para ejecutar la conducta diseñada previamente. La conducta tendrá la forma de una pregunta a la distribución de la unión de variables del mapa fusionado $P(A|L' \otimes R)$, es decir, dada la lectura actual y la localización donde deseamos ir, ¿qué acciones tenemos que realizar?

Esta pregunta se puede reescribir en términos de la distribución conjunta de variables del mapa fusionado:

$$\begin{aligned}
 P(A|L' \otimes R) &= \sum_L P(R \otimes L \otimes L' \otimes A) \\
 &= \sum_L \left(\frac{P(L) \times \prod_{i=1}^n P(R_i \otimes L_i \otimes L'_i \otimes A_i|L) \times P(L') \times P(A|L \otimes L')}{P(L') \times P(A|L \otimes L')} \right) \\
 &= \frac{1}{\Sigma} \sum_L \left(\prod_{i=1}^n P(R_i \otimes L_i \otimes L'_i \otimes A_i|L) \right) \times P(A|L \otimes L')
 \end{aligned}$$

Una alternativa simple para su cálculo es localizar al robot en el mundo. Para ello realizaremos la pregunta $P(L_t|R)$ al mapa global obteniendo dada una lectura cuál es la probabilidad de que el robot esté en un pasillo (c_p) una pared (c_a) una puerta (c_u) o un fin de pasillo (c_e) y cómo se sitúa respecto a estos. Una vez obtenida esa probabilidad se obtendrá la acción a ejecutar a partir de la distribución $P(A|L_t \otimes L'_t)$ del mapa global. Esta distribución, que es la que definirá la conducta a realizar, se describe en la figura 5.30 (para mejorar la legibilidad se representa únicamente el valor más probable para la variable A).

Como se puede observar en la figura 5.31 la unidad de localización que guía al robot es capaz de desarrollar la conducta planteada de manera correcta. No obstante existen situaciones que la unidad no es capaz de solventar. Por ejemplo en la ruta verde de la figura 5.31 el robot se introduce en un despacho. Al desplazarse hacia el centro del mismo siempre detecta un fin de pasillo siendo incapaz de salir. Por otra parte en la ruta roja el mapa realiza una detección de fin de pasillo que desemboca en la entrada a un despacho. En la siguiente sección se plantearán soluciones para tratar con estas situaciones especiales.



**CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA
COMPLEMENTARIA**

173

L	L'	A
c_p	c_p	<i>Cp.SeguirPasilloCentrado</i>
c_p	c_a	<i>Cp.SeguirPasilloCentrado</i>
c_p	c_u	<i>Cp.SeguirPasilloCentrado</i>
c_p	c_e	<i>Cp.SeguirPasilloCentrado</i>
c_a	c_p	<i>Ca.SeguirPared</i>
c_a	c_a	<i>Ca.SeguirPared</i>
c_a	c_u	<i>Ca.SeguirPared</i>
c_a	c_e	<i>Ca.SeguirPared</i>
c_u	c_p	<i>Cu.IgnorarPuerta</i>
c_u	c_a	<i>Cu.IgnorarPuerta</i>
c_u	c_u	<i>Cu.IgnorarPuerta</i>
c_u	c_e	<i>Cu.PrepararseAtravesar</i>
c_e	c_p	<i>Ce.Evitar</i>
c_e	c_a	<i>Ce.Evitar</i>
c_e	c_u	<i>Ce.Evitar</i>
c_e	c_e	<i>Ce.AvanzarHastaLimite</i>

Figura 5.30: Tabla que especifica la distribución $P(A|L_t \otimes L'_t)$

Extracción del grafo inducido

Tal y como se describe en la sección 3.4.4 es posible obtener el grafo inducido de un mapa Bayesiano. De esta manera el mapa de localización puede representarse utilizando un grafo, que es la forma en la que tradicionalmente se representan los mapas. En este tipo de representación se resalta la información más importante de las distribuciones de probabilidad, sólo representando los arcos que corresponden a los valores de probabilidad más elevados (ya que de otra manera la mayoría de los grafos serían completos). Esta limitación de la traducción de un mapa Bayesiano a un grafo es la consecuencia de la diferencia de riqueza de expresión entre los formalismos probabilísticos y los grafos.

En la figura 5.32 se propone un grafo inducido a partir del mapa Bayesiano de localización.

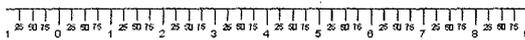
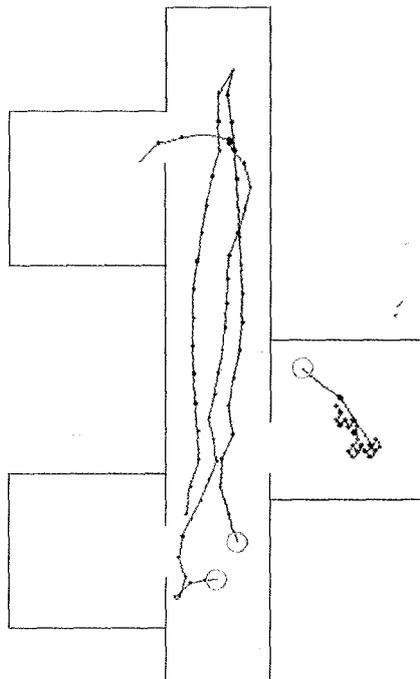


Figura 5.31: Movimiento del robot en el entorno utilizando el mapa fusionado junto con una conducta simple de desplazamiento por el pasillo. Se muestran tres rutas generadas con distintas posiciones iniciales del robot.

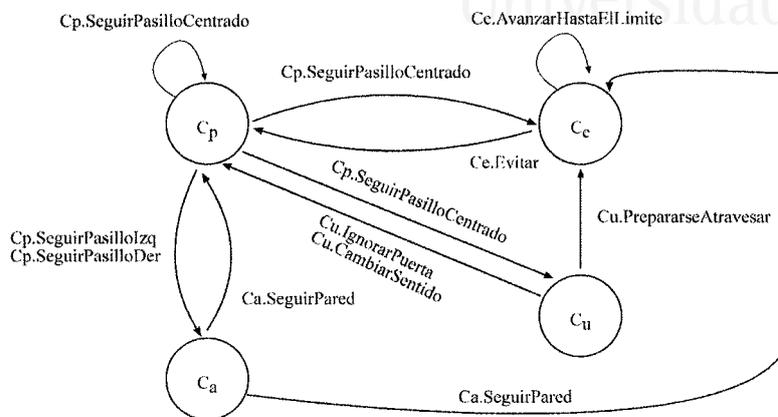


Figura 5.32: Grafo inducido a partir de mapa Bayesiano de localización.

5.4. Conducta con una unidad Bayesiana complementaria

La finalidad principal de un mapa Bayesiano es proporcionar comportamientos. De esta manera el mapa resultante del proceso de fusión, la unidad Bayesiana de localización, proporciona comportamientos al robot para realizar una determinada tarea. No obstante, como se ha comentado anteriormente, aunque la unidad de localización funciona correctamente para la mayoría de las tareas, existen situaciones en las cuales el robot es incapaz de desarrollar las tareas requeridas. Se puede dar el caso en el que para una determinada localización/acción el robot quede atrapado debido a la baja probabilidad del resto de localizaciones, como se puede observar en la ruta verde de la figura 5.31. En este caso el robot detecta un fin de pasillo, lo evita pero se introduce en otro fin de pasillo, y así sucesivamente. Por otra parte, una localización ambigua puede desembocar en que el robot desarrolle una conducta equivocada y colisione con algún obstáculo.

Este tipo de situaciones no son fáciles de evitar añadiendo más submapas a la unidad de localización. Hay que tener en cuenta que cuantos más mapas se añadan mayor probabilidad existe de que el proceso de localización sea incorrecto y se clasifiquen incorrectamente las lecturas en distintos mapas. Por ello que la unidad de localización sólo debe contener las estructuras simples que se repiten en el entorno.



CONDUCTA CON UNA UNIDAD BAYESIANA COMPLEMENTARIA

176

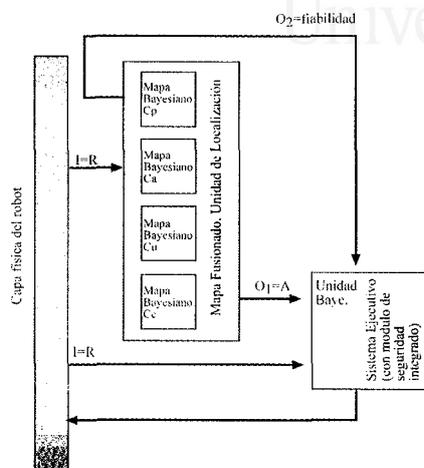


Figura 5.33: Arquitectura del sistema.

No obstante, se puede completar la unidad de localización con sistemas encargados de detectar determinadas situaciones y reaccionar de forma acorde con ellas. Estos sistemas no siempre pueden ser mapas Bayesianos. Un mapa Bayesiano debe proporcionar comportamientos y además tiene que responder de manera informativa a las preguntas de predicción, localización y control. Cabe destacar que en este caso sí que se pueden utilizar unidades Bayesianas, que no tienen esta limitación.

A continuación se va a proponer una unidad Bayesiana encargada de proteger al robot contra colisiones inminentes. Esta unidad se llamará sistema ejecutivo y se encargará de ejecutar las acciones propuestas por el mapa de localización siempre y cuando éstas no pongan en peligro la integridad del robot.

5.4.1. Arquitectura del sistema

Se propone la arquitectura de la figura 5.33 compuesta por dos unidades Bayesianas. Tal y como se dijo anteriormente la unidad de localización será la encargada de representar el espacio y generar comportamientos útiles. Por otra parte la unidad de ejecución obtendrá los datos sensoriales y las salidas de la unidad de localización para retornar la acción a realizar a la capa física del robot.

Como se observa en la figura 5.33 la unidad Bayesiana de ejecución obtiene



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

177

información relativa a la fiabilidad de la unidad de localización (por ejemplo se puede obtener la distribución de una variable y calcular qué porcentaje representa la selección actual respecto al total). Al trabajar con distribuciones de probabilidad es relativamente simple obtener con qué fiabilidad se trabaja en un instante dado y retroalimentar a la unidad de ejecución con esta información. De esta manera la unidad de ejecución dispondrá de mayor información a la hora de determinar qué acciones ejecutar.

Por otra parte conviene resaltar que las acciones desarrolladas por el mapa de localización son comportamientos elaborados del tipo *cp.SeguirPasillo*. La ejecución de estos comportamientos por el robot requiere su descomposición, por parte de los submapas que lo forman, en las acciones básicas $Vrot$ y $Vtras$. Se supone que las acciones que recibe la unidad de ejecución son acciones básicas a ejecutar por el robot.

5.4.2. Unidad de ejecución

Las acciones enviadas al robot se transmiten a través de una unidad Bayesiana. Esta unidad obtiene los datos de los sónares R directamente a través del robot y también las acciones y la fiabilidad de las mismas a través de la unidad de localización. De esta manera la unidad ejecutiva se define utilizando las siguientes variables: $I = \{\rho, \theta, A_{UL}\}$ donde ρ es la mínima distancia detectada por los sensores de ultrasonidos $R = \{R_1, R_2, \dots, R_7\}$ y θ el ángulo del sensor con distancia mínima. A_{UL} es la distribución de variables de salida retornada por la unidad de localización que representa las acciones que serían enviadas al robot. La unidad de ejecución se define para desarrollar un comportamiento altamente reactivo por lo cual el estado S no es necesario. La salida de esta unidad se define como la velocidad rotacional y traslacional que será enviada al robot $O = \{Vrot, Vtras\}$.

Utilizando las variables anteriores, se propone la siguiente descomposición de la unión de variables:

$$\begin{aligned} P(I \otimes S \otimes O) &= P(\rho \otimes \theta \otimes A_{UL} \otimes S \otimes Vrot \otimes Vtras) = \\ &= P(\rho \otimes \theta \otimes A_{UL}) \times P(S|\rho \otimes \theta \otimes A_{UL}) \times \\ &\quad \times P(Vrot \otimes Vtras|\rho \otimes \theta \otimes A_{UL} \otimes S) = \\ &= P(\rho \otimes \theta \otimes A_{UL}) \times P(Vrot \otimes Vtras|\rho \otimes \theta \otimes A_{UL}) \end{aligned}$$

Como se comentó anteriormente, al tratarse de un comportamiento reactivo, la variable S no es útil y $P(S|\rho \otimes \theta \otimes A)$ puede asumirse como una



distribución uniforme. Por otra parte, como no se conoce nada sobre la distribución de los obstáculos en el mundo, $P(\rho \otimes \theta \otimes A_{UL})$ se asume también como uniforme. $P(Vrot \otimes Vtras|\rho \otimes \theta \otimes A_{UL})$ puede ser especificado por el programador o bien en forma de tabla o bien especificarse de forma algorítmica como se vio en la sección anterior. De esta manera, dada la distancia mínima a un objeto, su ángulo y la acción propuesta por la unidad de localización se desea obtener la velocidad rotacional y traslacional a ser ejecutadas por el robot. Esta distribución proporciona al programador un mecanismo para reducir la velocidad del robot en entornos hostiles o incluso realizar una parada de emergencia.

En nuestro caso es conveniente que la distribución $P(Vrot \otimes Vtras|\rho \otimes \theta \otimes A_{UL})$ determine a priori la utilización de A_{UL} o bien, en caso de una posible colisión, genere un comportamiento en reacción al obstáculo que se encuentra en (ρ, θ) . Los comandos que se envían al robot, $A = (Vrot, Vtras)$ se especificarían de la siguiente manera:

$$A = \alpha \cdot A_{UL} + \beta \cdot evitarObs$$

Obviamente es la variable α la que determina el tipo de transición entre ambos comportamientos siendo $\beta = 1 - \alpha$. En nuestro caso se ha utilizado una distribución discreta basada en la función logística $y = \left(1 + e^{10 - \frac{1}{50}x}\right)^{-1}$. Esta distribución deja total libertad a la unidad de localización siempre que no exista ningún obstáculo a menos de medio metro, en caso contrario se prima al sistema de evitación de obstáculos. Además el uso de esta función nos permite conseguir una transición gradual entre ambos comportamientos.

Por otra parte *evitarObs* define la acción a realizar para evitar el objeto definido por (ρ, θ) . En nuestro caso la conducta girará evitando el obstáculo con una intensidad $\beta \times \frac{\pi}{4}$, además la velocidad traslacional quedará reducida sustancialmente definiéndose por $\beta \times 10mm/s$.

5.4.3. Experimentación

El problema principal de utilizar únicamente una unidad de localización es que una localización ambigua puede desembocar en una colisión del robot. Por ejemplo, en la figura 5.34 podemos observar una posición inicial del robot en la cual la localización más intensa provenía del mapa c_u . No obstante el robot además de estar próximo a la pared de la derecha tiene una inclinación que provocará una colisión inminente si no rectifica su trayectoria. Para la

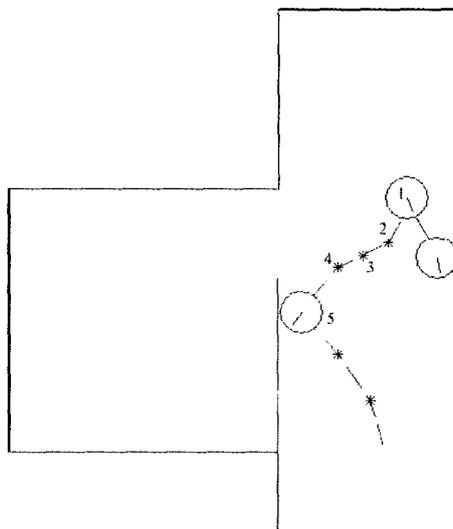


Figura 5.34: Movimiento del robot utilizando la unidad de ejecución (azul discontinuo) y movimiento sin utilizar dicha unidad (rojo). El robot sin la unidad de ejecución cree detectar una puerta continuando recto y colisionando por tanto con la pared derecha.

unidad de localización el robot ha detectado una puerta y debe ignorarla, no pudiendo percibir la inclinación del mismo, lo que desemboca en una colisión del robot.

El uso de la unidad de ejecución permite evitar este tipo de colisiones. Como se puede ver en el punto 1 de la figura 5.34 (trayecto de color azul discontinuo) el robot detecta la proximidad a la pared obteniendo un parámetro $\alpha = 0,018$ (primando por tanto el sistema de evitación de obstáculos que gira evitando la pared). En el hito 2 el robot utiliza únicamente las ordenes de la unidad de ejecución con un parámetro $\alpha = 1$. Por otra parte en el hito 4 el robot se aproxima a la pared izquierda por lo que el sistema de evitación de obstáculos se vuelve a activar con un $\alpha = 0,005$.

Como se puede observar la unidad de ejecución consigue proporcionar más fiabilidad al sistema de localización completándolo en aquellas situaciones en las cuales el robot puede sufrir una colisión. De esta manera se consigue reducir el número de colisiones del sistema y por tanto aumentar la fiabilidad del mismo.



5.5. Detección de incoherencias

En la sección 5.3.5 se vio una aproximación para fusionar cada uno de los sensores de ultrasonidos. Aunque se aportaban algunas ventajas de esta descomposición no se vio la ventaja principal. Al elegir la fusión como un producto de términos más simples relativos a cada sensor disponible en el sistema, se puede obtener la lectura esperada de un sensor de manera sencilla en el caso de un determinado submapa:

$$P(Rx_1^t | Rx_2^t \otimes Rx_3^t \otimes \dots \otimes Rx_7^t) = \frac{1}{\Sigma} \sum_L \prod_i P(Rx_i^t | L) \quad (5.5)$$

De esta manera es posible verificar si el sensor Rx_1 está emitiendo lecturas incorrectas si su lectura en el instante t persiste en ser inconsistente con las lecturas de los otros sensores en un determinado periodo de tiempo. Esta inconsistencia se puede obtener por una baja probabilidad de Rx_1^t .

Como se ha comentado anteriormente, el combinar unidades utilizando el operador de fusión complementaria nos proporciona múltiples ventajas, una de las principales es la emergencia de una mayor robustez en el sistema. Sería pues conveniente proporcionar un mecanismo para combinar también la fiabilidad sensorial. A partir de 5.5 se ha definido un método para calcular la fiabilidad de un determinado sensor en una subunidad de proceso Bayesiana. Ésta depende únicamente del término $P(Rx|L)$. Según la definición del operador de fusión complementaria podemos obtener este término para la unidad de localización global:

$$\begin{aligned} P(Rx|L) &= \sum_{AL'} P(L) \prod_i P(R^i \otimes L^i \otimes L^{i'} \otimes A^i | L) \\ &\times P(L') \times P(A | L \otimes L') = \\ &= P(L) \times \prod_i P(R^i \otimes L^i \otimes L^{i'} \otimes A^i | L) \times \sum_{L'} P(L') \times \\ &\times \sum_A P(A | L \otimes L') = \\ &= \frac{1}{\Sigma} \prod_i P(R^i \otimes L^i \otimes L^{i'} \otimes A^i | L) \end{aligned}$$

Es decir, en la unidad de localización global (la unidad obtenida al aplicar el operador de fusión complementaria a los distintos mapas) la localización depende de la localización de las subunidades así como de las medidas sensoriales y salidas tomadas por las mismas. De esta manera se define la probabilidad de que un sensor de ultrasonidos Rx_1 sea consistente con el resto



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

181

de sensores como:

$$PSF_1^t = P(Rx_1^t | Rx_2^t \otimes Rx_3^t \otimes \dots \otimes Rx_7^t) = \frac{1}{\Sigma} \times \prod_n \left(\begin{array}{l} P(R \otimes L \otimes L' \otimes A | [L = c_1]) \times \frac{1}{\Sigma_1} \times \sum_{L=c_1} \prod_i P(R_i | L) \times \\ P(R \otimes L \otimes L' \otimes A | [L = c_2]) \times \frac{1}{\Sigma_2} \times \sum_{L=c_2} \prod_i P(R_i | L) \times \\ \vdots \\ P(R \otimes L \otimes L' \otimes A | [L = c_n]) \times \frac{1}{\Sigma_n} \times \sum_{L=c_n} \prod_i P(R_i | L) \end{array} \right) \quad (5.6)$$

El cálculo de la probabilidad anterior se puede interpretar como una comparación Bayesiana de la relevancia de los distintos mapas respecto a la probabilidad de fallos de los mismos.

La probabilidad PSF_1^t se puede utilizar para detectar a un sensor incorrecto, descartar sus lecturas y continuar utilizando los sensores restantes. Para ello habrá que establecer un determinado umbral a partir del cual se considerará que una lectura es incorrecta y tendrá que ser descartada. En nuestro caso este umbral se define como:

$$fallo_{P_1} = \left(\frac{\sum_i P(R_i | S)}{i} - P(R_1 | R_2 \otimes R_3 \otimes \dots \otimes R_7) + \mu \right) < 0 \quad (5.7)$$

donde $S \subseteq \{Rx \setminus R_i\}$

De manera que la media de error por sónar determina, a parte del umbral (proporcionado por el programador), la fiabilidad para un determinado sensor. Así pues, en entornos donde la incertidumbre es muy elevada (por ejemplo en entornos para los cuales nuestro modelo no está preparado para funcionar), no se darán fallos inexistentes ni se dará el caso de que el robot piense que todas sus lecturas son incorrectas.

5.5.1. Validación Experimental

Para comprobar si el modelo actual puede detectar y corregir el funcionamiento de un sónar incorrecto se ha introducido ruido en el sensor lateral izquierdo del robot (ver figura 5.2) falseando los datos con números aleatorios

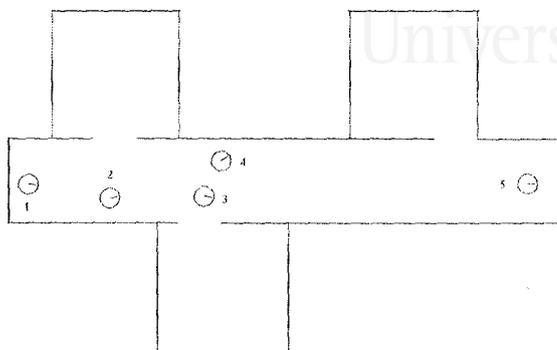


Figura 5.35: Posiciones seleccionadas para realizar la validación experimental. Se ha intentado escoger posiciones que provoquen la elección de cada uno de los mapas descritos anteriormente. En cada posición se obtuvieron 30 lecturas a partir de las cuales se calculó la fiabilidad del sistema.

uniformes. En la figura 5.35 se muestran las posiciones del robot en una serie de situaciones seleccionadas.

Es interesante analizar los resultados mostrados en la figura 5.37 donde podemos observar la fiabilidad de cada sensor en un momento determinado.

Por una parte podemos observar cómo la fiabilidad sensorial para un determinado hito² varía dependiendo del submapa que utilizemos. Esto es debido a que cada mapa espera unas lecturas determinadas y por tanto las medidas esperadas para un determinado sensor, dado el resto, pueden no ser totalmente acertadas. Una vez obtenida la fiabilidad de cada mapa aplicando la ecuación 5.6 podemos combinar dichas fiabilidades para obtener la fiabilidad común para el hito n . Si se recorre la sexta fila (la fiabilidad global) se puede observar como para el sensor 1 existe una fiabilidad muy baja (próxima a 0) comparada con la fiabilidad del resto de sensores, viéndose así que es muy probable que el sensor 1 funcione de manera incorrecta. Sería ya cuestión del programador el determinar el umbral μ a partir del cual no se tendrían en cuenta las lecturas del sensor erróneo.

²Para este ejemplo concreto se ha mostrado la fiabilidad en cada hito, sin tener en cuenta las fiabilidades anteriores, para facilitar la comparación de datos.



**CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA
COMPLEMENTARIA**

Universitat d'Alacant
Universidad de Alicante

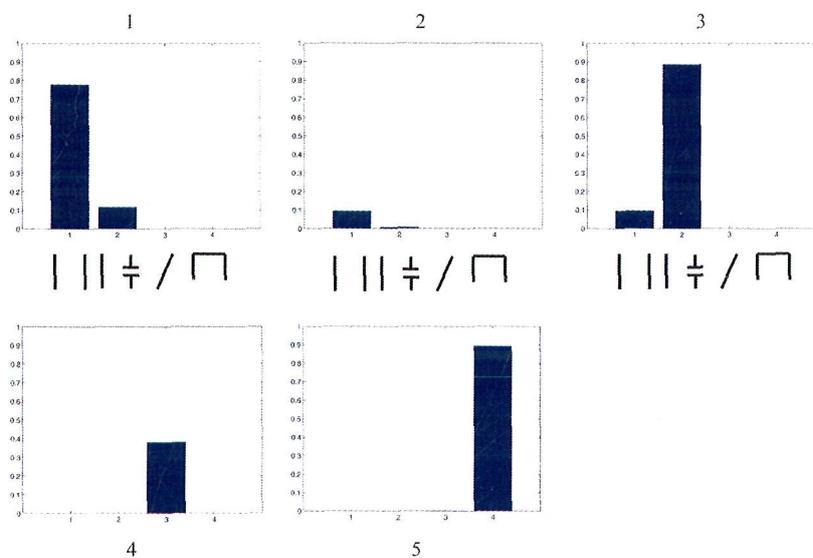


Figura 5.36: $P(L|Rx)$ para cada punto clave (representado en la figura 5.35) y cada unidad i (una unidad viene representada por su símbolo: pasillo, puerta, pared y fin de pasillo).

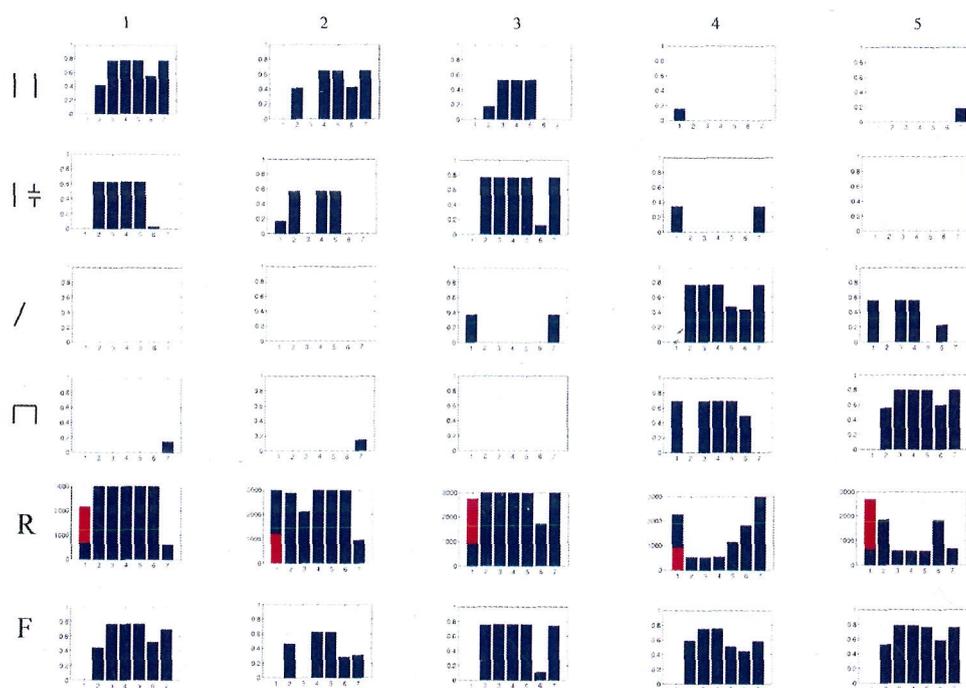


Figura 5.37: Para cada s3onar se calcula la fiabilidad $P(Rx_1^t | Rx_2^t \otimes Rx_3^t \otimes \dots \otimes Rx_7^t)$ seg3un cada subunidad y cada punto clave. Se muestra tambi3en la fiabilidad de la unidad global (F) en la 3ultima fila de la tabla. Para realizar las pruebas se contaminaron las lecturas del primer sensor aleatoriamente (fila R). La barra roja (la m3as clara de las dos) especifica el valor aleatorio utilizado y la barra azul el valor real que corresponder3a a la medida de dicho sensor.



5.6. Comparación del sistema presentado con un sistema de control difuso

En esta sección se presentará un sistema de control difuso para la evitación de obstáculos desarrollado para el robot Pioneer I. Una vez probado el funcionamiento del sistema se introducirán medidas erróneas en los dos sonares laterales del robot y se comparará el comportamiento del sistema difuso frente al sistema Bayesiano.

5.6.1. Sistema de control difuso

La lógica difusa es un método para el razonamiento con expresiones lógicas que describen las pertenencias a los conjuntos difusos (ver [Zad65], [Var99], [Zim01]). El control difuso es una metodología para la construcción de sistemas de control en los que las correspondencias entre las entradas y los parámetros de salida están representadas por reglas difusas. Este tipo de sistemas de control ha tenido éxito en su implantación a sistemas robóticos, productos industriales. . .

Un sistema de control difuso requiere la definición de conjuntos difusos y reglas difusas. El funcionamiento, a grandes rasgos, de este tipo de sistemas se puede dividir en los siguientes pasos:

- Se comparan los valores numéricos de entrada al sistema con las funciones de pertenencia asociadas a los términos lingüísticos de la parte del antecedente de la regla asociada a esa entrada. Se calcula el grado de pertenencia a dichos términos. A este paso se le conoce como *fuzzyficación*.
- Se combinan, mediante un operador T-norma, los grados de pertenencia a cada término asociado con el antecedente de la regla para determinar el grado de cumplimiento de la misma.
- Se genera un consecuente para cada regla en función del grado de cumplimiento de cada una.
- Se obtiene una salida numérica a partir de todos los consecuentes obtenidos. A este paso se le conoce como *defuzzyficación*.

Para la tarea de evitación de obstáculos requerida se han definido 9 conjuntos difusos, 7 de los cuales representan variables de entrada (las lecturas

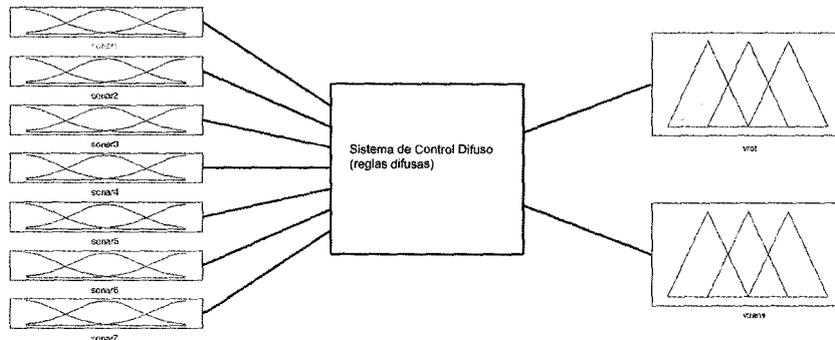

 COMPARACIÓN DEL SISTEMA PRESENTADO CON UN
 SISTEMA DE CONTROL DIFUSO


Figura 5.38: Estructura del sistema de control difuso presentado.

de los s3nars de ultrasonidos), siendo los dos restantes las salidas enviadas a los actuadores ($Vrot$ y $Vtras$). Ver figura 5.38

Las variables de entrada $sonar_{1,...,7}$ tienen como conjunto de t3rminos asociados (valores lingüísticos) que un obst3culo se encuentre *muy cerca*, *cerca*, *medio* o *lejos*, movi3ndose en un universo del discurso de 0 a 3000 mil3metros. En la figura 5.39 se muestran de manera gr3fica las funciones de pertenencia para cada una de estas variables.

Por otra parte la variable de salida $Vrot$ dispone de los siguientes valores lingüísticos: *Giro intenso a la izquierda*, *Giro a la izquierda*, *Pequeño giro a la izquierda*, *Centro*, *Pequeño giro a la derecha*, *Giro a la derecha*, *Giro intenso a la derecha*, *Dar la vuelta*. Esta variable se mueve en el universo del discurso de -180° a 180° . En la figura 5.40 podemos ver las funciones de pertenencia de cada uno de sus valores lingüísticos. La variable $Vtras$ requiere los siguientes valores lingüísticos: *velocidad baja*, *velocidad media*, *velocidad alta*. Para esta variable se utiliza un universo del discurso de 0 a 3m/s. Sus funciones de pertenencia se pueden ver en la figura 5.41.

Un sistema de control difuso tambi3n requiere de reglas que determinen salidas a partir de las entradas obtenidas. Para este sistema de evitaci3n de obst3culos se proponen las reglas de la figura 5.42.

Se ha desarrollado la ruta de la figura 5.43, utilizando este sistema de control difuso, donde se puede observar la correcci3n del mismo. El robot entra en la primera habitaci3n de la izquierda, evitando las paredes y volviendo en ruta contraria hasta donde empez3, donde es capaz de evitar el fin de pasillo girando 180° .



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

187

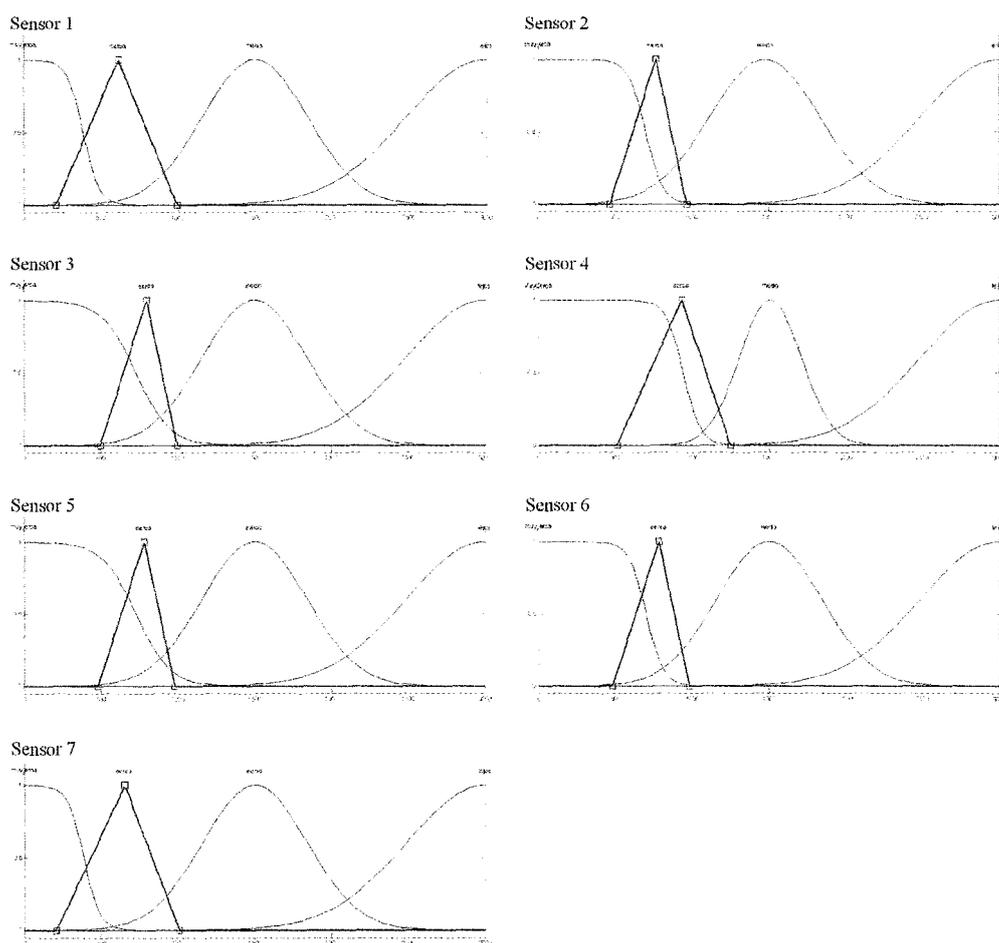


Figura 5.39: Funciones de pertenencia para cada una de las variables de entrada. Los valores lingüísticos de esta variable son (de izquierda a derecha): *muy cerca, cerca, medio o lejos*

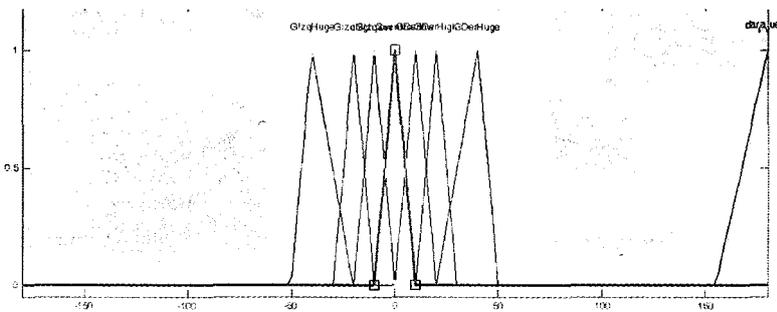


Figura 5.40: Funciones de pertenencia para la variable V_{rot} . Los valores lingüísticos de esta variable son (de izquierda a derecha): *Giro intenso a la izquierda*, *Giro a la izquierda*, *Pequeño giro a la izquierda*, *Centro*, *Pequeño giro a la derecha*, *Giro a la derecha*, *Giro intenso a la derecha*, *Dar la vuelta*

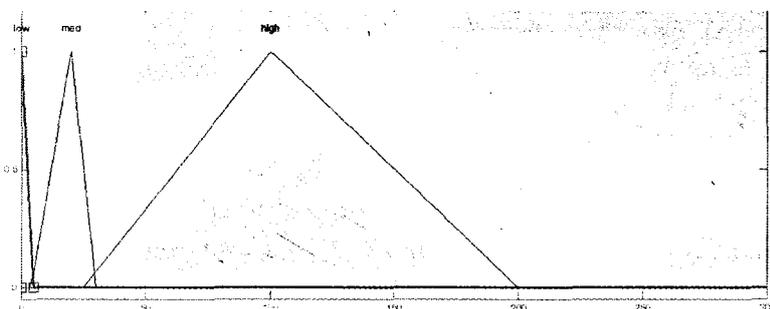


Figura 5.41: Funciones de pertenencia para la variable V_{tras} . Los valores lingüísticos de esta variable son (de izquierda a derecha): *velocidad baja*, *velocidad media*, *velocidad alta*



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

189

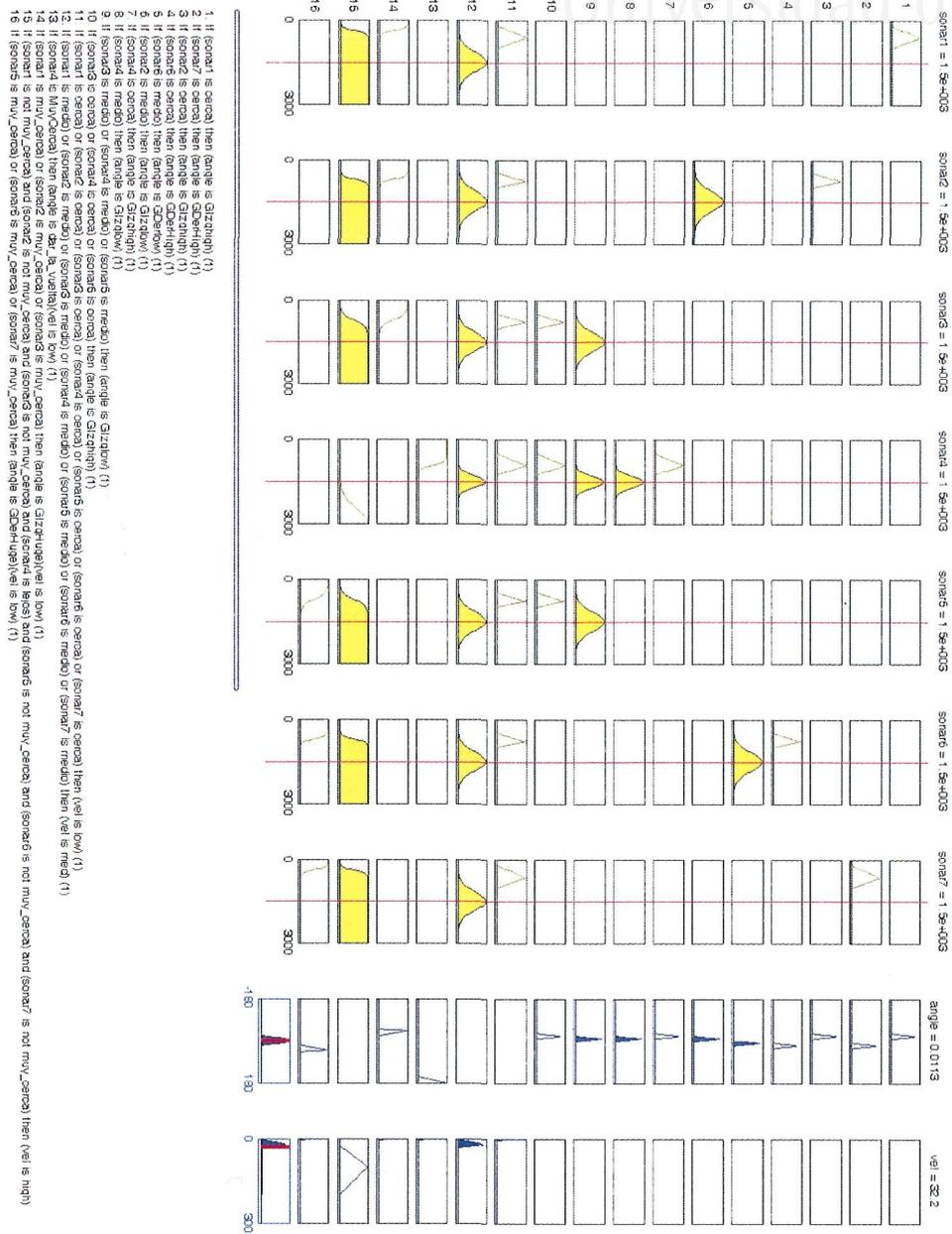


Figura 5.42: Reglas del sistema de control difuso.



COMPARACIÓN DEL SISTEMA PRESENTADO CON UN SISTEMA DE CONTROL DIFUSO

190

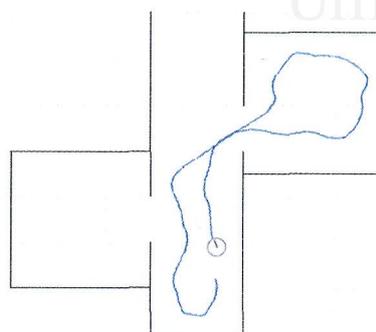


Figura 5.43: Ruta desarrollada por el agente autónomo utilizando el sistema de control difuso propuesto.

Simulación de fallos en el sistema sensorial

Como se ha visto anteriormente tanto el sistema aquí propuesto como el sistema de control difuso funcionan correctamente. No obstante puede ser conveniente compararlos cuando alguno de los sensores del robot falle (emita lecturas que no tengan nada que ver con el entorno por el que navega). De esta manera se ha introducido ruido uniforme en los dos sensores laterales del robot (ver figura 5.2) y se ha comprobado el funcionamiento de ambos sistemas.

El sistema difuso, aunque trata de manera correcta la incertidumbre, no es capaz de detectar ni de eliminar las lecturas incorrectas de los sensores laterales y por tanto la trayectoria que desarrolla el robot no es coherente (ver figura 5.44).

Por otra parte, utilizando tanto la unidad de localización como la unidad de ejecución y combinando estas dos con el sistema detección de incoherencias podemos observar en la figura 5.45 como el robot es capaz de detectar como ambos sensores laterales emiten lecturas que no son coherentes con el resto. De esta manera, cuando la fiabilidad de estos sensores desciende de un umbral (en nuestro caso de 0,05) sus medidas no se tienen en cuenta. Así se consigue un sistema mucho más robusto, capaz de detectar y evitar fallos sensoriales.

Como hemos visto, aunque tanto el sistema de control difuso como el sis-



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

191

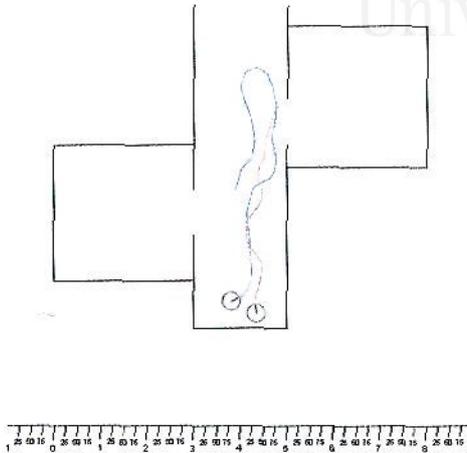


Figura 5.44: Dos rutas desarrolladas por el robot utilizando el sistema de control difuso. Se puede observar como el robot desarrolla comportamientos sin sentido (giros en medio del pasillo, una colisión frente a una puerta...) debido a los errores provenientes de las lecturas de los sensores laterales.

tema Bayesiano presentado se comportan de manera satisfactoria bajo condiciones normales no ocurre lo mismo cuando el sistema sensorial falla. El sistema Bayesiano es capaz de detectar dichos fallos y actuar en consecuencia, anulando las lecturas de los sónares que no trabajan de manera coherente con el resto.

5.7. Resumen

En este capítulo hemos visto un ejemplo de aplicación del operador de fusión complementaria, donde se ha fusionado información proveniente de distintos sensores del mismo tipo. Más concretamente se ha utilizado un anillo de sensores de ultrasonidos. Para el desarrollo de las tareas de fusión se ha utilizado una plataforma de desarrollo donde se han probado de manera realista los modelos propuestos para la realización de las pruebas.

Inicialmente se ha definido una tarea de actuación para el robot suponiendo que este se moverá en un pasillo del Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante. El agente robótico ha servido de conexión entre los distintos despachos pudiéndose

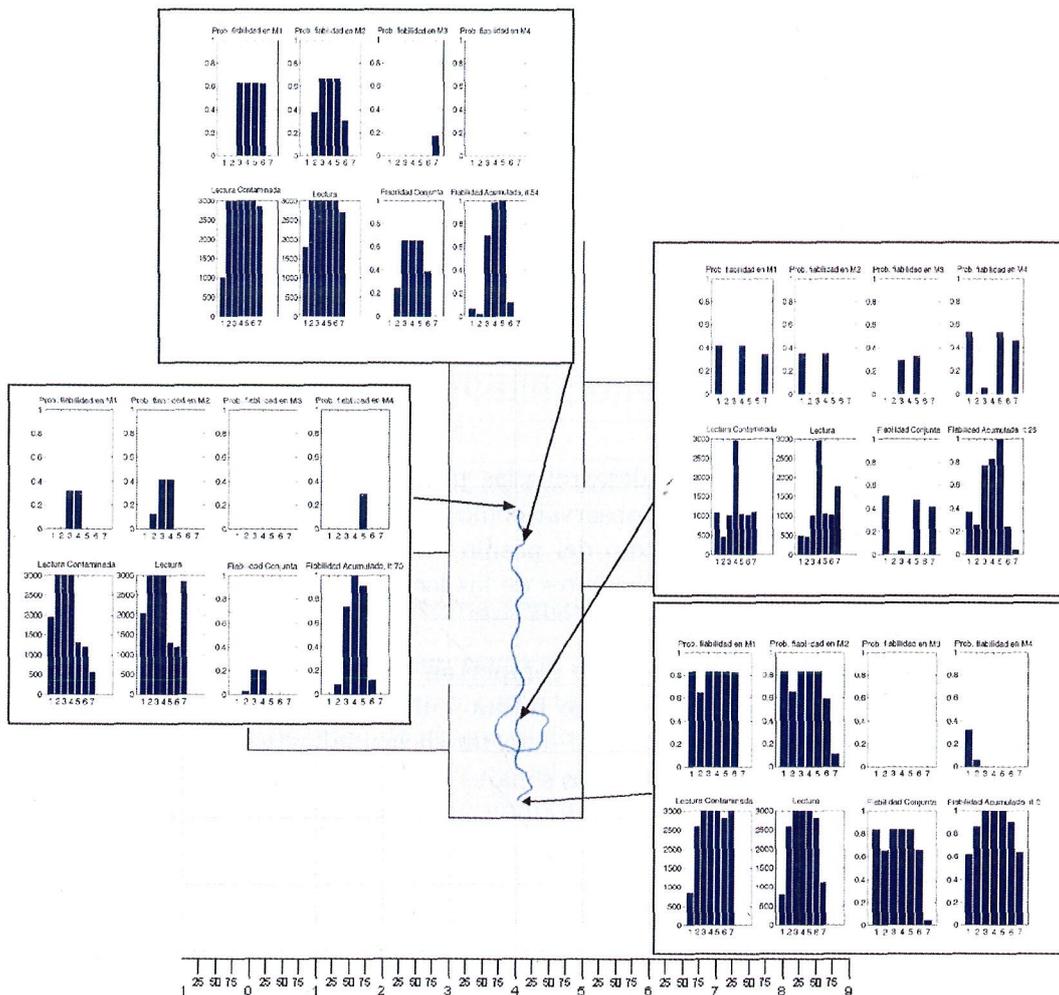


Figura 5.45: Ruta desarrollada por el robot utilizando el sistema Bayesiano. Se muestran para 4 hitos distintos las lecturas del s3onar de ultrasonidos, las lecturas contaminadas con ruido, la fiabilidad de un sensor en un mapa determinado, la fiabilidad global y la fiabilidad acumulada. Se puede observar como conforme pasan las iteraciones la fiabilidad de los sensores 1 y 7 decrece hasta que en la 3ltima iteraci3n estos sensores no se tienen en cuenta para el funcionamiento global del sistema.



CAPÍTULO 5. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPLEMENTARIA

193

desplazar de un despacho a otro. Esta tarea se ha dividido convenientemente en cuatro subtareas. Para cada una de ellas se ha proporcionado un mapa Bayesiano que se ha encargado de desarrollar las labores iniciales de fusión de la información. Más concretamente se ha aportado un mapa Bayesiano del pasillo, que representa el estado del robot cuando se encuentra en un pasillo y tiene que avanzar a través de él, un mapa Bayesiano de la puerta, donde el robot reconoce una puerta a su derecha y tiene opción de disponerse a atravesarla o incluso ignorarla, un mapa de final del pasillo, donde el robot debe girar 180° y por último un mapa Bayesiano de la pared, donde se determinan las acciones a realizar cuando el robot sólo es capaz de localizar una pared.

Se ha demostrado que un mapa Bayesiano es un tipo especial de unidad Bayesiana, y por tanto de esta manera los operadores de fusión definidos para las unidades Bayesinas también se pueden aplicar a este tipo de mapas. Ya definidos los cuatro mapas anteriores se ha procedido a su fusión aplicando uno de los operadores de fusión complementaria definidos en los capítulos anteriores. De esta manera se ha obtenido un nuevo mapa que aglutina el conocimiento de los submapas que lo forman. Además se ha proporcionado una representación del mapa de localización resultante del proceso de fusión utilizando un grafo inducido, que es la forma en la que tradicionalmente se representan los mapas.

Se ha comprobado el funcionamiento del mapa, también llamado unidad de localización, definiendo un ejemplo de conducta simple. Aunque los mapas Bayesianos (la unidad de localización) funcionan correctamente para la mayoría de las tareas, existen situaciones en las cuales el robot es incapaz de desenvolverse de manera correcta. Para aliviar este problema se ha proporcionado una nueva unidad que complementa a los mapas Bayesianos. De esta manera se utilizan dos unidades Bayesianas, la primera será la encargada de representar el espacio y generar comportamientos útiles (la unidad de localización), la segunda obtendrá los datos sensoriales y las salidas de la unidad de localización para retornar la acción a realizar por la capa física del robot (la unidad de ejecución). Esta última unidad proporciona un sistema más fiable completando a los mapas Bayesianos (la unidad de localización) en aquellas situaciones en las cuales el robot puede sufrir, por ejemplo, una colisión.

Se ha aportado un mecanismo de detección de incoherencias utilizando el mapa Bayesiano propuesto. Al definir la fusión de sensores como un producto de términos más simples, relativos a cada sensor, se obtiene de manera sencilla la lectura esperada de un determinado sensor. De esta manera se puede



verificar si un determinado sensor está emitiendo lecturas incorrectas.

Por último se ha comparado el sistema Bayesiano con un sistema de control basado en lógica difusa. Aunque ambos sistemas se comportan de manera satisfactoria en condiciones normales no ocurre lo mismo cuando hay fallos en el sistema sensorial. El sistema Bayesiano aportado es capaz de detectar dichos fallos y actuar en consecuencia, anulando las lecturas de los sónares que no trabajan de manera coherente con el resto.

En el próximo capítulo se proporcionará el operador de fusión competitiva. Este operador de fusión junto al complementario y las unidades Bayesianas, definen el modelo FSR-BAY.



Universitat d'Alacant
Universidad de Alicante

“En teoría, no existe diferencia entre teoría y práctica; en la práctica sí la hay”

JAN L.A. VAN DE SNEPSCHEUT

Capítulo 6

Caso de estudio: Fusión Bayesiana Competitiva

En este capítulo se proporcionan dos aplicaciones de fusión competitiva, una que fusiona la información proveniente de los sonares de ultrasonidos con una cámara de vídeo y la otra que combina medidas láser con visión omnidireccional.

Para la primera aplicación dispondremos de un robot con una cámara de vídeo y un anillo de sensores de ultrasonidos. Dos unidades serán las encargadas de manejar el robot, una para cada tipo de sensor. Se trata de unidades que, aunque proporcionan respuestas muy rápidas, son flexibles y por tanto obtienen previamente a su utilización (en una fase de aprendizaje) información del entorno donde realizarán su tarea, proporcionando de esta manera una mayor versatilidad y adaptabilidad al sistema. El agente autónomo utilizará el sensor de ultrasonidos para evitar los obstáculos del entorno. A partir de este sensor aprenderá a realizar la misma tarea de evitación de obstáculos utilizando únicamente la cámara de vídeo, gracias al proceso de fusión de ambas unidades mediante el operador de fusión competitivo.

Por otra parte, la reconstrucción de mapas y la localización de un robot son dos problemas esenciales en el campo de la robótica y la inteligencia artificial. Un robot puede necesitar un modelo del entorno, que puede ser incompleto, y por tanto debe trabajar teniendo en cuenta la incertidumbre. Las unidades Bayesianas tienen en cuenta la

incertidumbre y permiten fusionar información procedente de distintos sensores. En este capítulo se aporta un sistema de reconstrucción de mapas en 3D basado en unidades Bayesianas. La reconstrucción se lleva a cabo integrando la información obtenida por un láser y un sistema de omnivisión. Además, para mejorar la calidad de la reconstrucción, se define la fusión de varias unidades Bayesianas utilizando un operador de fusión competitiva. Finalmente se comentarán los resultados obtenidos así como la validez de la aproximación planteada.

6.1. Aprendiendo a ver con unidades Bayesianas

En esta sección se proporcionará un ejemplo de cómo se pueden fusionar unidades Bayesianas utilizando un operador de fusión competitiva. Aplicando este operador de fusión se aportará, entre otras características, más robustez al sistema. Revisando la independencia condicional del operador de fusión además seremos capaces de *enseñar* al robot a utilizar otro tipo de sensores sobre los cuales no se tiene información a priori sobre su funcionamiento.

Inicialmente dispondremos de un robot con una cámara de video y un anillo de sensores de ultrasonidos. Dos unidades serán las encargadas de manejar el robot, una para cada tipo de sensor. Se trata de unidades que proporcionan respuestas muy rápidas y flexibles. Estas requieren de una fase de aprendizaje, obteniendo información del entorno donde realizarán su tarea, y proporcionando así una mayor versatilidad y adaptabilidad al sistema.

En este caso concreto, el agente autónomo utilizará sus sensores de ultrasonidos para evitar obstáculos y además, a partir de estos sensores, aprenderá a realizar la misma tarea utilizando únicamente una cámara de video.

6.1.1. Descripción del entorno utilizado

Para comprobar el comportamiento de las unidades y operadores aquí presentados es necesario disponer de un entorno de pruebas. Tal y como se comentó anteriormente se ha implementado un Toolbox de Matlab que permite interactuar desde esta aplicación con el software de control robótico Saphira. De esta manera, utilizando este Toolbox, podemos interactuar directamente con un robot real o utilizar un simulador. Para las pruebas relativas



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

197

a esta sección se han realizado pruebas tanto con el robot real como con el simulador Pioneer. A continuación se describirán ambas plataformas.

Robot Pioneer 3DX

El robot Pioneer 3DX, pertenece a una familia de robots de dos y cuatro ruedas entre los cuales se incluyen los robots Pioneer AT, Pioneer II DX, Pioneer II DX8... Estos pequeños robots están diseñados para desarrollar tareas de investigación que compartan una misma arquitectura e interfaz. El tamaño de este robot lo hace particularmente apropiado para tareas de navegación en espacios de interior reales, como aulas de clases, laboratorios y oficinas. Este robot está dotado con dos anillos de sónares de ultrasonidos: uno frontal con ocho sensores y otro trasero también con ocho sensores. El robot Pioneer 3DX dispone de un sistema de motores de alta velocidad y par elevado. Además estos motores están equipados con un sistema odométrico avanzado que permite minimizar los errores relativos a la posición del robot. Tanto la disposición de los sónares de ultrasonidos como la forma y medidas del robot se pueden observar en la figura 6.1.

Sobre el robot se montará una cámara de baja resolución, concretamente la cámara Benq DC1300, apropiada por sus capacidades de video streaming a 15f/s con una resolución de 640×480 píxeles.

Las aplicaciones desarrolladas se han ejecutado en un PC portátil, situado en la parte trasera del robot. Este PC controla el robot (mandándole comandos a través del puente Matlab-Saphira) utilizando el puerto RS-232. A la vez se obtienen imágenes de la cámara en tiempo real, conectada a uno de sus puertos USB (estas imágenes pueden capturarse de manera sencilla desde Matlab utilizando el Toolbox "Image acquisition" proporcionado por Mathworks).

Simulador Saphira y módulo de simulación visual

Como se comentó anteriormente el fabricante de los robots, ActivMedia, proporciona una aplicación de control capaz de conectar con cualquiera de sus robots. Para completar esta aplicación además provee un simulador realista del robot Pioneer 3DX. No obstante este simulador sólo es capaz de trabajar con datos odométricos y con sensores de rango, es decir, no proporciona ningún mecanismo para la integración de un sensor visual.

En nuestro caso era necesario disponer de una cámara en el simulador

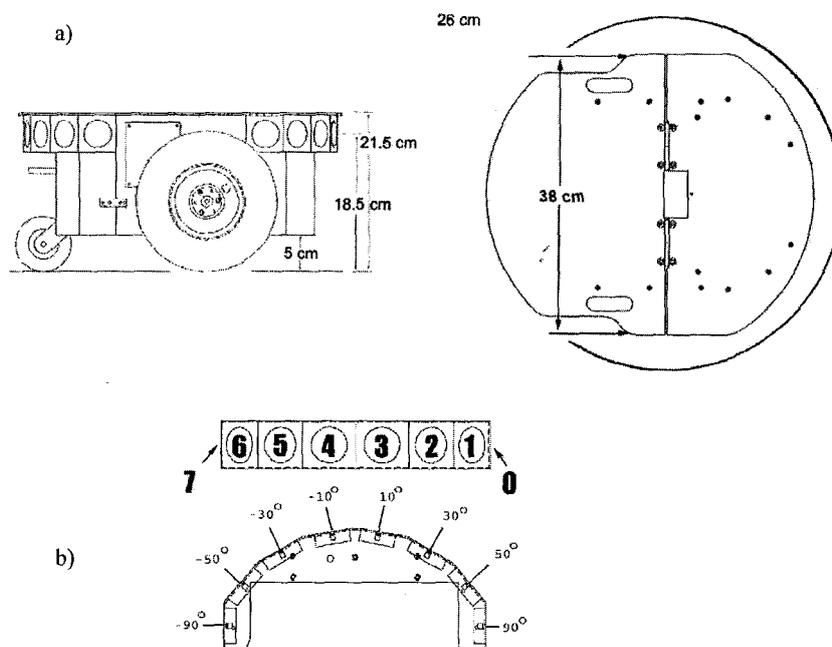


Figura 6.1: a) Forma y tamaño del robot Pioneer 3DX. b) Disposición del anillo de sónares frontal, el trasero es simétrico.



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

199

que nos devolviera una imagen del entorno. El uso de un simulador permite ahorrarse mucho tiempo a la hora de ajustar los parámetros relativos al funcionamiento del sistema propuesto.

Se ha desarrollado un módulo complementario al Toolbox de conexión Matlab-Saphira que permite, mediante la construcción de un mundo en 3D (utilizando el lenguaje de modelado VRML), obtener una imagen de la posición del robot en un momento determinado. De esta manera es posible aprovechar las características existentes en el simulador de ActivMedia completándolas con un sensor visual. Al permitir diseñar los mundos con un lenguaje como VRML, es relativamente sencillo proporcionar entornos realistas para la realización de pruebas con el robot.

6.1.2. Unidad de rango para evitar obstáculos (u_1)

Necesitamos disponer de una unidad reactiva que sea capaz de evitar obstáculos recorriendo el entorno. Para ello esta unidad utilizará únicamente cinco sensores frontales del robot Pioneer 3 DX. Los sensores laterales aunque son útiles para tareas de mapeado no aportan demasiada información para el proceso de evitación de obstáculos que vamos a desarrollar. De esta manera esta unidad dispondrá de las siguientes variables:

$$\begin{aligned}
 (I, O) &= \begin{cases} I : R_1, R_2, \dots, R_6 \\ O : Vrot, Vtras \end{cases} \\
 R &= \{L, M, C\} = \{Lejos, Medio, Cerca\}, |R| = 3 \\
 Vrot &= \{HI, I, C, D, HD\} = \left\{ \begin{array}{l} Izquierda Fuerte, Izquierda, \\ Centro, Derecha, Derecha Fuerte \end{array} \right\}, \\
 |Vrot| &= 5 \\
 Vtras &= \{VL, L, M, H\} = \{MuyBaja, Baja, Media, Alta\}, \\
 |Vtras| &= 3
 \end{aligned}$$

Siendo R_i la distancia al obstáculo más cercano detectado por el s3nar i . $Vrot$ y $Vtras$ representan la velocidad rotacional y traslacional respectivamente y quedan discretizadas en 5 y 3 zonas respectivamente. Como se puede observar hemos discretizado tanto las variables de entrada como las variables de salida, de esta manera disponemos de un modelo m3s compacto y por otra, suficientemente rico para la tarea de evitaci3n de obst3culos que vamos a desarrollar. Utilizando estas variables proponemos la siguiente descomposici3n:



$$P(I \otimes O) = P(I) \times P(O|I)$$

Donde asumimos que $P(I)$ es uniforme ya que suponemos que todas las lecturas inicialmente tienen la misma probabilidad de surgir al desconocer la forma del mundo. De esta manera debemos identificar el término $P(O|I)$. Este término nos indica dada una entrada cuál es la salida esperada por el robot. Se ha considerado que este término se puede representar de manera sencilla utilizando una tabla. Como se comentó en capítulos anteriores hay varias maneras de obtener el valor de esta distribución, se pueden aprender los valores de la salida O a partir de recorridos guiados del robot o se puede definir la distribución por el programador. En este ejemplo se ha definido de manera discreta la distribución $P(O|I)$, asumiendo que dada una entrada sólo se preferirá una única salida. A continuación se muestra un fragmento de esta distribución (para mejorar la legibilidad se representa únicamente el valor más probable para la variable O):

	R_1	R_2	R_3	R_4	R_5	R_6	V_{rot}	V_{tras}
	L	L	L	L	L	L	C	H
	L	L	L	L	L	M	C	H
	L	L	L	L	L	C	I	M
	L	L	L	L	M	L	C	H
	L	L	L	L	M	M	C	M
$P(O I) \approx$	L	L	L	L	M	C	HI	M
	L	L	L	L	C	L	I	M
	L	L	L	L	C	M	HI	M
	L	L	L	L	C	C	HI	L
	L	L	L	M	L	L	C	M
	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
	C	C	C	C	C	C	HI	VL

El uso de la tabla anterior es sencillo. Inicialmente se discretizan las variables de entrada, obteniendo por separado la probabilidad de cada R_i . Una vez obtenida esta probabilidad se calcula la distribución de salida referente a las variables V_{rot} y V_{tras} . De esta distribución se extrae al azar una de las posibles salidas (obviamente cuanta más probabilidad tiene una salida más posibilidad tiene de ser elegida).

De esta manera obteniendo la entrada I en un momento determinado podemos realizar la pregunta $P(O|I)$ a la distribución conjunta para obtener

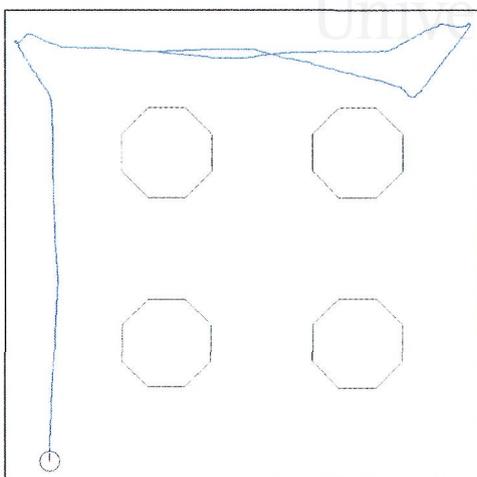


Figura 6.2: Ruta recorrida por el robot utilizando la unidad de evitación de obstáculos por ultrasonidos.

los comandos que serán enviados a los actuadores del robot. En la figura 6.2 se muestra un ejemplo de ruta, en el mundo virtual sobre el que se realizarán las pruebas, utilizando esta unidad.

6.1.3. Unidad visual para evitar obstáculos (u_2)

De la misma manera que en el caso anterior, deseamos realizar una tarea de evitación de obstáculos pero esta vez utilizando como único sensor una cámara de video. En una imagen existen multitud de datos que no aportan información para la tarea a desarrollar. En nuestro caso hemos discretizado la imagen de entrada únicamente en cinco zonas. Para cada zona se ha obtenido la media del color que la representa. La discretización utilizada se muestra en la figura 6.3.

De esta manera disponemos de cinco variables de entrada, una para cada zona:

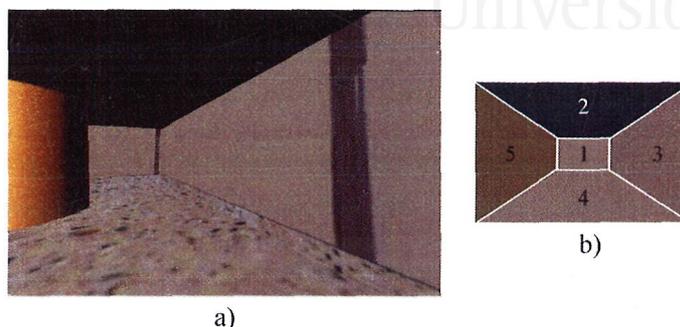


Figura 6.3: a) Ejemplo de imagen obtenida mediante la cámara virtual. b) Discretización de esta imagen en las 5 zonas utilizadas por la unidad de proceso.

$$(I, O) = \begin{cases} I : C_1, C_2, C_3, C_4, C_5 \\ O : Vrot, Vtras \end{cases}$$

$$C = \{0, 1, \dots, 18\}, |C| = 18$$

$$Vrot = \{HI, I, C, D, HD\} = \left\{ \begin{array}{l} Izquierda Fuerte, Izquierda, \\ Centro, Derecha, Derecha Fuerte \end{array} \right\},$$

$$|Vrot| = 5$$

$$Vtras = \{VL, L, M, H\} = \{MuyBaja, Baja, Media, Alta\},$$

$$|Vtras| = 3$$

Siendo $Vrot$ y $Vtras$ la velocidad rotacional y traslacional respectivamente. Al igual que en el caso anterior hemos discretizado tanto las variables de entrada (donde C_i es la intensidad de una determinada zona i) como las variables de salida. Utilizando estas variables proponemos la misma descomposición:

$$P(I \otimes O) = P(I) \times P(O|I)$$

Donde $P(I)$ suponemos que a priori es uniforme ya que todas las lecturas inicialmente tienen la misma probabilidad de surgir al desconocer la forma del mundo, de esta manera necesitamos identificar únicamente el término $P(O|I)$.

No obstante este término no es tan fácil de especificar como en el caso anterior. Por una parte, la simplificación de que dada una entrada siempre se



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

203

va a generar la misma salida no tiene porqué funcionar de manera correcta en este caso. Por otra, en la unidad anterior el tamaño de la distribución era tratable, con $3^6 = 729$ elementos. En el caso actual cada variable puede tener 18 grados de intensidad, con lo que nos encontramos con $18^5 = 1889568$ combinaciones a especificar, con lo cual la definición exhaustiva de la distribución se hace inviable.

Existen dos dificultades principales por las cuales es complicado para el programador determinar dada una entrada cuál es la salida deseada: por una parte no es apropiado en este caso suponer que una entrada genera siempre la misma salida, por otra ésta distribución depende sobremanera del entorno donde vaya trabajar el robot y por lo tanto no es útil el fijar el valor de esta distribución no dando opción al aprendizaje.

Por tanto puede ser apropiado definir cada entrada como una distribución gaussiana, donde se prefiere una opción de salida (pero no es única). De esta manera es necesario proporcionar un mecanismo de aprendizaje que permita obtener el valor de esta distribución a partir de un determinado entorno. Este mecanismo vendrá dado tras la aplicación del operador de fusión competitiva.

6.1.4. Fusionando unidades Bayesianas

Como se ha comentado anteriormente en el capítulo 4, la conexión de unidades de procesamiento suele requerir un flujo de información continuo que ha de ser proporcionado por preguntas a otras unidades. Es posible que en determinadas situaciones o que para algunos tipos de información sea conveniente combinar dos o más unidades de procesamiento para formar una única unidad. La combinación de unidades dependerá del tipo de información y del método de fusión que se desee utilizar. En nuestro caso nos interesa mostrar de manera sencilla las posibilidades que aporta el operador de fusión competitivo. Para ello nos centraremos en un método de fusión competitiva reactiva.

Se define una configuración de unidades de proceso como competitiva o redundante si cada unidad obtiene medidas independientes de una misma propiedad. Se distinguen dos configuraciones competitivas [VG99]: la fusión de datos provenientes de diferentes sensores o las medidas obtenidas por un único sensor en diferentes instantes de tiempo. Las configuraciones competitivas, entre otras ventajas, proporcionan robustez al sistema.

Si disponemos de dos unidades u^1 y u^2 , capaces de responder pertinentemente a sus preguntas asociadas se desea obtener una nueva unidad u que



aglutine la información de ambas unidades. En nuestro caso, como deseamos obtener unidades reactivas, prescindiremos del estado en ambas unidades, simplificando los resultados del proceso de fusión. Así pues se definen las variables de la nueva unidad u como:

$$\begin{aligned} I &= I_1 \wedge I_2, |I| = |I_1| + |I_2| \\ O &= O_1 \cup O_2, |O| = |O_1| + |O_2| - |O_1 \cap O_2| \\ S &= \emptyset, |S| = 0 \end{aligned}$$

Proponiendo la siguiente descomposición:

$$\begin{aligned} P(I \otimes O) &= P(I_1 \otimes I_2 \otimes O) = \\ &= P(O) \times P(I_1|O) \times P(I_2|O \otimes I_1) = \\ &= P(O) \times P(I_1|O) \times P(I_2|O) \end{aligned}$$

Donde la segunda igualdad se deduce de la independencia condicional entre I_1 y I_2 . Esta independencia será revisada en la siguiente sección.

6.1.5. Obtención de la unidad u_2

En el operador de fusión competitiva genérico se estableció la siguiente igualdad $P(I_2|O \otimes I_1) = P(I_2|O)$ derivada del hecho de considerar la independencia condicional de la información de la primera unidad con respecto a la segunda. No obstante esta independencia puede no ser válida en el caso actual. Es poco probable que para un robot que utilice dos tipos de sensores que observan una misma propiedad sus medidas no estén en cierta manera relacionadas. Obviamente no sabemos a priori que distribución puede representar dicha relación.

No obstante podemos intentar obtener la distribución $P(I_2|O \otimes I_1)$ de manera experimental. Anteriormente hemos definido una unidad de rango u_1 para la evitación de obstáculos. Esta unidad es totalmente funcional y por tanto la podemos utilizar de manera independiente. De esta manera para obtener la distribución $P(I_2|O \otimes I_1)$ podemos recorrer el entorno utilizando u_1 a la vez que se obtiene la entrada de la segunda unidad I_2 (en nuestro caso obtendríamos medidas en paralelo de los sónares de ultrasonidos y de la cámara de video). Dada una salida O y una entrada I_1 existirá una entrada I_2 que representará la distribución pero esta no será la única, asumiremos que $P(I_2|O \otimes I_1)$ estará formada por un conjunto de gaussianas discretas $P(I_2|O \otimes I_1) = G(O, I_1)$ que primarán a un conjunto de salidas cercanas entre sí.



6.1.6. Experimentación

Pruebas con el simulador

Como se ha comentado anteriormente la primera fase del proceso de experimentación trata de obtener los valores de las entradas de las dos subunidades, I_1, I_2 utilizando únicamente la primera unidad u_1 . A continuación, en la figura 6.6, se muestra un recorrido de 4 minutos utilizando la unidad u_1 . Además se resaltan cuatro posiciones (dos detalladas en las figuras 6.4 y 6.5) con sus correspondientes medidas de ultrasonidos (I_1) y visuales (I_2).

Obviamente aparte de la discretización en zonas debemos discretizar el espacio RGB. En este caso se ha dividido uniformemente el cubo RGB en 18 zonas. En nuestro mundo virtual este número de zonas ha sido suficiente para diferenciar el entorno de manera correcta.

La segunda parte del proceso de experimentación consiste en desconectar la unidad de ultrasonidos y dejar en funcionamiento al robot únicamente con el sistema visual. Si se ha producido un aprendizaje correcto el robot debe desarrollar un comportamiento en la misma línea que la unidad u_1 . A continuación se muestran tres recorridos realizados por el robot utilizando únicamente la unidad visual.

En la figura 6.7 se observan estas rutas desarrolladas por el robot utilizando únicamente la cámara de video. Como se puede observar el comportamiento del robot es el esperado. En la ruta azul observamos como el robot es capaz de seguir aproximadamente la misma trayectoria que en la fase de entrenamiento. Además con el sistema visual detecta antes los obstáculos con lo que no ajusta tanto sus movimientos como lo haría con el sónar de ultrasonidos. En la ruta verde se dispone el robot en una posición que no se alcanzó nunca durante la fase de entrenamiento (ver figura 6.6). Se puede observar como el robot es capaz de desenvolverse de manera correcta incluso en esta circunstancia. Por último la ruta roja refleja una situación no aprendida en la fase de entrenamiento, encontrarse de frente con una columna y por tanto saturando las 5 variables de entrada a naranja. En esta situación el robot no tiene información de como actuar por lo que no realiza ningún movimiento para evitar la columna.

De esta manera podemos observar como la unidad u_2 funciona correctamente incluso para situaciones no aprendidas previamente. No obstante es conveniente que la fase de entrenamiento sea lo más completa posible para evitar problemas como el de la ruta roja de la figura 6.7.

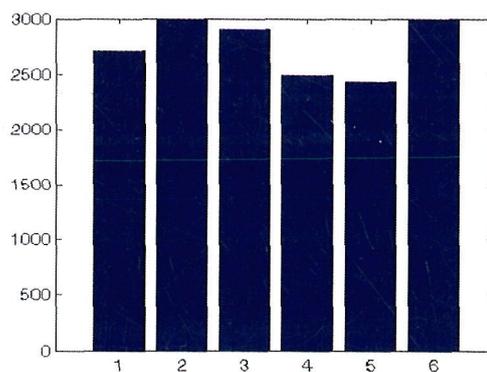
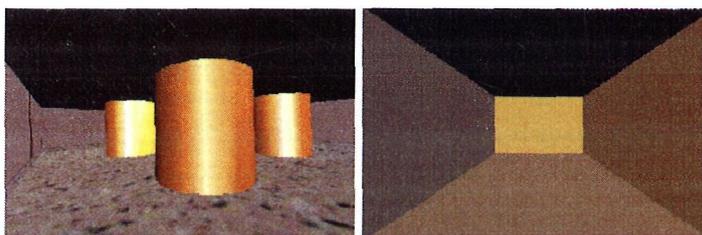


Figura 6.4: Imagen virtual detectada por el robot en la posición 2 del mapa de la figura 6.6 junto con su discretización espacial (para mayor claridad no se muestra la discretización de intensidad). Además se adjunta la lectura del anillo completo de sónares de ultrasonidos (para la unidad solo se utilizarán del 2 al 6).



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

207

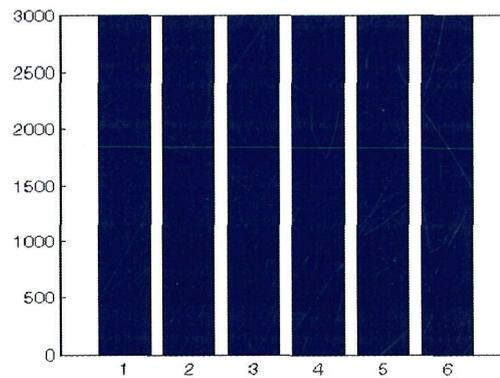
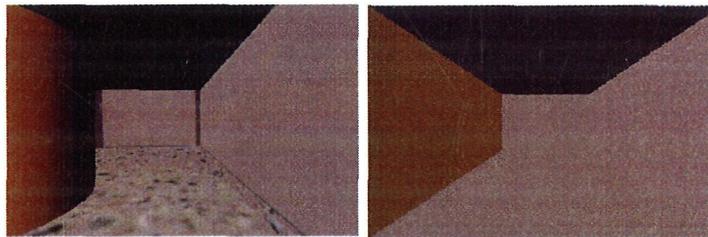


Figura 6.5: Imagen virtual detectada por el robot en la posición 3 del mapa de la figura 6.6 junto con su discretización espacial (para mayor claridad no se muestra la discretización de intensidad). Además se adjunta la lectura del anillo completo de sónares de ultrasonidos (para la unidad solo se utilizarán del 2 al 6).

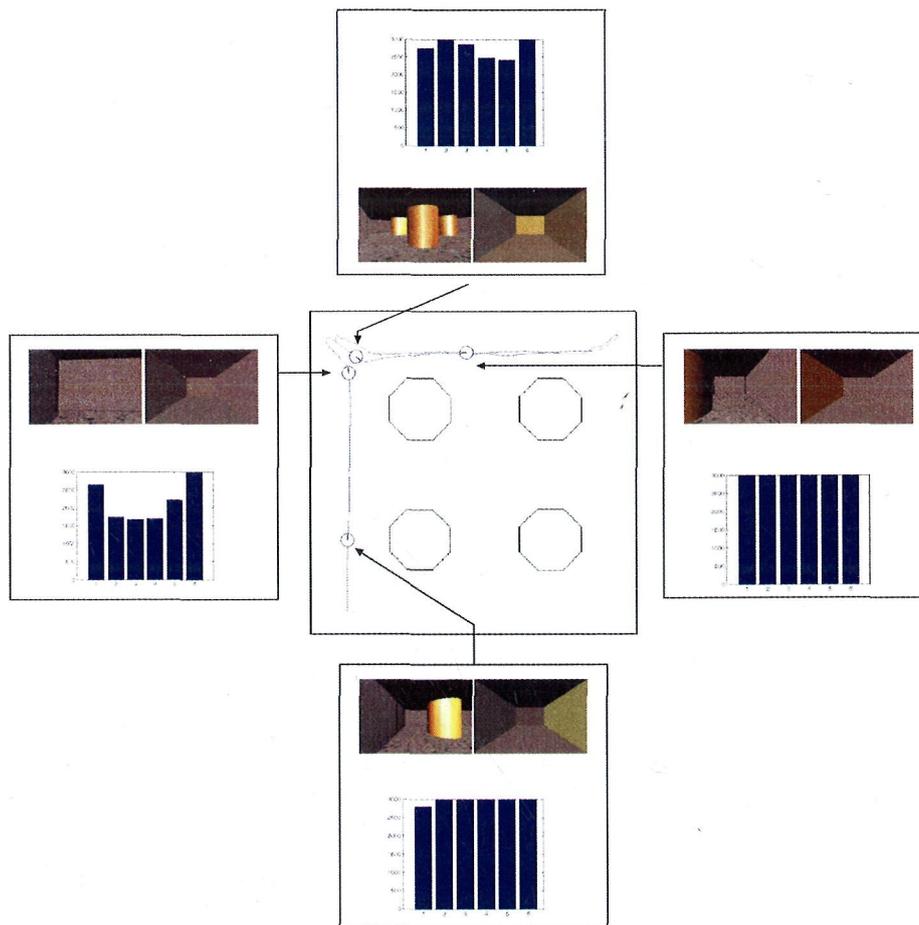


Figura 6.6: Ejemplo de ruta recorrida por el robot utilizando la unidad de evitación de obstáculos por ultrasonidos. Además se muestran tres hitos donde para cada uno de ellos se especifica la lectura de la cámara, su discretización y las lecturas de los sones de ultrasonidos.

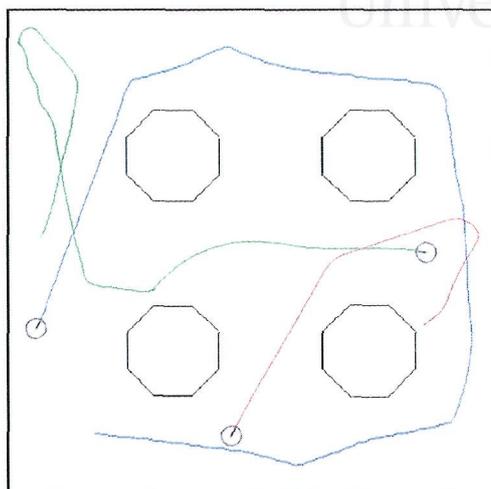


Figura 6.7: Rutas desarrolladas utilizando únicamente la unidad de evitación de obstáculos mediante cámara.

Pruebas con el robot Pioneer 3DX

Al igual que en el caso anterior se han desarrollado pruebas de funcionamiento de los sistemas de fusión presentados pero esta vez utilizando el robot Pioneer 3. Las tareas de navegación se han realizado en el pasillo de la figura 6.8. El recorrido de entrenamiento del robot se muestra con 8 hitos para los cuales se observan las imágenes obtenidas en el entorno real.

En la figura 6.9 se observa, para los hitos de la figura anterior, las medidas de los sones de ultrasonidos y la discretización en zonas para cada imagen (al igual que en el caso anterior no se muestra la discretización de niveles de intensidad).

Una vez obtenida la distribución $P(I_2|O \otimes I_1)$ podemos desconectar los sones de ultrasonidos y evitar los obstáculos utilizando únicamente la cámara de video. La figura 6.10 muestra un ejemplo de trayectoria utilizando la distribución aprendida en la fase anterior.

Es interesante comentar cómo en el hito 4 el sistema visual avanza sus movimientos, comparándolo con los sones de ultrasonidos. Esto se debe a la existencia de zonas donde el *perceptual aliasing*¹ hace que sean idénticas

¹Cuando dos o más objetos generan el mismo patrón sensorial, pero requieren de diferentes respuestas motrices

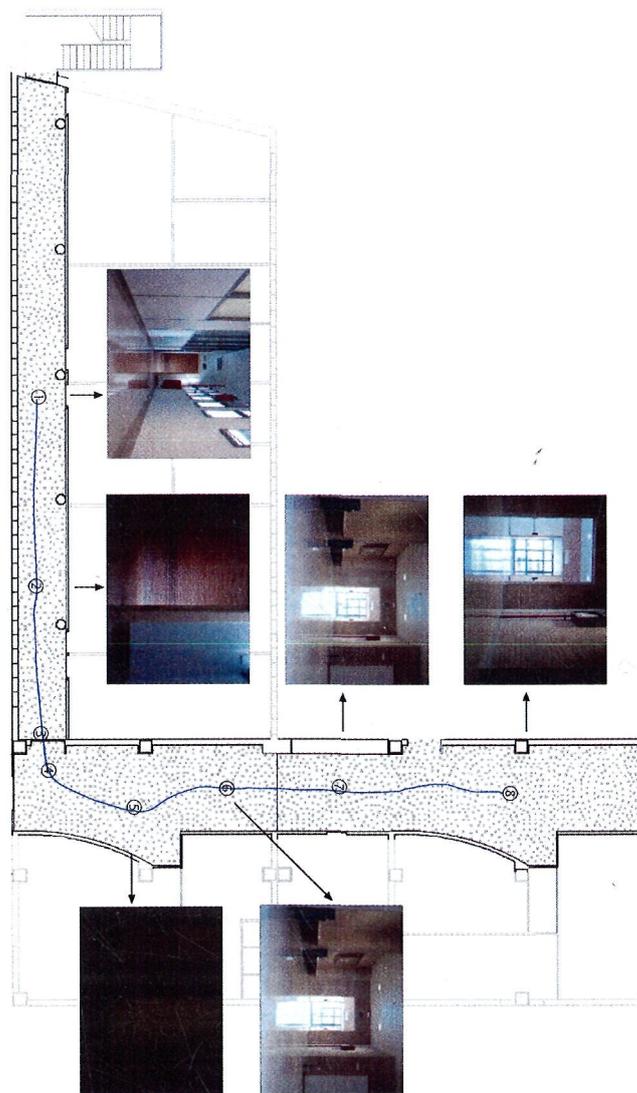


Figura 6.8: Recorrido de entrenamiento del robot Pioneer 3DX en la fase de entrenamiento de la unidad u_2 . Se muestran imágenes reales del entorno obtenidas en 8 hitos distintos.



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

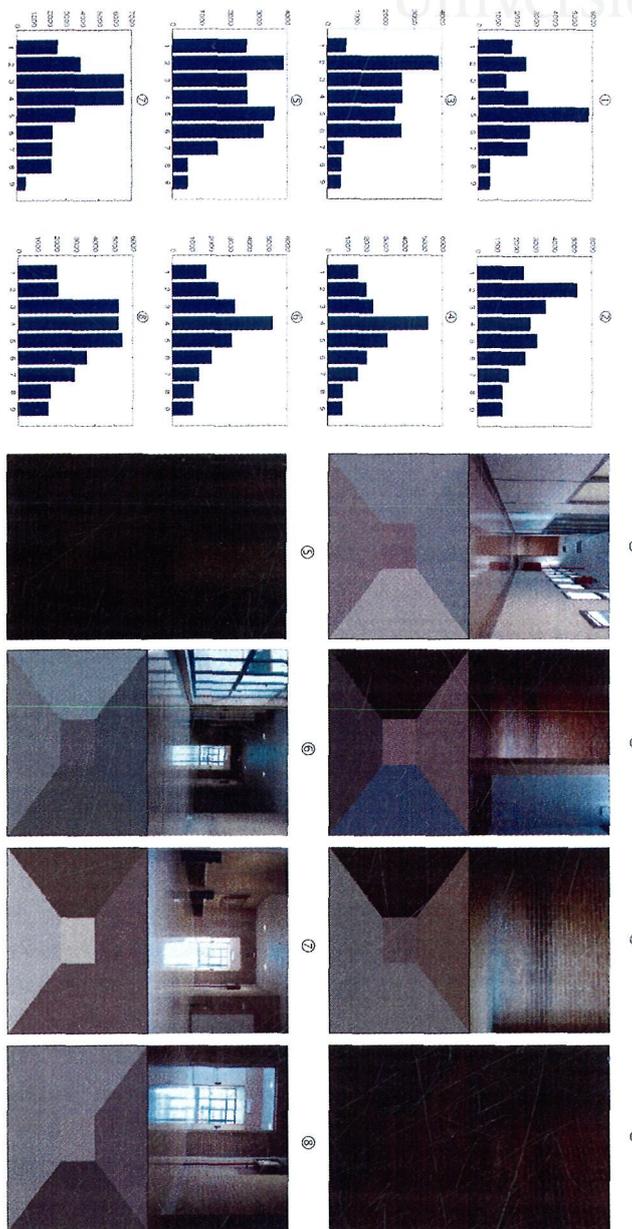


Figura 6.9: Lecturas de los sensores de ultrasonidos y discretización de las imágenes obtenidas para cada hito de la figura 6.8

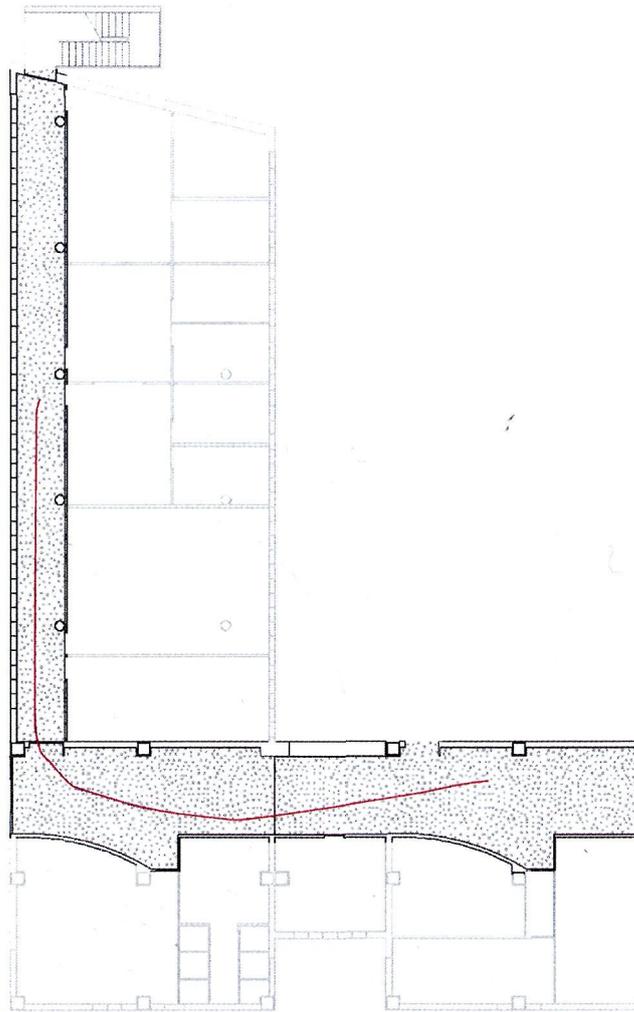


Figura 6.10: Trayectoria realizada por el robot utilizando únicamente la unidad visual u_2 .



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

213

para el número de variables utilizado. Aunque este efecto nos es beneficioso en este caso, puede desembocar en un funcionamiento incorrecto del sistema. Por ello es importante determinar el número de zonas y la discretización de los niveles de intensidad a utilizar para el problema a desarrollar.

6.1.7. Discusión

En esta sección se ha proporcionado una aplicación del operador de fusión competitiva que permite incluir un sensor, del cual no se tenía información relativa a su uso junto con el robot, a partir de otro sensor de distinto tipo. Revisando la independencia condicional del operador de fusión competitiva se ha proporcionado un mecanismo que permite aprender el uso de un nuevo sensor. Este método se puede utilizar de dos maneras diferenciadas: para aprender cómo utilizar un nuevo sensor que sustituirá al anterior (como en el caso mostrado anteriormente) o para aprender un nuevo sensor que complementará las observaciones del primero.

La especificación de la tarea a desarrollar es necesaria como fase previa al aprendizaje. Es importante esta discretización ya que una discretización incorrecta puede desembocar en un sistema incapaz de desarrollar la tarea requerida. Una de las líneas de investigaciones futuras versa en determinar de manera óptima el número de variables a utilizar para desarrollar una conducta determinada.

Ya se ha comentado que es necesario un proceso de aprendizaje para determinar el funcionamiento del nuevo sensor. Cabe destacar que este aprendizaje ha de ser lo más exhaustivo posible. Como se ha visto anteriormente, cualquier situación no contemplada puede desembocar en un funcionamiento incorrecto del sistema. Una solución para relajar este problema es sustituir progresivamente el uso del primer sensor por el segundo, de manera que se garantice la observación de un número suficiente de situaciones. Queda como línea futura de investigación la utilización de unidades Bayesianas con uno o más estados. Estas variables de estado pueden ser también de utilidad a la hora de implementar conductas complejas que sufran, por ejemplo, el problema de *perceptual aliasing*.

Es importante destacar las ventajas que conlleva el uso del operador de fusión competitiva para aprender un nuevo sensor que complementará las acciones del otro para desarrollar una misma tarea. Por una parte se mejora la robustez del sistema dada la redundancia intrínseca que capacita al sistema para proporcionar información incluso en el caso de fallos. También es impor-



GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA 214 DE LÁSER Y VISIÓN OMNIDIRECCIONAL

tante remarcar que al aumentar la dimensionalidad del espacio de medidas conseguimos un sistema más inmune al ruido. Al incluir sensores de distinto tipo el ruido que afecta a uno no tiene porqué afectar al otro disminuyendo de esta manera la sensibilidad del sistema a las interferencias. Resumiendo, obtenemos las ventajas propias de fusionar información sensorial (más robustez y confiabilidad, extensión de la cobertura espacial y temporal, incremento de la confianza, reducción de la ambigüedad y la incertidumbre, robustez contra interferencias, se mejora la resolución).

6.2. Generación de Mapas 3D Mediante Fusión Bayesiana de Láser y Visión Omnidireccional

La construcción de mapas y la localización de un agente robótico tienen gran importancia en la robótica y la inteligencia artificial, siendo dos campos de investigación ampliamente estudiados durante décadas [TBF00]. La localización robusta de un robot que trabaja en un determinado entorno depende de diversos factores, considerándose uno de los más importantes el mapa del entorno del que dispone el robot [PCM⁺99]. Por esta razón la construcción de mapas ha sido considerada como uno de los problemas más importantes para el diseño de robots autónomos.

No obstante, como se ha comentado en el capítulo 3, al utilizar un modelo del entorno, se debe considerar que este puede ser incompleto debido a la existencia de variables ocultas, que no se tienen en cuenta, pero que influyen sobre el modelo. El aprendizaje y la inferencia probabilística tratan de resolver este problema mediante una base formal. La programación Bayesiana es un formalismo propuesto como una solución para tratar problemas que conlleven incertidumbre y no completitud.

Por otra parte, en el capítulo 2 vimos que cuando se dispone de información procedente de distintos sensores se puede fusionar aumentando entre otras características, la robustez y seguridad del sistema [Elm02]. De la misma manera se consigue mejorar la consistencia del mapeado y la localización del robot [AZ03]. Como se comentó anteriormente las unidades de proceso Bayesianas son un modelo de fusión, basado en la programación Bayesiana, orientado al área de la robótica móvil.

En esta sección se aporta un sistema de reconstrucción de mapas en 3D a

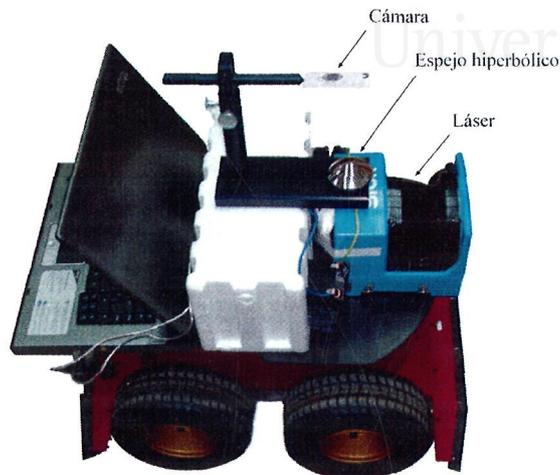


Figura 6.11: Robot Pioneer P3-AT con los sistemas de láser y omnivisión.

partir de la información obtenida por un láser y un sistema de omnivisión. La construcción del mapa se lleva a cabo en diferentes fases. En una primera fase se utiliza un láser Sick LMS 220² situado verticalmente en la parte superior del robot (ver figura 6.11). Utilizando esta disposición se obtiene con cada barrido del láser una sección del entorno. Uniendo todas estas secciones se registra una nube de puntos que forma el mapa tridimensional del entorno. Por otro lado, utilizando un cámara de vídeo de baja resolución (640x480) montada sobre un espejo hiperbólico, proporcionado por Neovision³, se obtiene una imagen omnidireccional cada vez que se registra un barrido del láser.

Para realizar el proceso de fusión básico de las imágenes omnidireccionales y de las medidas del láser se han definido dos unidades Bayesianas. Una primera unidad será la encargada de obtener la información espacial a partir de las medidas del láser. Utilizando esta información una segunda unidad la combinará con la imagen de omnivisión para añadir información de color al mapa tridimensional generado. Para finalizar el proceso de fusión se combinarán varias de las unidades encargadas de obtener el color de un punto, para conseguir mayor resolución y robustez en el proceso de mapeado.

²<http://www.sick.es>

³<http://www.neovision.cz/>



6.2.1. Operador de fusión competitivo

Aunque se pueden definir varios operadores de fusión que permiten agrupar unidades Bayesianas, en esta sección únicamente se va a utilizar un operador de fusión competitivo para unidades reactivas ya que se quiere primar sobre todo la velocidad de generación del mapa. Como se comentó en el capítulo 4 se define una configuración de unidades de proceso como competitiva o redundante si cada unidad obtiene medidas independientes de una misma propiedad. Disponemos de dos unidades u_1 y u_2 , capaces de responder pertinentemente a sus preguntas asociadas. Se desea obtener una nueva unidad u que aglutine la información de ambas unidades. Aunque el número de estados influirá en cómo se fusionan dos unidades, en este caso sólo mostraremos el operador capaz de fusionar unidades sin ningún estado. Además se supondrá que desconocemos cómo combinar a priori las salidas de u_1 y u_2 .

Fusión competitiva, unidades reactivas, $|S| = 0$.

En el caso de la fusión de unidades reactivas, aquellas unidades que no contienen ningún estado, se definen las variables de la nueva unidad u como:

$$\begin{aligned} I &= I_1 \wedge I_2, |I| = |I_1| + |I_2| \\ O &= O_1 \cup O_2, |O| = |O_1| + |O_2| - |O_1 \cap O_2| \\ S &= \emptyset, |S| = 0 \end{aligned} \quad (6.1)$$

De esta manera, la variable de entrada I estará formada por las entradas de las subunidades. Como no sabemos cómo combinar las salidas de las subunidades al fusionarlas, la salida de la unidad global estará formada por todas las posibles salidas de éstas. Basándose en la distribución de probabilidades anterior se propone la siguiente descomposición:

$$\begin{aligned} P(I \otimes O) &= P(I_1 \otimes I_2 \otimes O) = \\ &= P(O) \times P(I_1|O) \times P(I_2|O \otimes I_1) = \\ &= P(O) \times P(I_1|O) \times P(I_2|O) \end{aligned} \quad (6.2)$$

Donde la segunda igualdad se deduce de la independencia condicional entre I_1 y I_2 . La probabilidad de una determinada salida $P(O)$ se considera a priori uniforme. Por otra parte ambas distribuciones $P(I_1|O)$ y $P(I_2|O)$ han de ser especificadas por el programador, de manera que dada una salida se determine qué entradas pudieron generarlas.



6.2.2. Unidades Bayesianas propuestas

Obtención de los puntos en el espacio 3D

Una primera fase para el mapeado es la obtención de la nube de puntos que representa el entorno sobre el cual se desplazará el robot. Como se ha comentado anteriormente disponemos de un láser en posición vertical capaz de obtener una sección de 180° en una posición determinada. Desplazando el robot a través del entorno obtendremos un conjunto de secciones que unidas entre sí nos proporcionan la descripción del entorno deseada. Así pues, la unidad Bayesiana encargada del proceso de obtención de los puntos 3D del entorno dispondrá de las siguientes variables:

$$(I; S; O) : (A_L, D_L, P_R; \emptyset; P_W)$$

De esta manera las variables de entrada están formadas por: los ángulos del barrido del láser A_L que son conocidos y van desde 0° hasta 180° en intervalos de 0.5° , la distancia que devuelve el láser D_L para un ángulo dado (de 0 hasta 10m con una exactitud de 10mm) y la posición actual del robot P_R . El estado en esta unidad no es necesario al tratarse de un proceso reactivo donde dada una entrada se proporcionará directamente una salida. P_W es la nube de puntos que se desea obtener a partir de los datos de entrada.

Utilizando las variables anteriores construimos su distribución conjunta proponiendo una descomposición de la misma:

$$P(A_L \otimes D_L \otimes P_R \otimes P_W) = P(P_R) \times P(A_L) \times P(D_L | A_L \otimes P_R) \times \\ \times P(P_W | A_L \otimes D_L \otimes P_R)$$

Podemos suponer que $P(P_R)$ es uniforme si no necesitamos que el robot empiece el proceso de mapeado en una localización determinada, como es nuestro caso, y por tanto todas las posiciones del entorno tienen la misma probabilidad de ser visitadas. De la misma manera, $P(A_L)$ se considera uniforme ya que siempre se utilizarán para cada barrido todos los ángulos del láser A_L independientemente de la posición del robot. Por otra parte, $P(D_L | A_L \otimes P_R)$ viene determinado por las medidas del láser en una posición y ángulo dados y por las especificaciones del mismo. En nuestro caso el láser LMS 200 tiene un error normal de $err_L = \pm 15mm$ al realizar sus medidas. $P(P_W | A_L \otimes D_L \otimes P_R)$ corresponde exactamente a la pregunta que deseamos obtener sobre la distribución conjunta: dado el ángulo y la distancia



GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA DE LÁSER Y VISIÓN OMNIDIRECCIONAL

218

retornada por el láser en una posición dada, ¿a qué punto en el espacio 3D corresponde? Como se ve en la ecuación 6.3 (donde z es la dirección de avance, y es la vertical y x la horizontal) la obtención de un punto es sencilla de calcular aplicando las transformaciones geométricas oportunas. No obstante recordamos que una distancia D_L viene representada por una distribución normal. De esta manera para un ángulo del láser junto con la distancia obtenida en una posición dada no se obtiene únicamente un punto sino una gaussiana para cada coordenada (x, y, z) que indicará el conjunto de puntos candidatos a representar las medidas del láser junto a su probabilidad.

$$P_W = (P_{W_x}, P_{W_y}, P_{W_z}) = \begin{pmatrix} G(\mu(D_L \cdot \cos(A_L) + P_{R_{\mu x}}), \sigma(err_L + P_{R_{\sigma x}})), \\ G(\mu(D_L \cdot \sin(A_L)), \sigma(err_L)), \\ G(\mu(P_{R_{\mu z}}), \sigma(P_{R_{\sigma z}})), \end{pmatrix} \quad (6.3)$$

Se supone que el suelo es un único plano sin desnivel y por tanto las coordenadas del robot P_{R_y} son 0. Por otra parte se supone conocido el error de posicionamiento del robot representado por $P_{R_{\mu}}$ y $P_{R_{\sigma}}$.

Debemos señalar que se ha supuesto que la posición del robot P_R es conocida. Se podría utilizar otra unidad de proceso encargada de relocalizar el robot utilizando las medidas obtenidas previamente. Existen numerosos métodos en la literatura capaces de realizar dicha relocalización tanto en entornos de interior como en entornos no estructurados utilizando láser. Es posible el uso de alguna aproximación a la localización probabilística, como puede ser un filtro de Kalman para estimar la posición del robot. Un filtro de Kalman se puede definir utilizando el mismo paradigma sobre el que se fundamentan las unidades Bayesianas, la programación Bayesiana, como puede verse en [BG03].

Determinación del color de un punto

Cuando obtenemos una única fotografía de un entorno tridimensional es imposible llevar a cabo su reconstrucción ya que hay una pérdida de información espacial. Al combinar un sensor activo con la obtención de imágenes proporcionamos un mecanismo sencillo para obtener la información de color de un punto dado. En nuestro caso, cada vez que se obtiene un barrido del láser se guarda una imagen de omnivisión que almacena el color del entorno. Al usar una imagen panorámica obtenemos información visual sobre 180° del entorno y por tanto sobre todos los puntos devueltos por el láser.

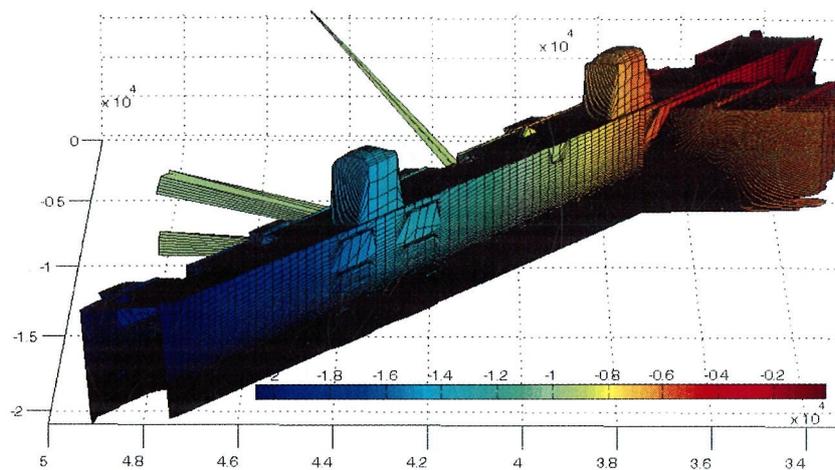


Figura 6.12: Mapa de puntos obtenido mediante la primera unidad Bayesiana, representando únicamente $\mu(P_W)$. La barra indica la profundidad del plano. Se observa un pasillo del departamento DCCIA. Partiendo del origen y moviéndose en el sentido del pasillo se puede ver una sala de 2 hasta 6 metros y dos puertas de despachos con su correspondiente ventanal superior a partir de los 14m. Se observan también dos bóvedas que proporcionan luz cenital al pasillo.



**GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA
DE LÁSER Y VISIÓN OMNIDIRECCIONAL**

220

Universidad de Alicante

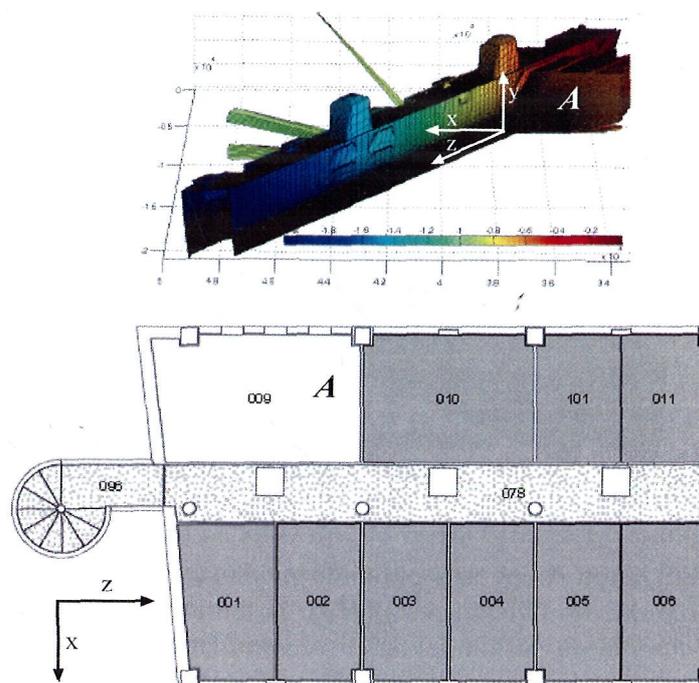


Figura 6.13: Plano del pasillo escogido para la realización de las pruebas. Se muestra también el espacio utilizado. La zona A de la figura 6.12 se sitúa en la A enmarcada en plano real del pasillo.



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

221

Una unidad de proceso Bayesiana, capaz de obtener para cada punto del entorno su color debe tener las siguientes variables de entrada:

$$(I; O) = (P_W, Omni; P_C)$$

Donde P_W representa los puntos tridimensionales obtenidos por la unidad anterior, $Omni$ es la imagen omnidireccional y P_C es la tripleta de color que corresponde a cada valor P_W . De esta manera podemos descomponer la unión de variables como:

$$P(P_W \otimes Omni \otimes P_C) = P(P_W) \times P(Omni|P_W) \times P(P_C|P_W \otimes Omni)$$

Donde la probabilidad de que se dé un punto en el espacio $P(P_W)$ es uniforme ya que a priori suponemos que no se tiene información sobre la forma del entorno. $P(Omni|P_W)$ es uniforme porque un punto en el espacio puede tener a priori cualquier color. Por otra parte la distribución $P(P_C|P_W \otimes Omni)$ está íntimamente relacionada con la proyección de un punto P_W en el espejo hiperbólico utilizado para la generación de la imagen $Omni$.

$$R : \begin{cases} x = p_x + t(q_x - p_x) \\ y = p_y + t(q_y - p_y) \\ z = p_z + t(q_z - p_z) \end{cases} \quad H : \begin{cases} \frac{z^2}{c^2} - \frac{x^2+y^2}{a^2} = 1 \\ a^2 = b^2 = 548,1440; c^2 = 789,3274 \\ q = \sqrt{a^2 + b^2} \end{cases}$$

Siendo R la recta que pasa por un punto p hasta el foco de la hipérbola q , la proyección de p sobre la hipérbola se define como la intersección de la hipérbola H y la recta R , de esta manera:

$$\text{Proj}_H(p) : \begin{cases} (a^2(q_z - p_z)^2 - c^2(q_x - p_x)^2 - c^2(q_y - p_y)^2) t^2 + \\ (a^2 2p_z(q_z - p_z) - c^2 2p_x(q_x - p_x) - c^2 2p_y(q_y - p_y)) t + \\ a^2 p_z^2 - c^2 p_x^2 - c^2 p_y^2 - a^2 c^2 = 0 \end{cases}$$

Donde t es la incógnita a despejar obteniendo como resultado los dos puntos de intersección de la recta sobre la hipérbola.

Tal y como se comentó anteriormente P_W no contiene puntos sino distribuciones gaussianas. Aproximando la proyección de la distribución gaussiana a un plano tangente a la hipérbola (ver figura 6.15) definimos la distribución $P(P_C|P_W \otimes Omni)$ como una gaussiana bidimensional relativa a los píxeles de la imagen $Omni$:



GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA
222 DE LÁSER Y VISIÓN OMNIDIRECCIONAL

$$\frac{z^2}{789,3274} - \frac{x^2 + y^2}{548,1440} = 1$$

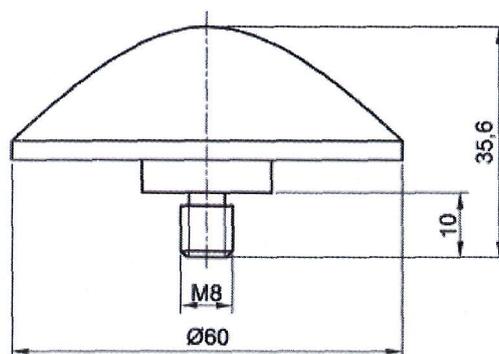


Figura 6.14: Forma, ecuación y medidas del espejo H3G de omnivisión proporcionado por Neovision.

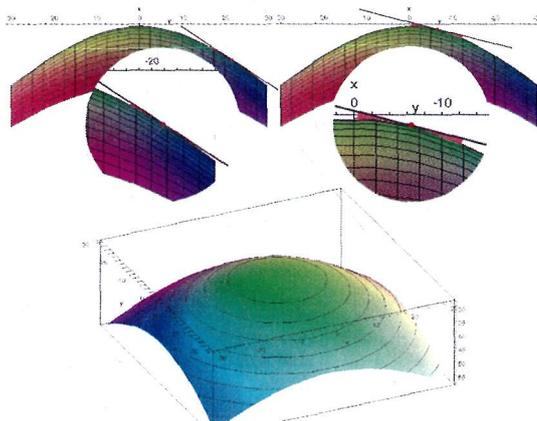


Figura 6.15: Plano tangente a la hipérbola. Los puntos superiores del espejo son los que sufren más distorsión, no obstante son los menos importantes ya que a priori el techo no se utilizará para tareas de navegación. Los puntos cercanos a las paredes y el suelo presentan un error prácticamente inapreciable.



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

223

$$\begin{aligned}
 P(P_C | P_W \otimes Omni) &= G_{Omni_x, Omni_y}(\mu_1, \mu_2, \sigma_1, \sigma_2, [\rho = 0]) = \\
 &= G_{Omni_x, Omni_y}(\text{Proj}_H(P_W)_x, \text{Proj}_H(P_W)_y, \text{Proj}_H(\sigma_x), \text{Proj}_H(\sigma_y)) = \\
 &= \frac{e^{-\left(\frac{(x - \text{Proj}_H(P_W)_x)^2}{2\text{Proj}_H(\sigma_x)^2} - \frac{(y - \text{Proj}_H(P_W)_y)^2}{2\text{Proj}_H(\sigma_y)^2}\right)}}{2\pi \text{Proj}_H(\sigma_x) \cdot \text{Proj}_H(\sigma_y)}
 \end{aligned}$$

6.2.3. Fusión de unidades de proceso del color

Es fácil darse cuenta de que la unidad Bayesiana que determina el color de un punto del espacio descarta gran cantidad de información, que puede ser fusionada. Por una parte sólo obtiene el color relativo a los puntos retornados por el plano actual del láser (descartando las lecturas cercanas que también aportan información relevante). Por otra parte para cada plano obtenido por el láser en una posición z sólo se utiliza la imagen de $Omni_z$ correspondiente.

Sería por tanto conveniente para mejorar la calidad del proceso de mapeado introducir más información sobre el color de un punto en el espacio fusionando dos unidades de proceso de color. De esta manera, aplicando las ecuaciones 6.1 y 6.2 obtenemos:

$$\begin{aligned}
 U : (I_1; I_2; O) &= (P_{W_1}, Omni_1; P_{W_2}, Omni_2; P_{C_1} \cup P_{C_2}) \\
 P(I_1 \otimes I_2 \otimes O) &= P(O) \times P(P_{W_1} \otimes Omni_1 | O) \times P(P_{W_2} \otimes Omni_2 | O)
 \end{aligned}$$

Siendo $P(O)$ uniforme ya que a priori suponemos que todos los colores son equiprobables. Tanto $P(P_{W_1} \otimes Omni_1 | O)$ como $P(P_{W_2} \otimes Omni_2 | O)$ determinan la probabilidad de que un determinado color represente al punto del espacio definido por P_W y $Omni$. No obstante resultaría más cómodo trabajar con la distribución $P(O | P_W \otimes Omni)$ al estar definida previamente. Es fácil comprobar que:

$$\begin{aligned}
 P(O | P_{W_1} \otimes Omni_1) &= \frac{1}{\Sigma} \times \sum_{P_{W_2}, Omni_2} \left(\frac{P(O) \times P(P_{W_1} \otimes Omni_1 | O) \times}{P(P_{W_2} \otimes Omni_2 | O)} \right) = \\
 &= \frac{1}{\Sigma} \times P(O) \times P(P_{W_1} \otimes Omni_1 | O) \times \sum_{P_{W_2}, Omni_2} P(P_{W_2} \otimes Omni_2 | O) = \\
 &= \frac{1}{\Sigma} \times P(P_{W_1} \otimes Omni_1 | O)
 \end{aligned}$$

Siendo $\frac{1}{\Sigma}$ una constante de normalización. Así podemos reescribir la fusión de las dos unidades como:

$$P(I_1 \otimes I_2 \otimes O) = P(O) \times P(O | P_{W_1} \otimes Omni_1) \times P(O | P_{W_2} \otimes Omni_2)$$



**GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA
DE LÁSER Y VISIÓN OMNIDIRECCIONAL**

224

Universidad de Alicante

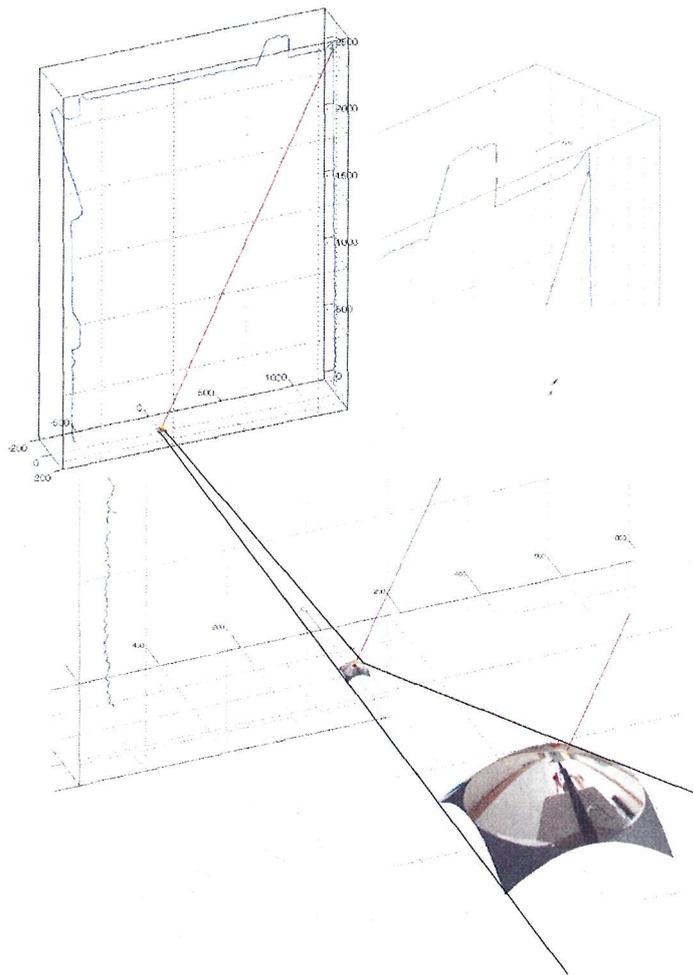


Figura 6.16: Ejemplo de una sección del entorno tomada por el láser junto con el espejo de omnivisión y el píxel que le corresponde (en amarillo).



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

225

Donde tanto $P(O|P_{W_1} \otimes Omni_1)$ como $P(O|P_{W_2} \otimes Omni_2)$ se han definido previamente en las subunidades fusionadas cuando $[O \subseteq P_{W_1}]$ y $[O \subseteq P_{W_2}]$ y se consideran uniformes en otro caso. De esta manera la combinación de unidades resulta en un producto de gaussianas bidimensionales.

Así pues se pueden realizar dos tipos de fusión utilizando la combinación de unidades. Por una parte dos subunidades que se van a fusionar pueden compartir la misma imagen *Omni* pero distintos puntos. Por otra parte se pueden mantener fijos los puntos y variar las imágenes de omnivisión para las subunidades.

6.2.4. Resultados y experimentación

El proceso de reconstrucción del mapa se divide en tres fases. En una primera fase se obtiene la nube de puntos, que representa el entorno, utilizando la información procedente del láser. Como se ha comentado anteriormente, se debe tener en cuenta que la información que proporciona el láser para un ángulo dado no es únicamente un punto sino una gaussiana que indica el conjunto de puntos candidatos a representar las medidas del láser junto a su probabilidad. En la figura 6.12 se observa la reconstrucción de la malla 3D para uno de los pasillos de nuestro departamento.

En una segunda fase se añade el color. Para cada barrido del láser se captura una imagen omnidireccional (ver fig. 6.22a). Para ello se obtiene para cada punto del láser el color que lo representa en la imagen panorámica (ver fig. 6.18).

En la reconstrucción anterior se descarta gran cantidad de información que puede ser fusionada para aumentar la robustez y precisión de la representación. La fusión puede seguir dos vertientes. Por una parte se pueden obtener planos interpolados linealmente entre los datos de un plano pl_i y su plano siguiente pl_{i+1} y calcular los colores de cada punto utilizando una unidad Bayesiana de color. En nuestro caso se han utilizado 10 subplanos para la imagen obtenida en la figura 6.19. Por otra parte para cada interpolación entre pl_i y pl_{i+1} se pueden utilizar dos unidades, una para la imagen $Omni_i$ y otra para la imagen $Omni_{i+1}$ de manera que al fusionarlas se consiga una representación más uniforme.

Sin embargo, se podría plantear un tercer método de fusión: utilizar varias unidades (cada una con una imagen *Omni* distinta) para cada plano y combinarlas. Este método puede introducir mucho ruido en el sistema y por tanto no aumentar la calidad de las imágenes generadas. Hay que tener en cuenta



GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA
 DE LÁSER Y VISIÓN OMNIDIRECCIONAL

226

Alacant
 Universidad de Alicante

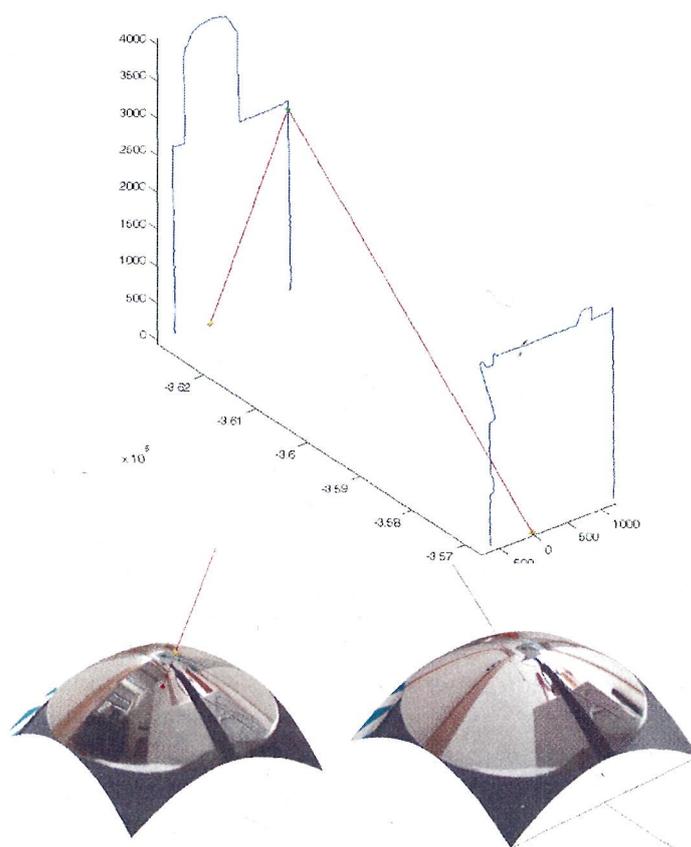


Figura 6.17: Dado un determinado punto (x,y,z) es posible obtener más de una proyección para determinar su color. En esta figura se muestra un ejemplo de proyección utilizando dos imágenes obtenidas en dos posiciones distintas para un mismo punto.



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

227

Universitat d'Alacant
Universidad de Alicante

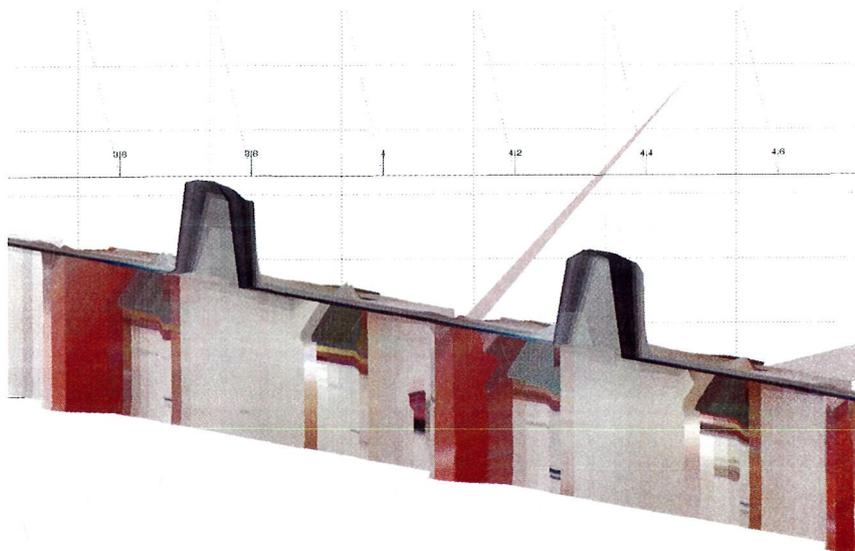


Figura 6.18: Mapa de puntos con la información de color P_C obtenida por una unidad Bayesiana de proceso del color para todos los planos del mapa. Se representa una sección del pasillo y únicamente el color más probable de la distribución.



GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA
DE LÁSER Y VISIÓN OMNIDIRECCIONAL

228

Universitat d'Alacant
Universidad de Alicante

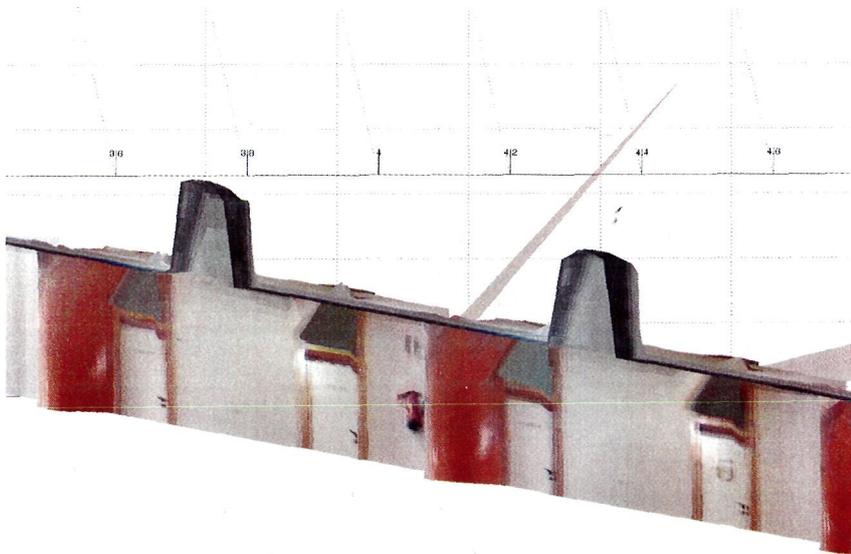


Figura 6.19: Reconstrucción de color P_C para cada plano obtenido fusionando la información de dos unidades. Se representa una sección del pasillo y únicamente el color más probable de la distribución.



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

229



Figura 6.20: Reconstrucción de color P_C para cada plano obtenido fusionando la información de dos unidades. Se representa la vista exterior del pasillo completo y únicamente el color más probable de la distribución.



GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA DE LÁSER Y VISIÓN OMNIDIRECCIONAL

230

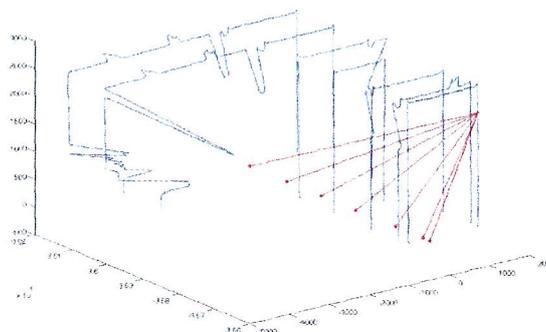


Figura 6.21: Proyección de un punto en distintas imágenes obtenidas por el espejo de omnivisión. Es necesario fusionar esta información para mejorar el mapa tridimensional obtenido.

que la unidad Bayesiana de color tiene una estrecha relación con la proyección de un punto (x, y, z) en la hipérbola H , de esta manera, cualquier error odométrico puede afectar sobremanera la proyección de planos lejanos en una imagen de omnivisión determinada proporcionando resultados incorrectos.

Por último cabe destacar la diferencia de nitidez entre una reconstrucción sin fusión de unidades Bayesianas de color (figura 6.22c) y una reconstrucción utilizando los mecanismos de fusión descritos previamente (figura 6.22d).

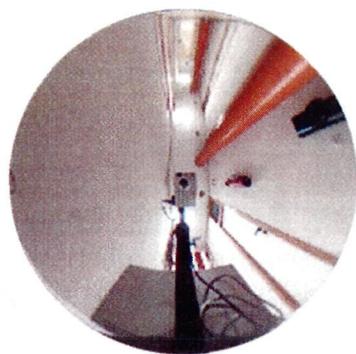
6.2.5. Comparación entre la reconstrucción con y sin fusión

Como se ha comentado anteriormente se observa claramente la diferencia de nitidez entre la reconstrucción con y sin fusión. No obstante nos interesa obtener una medida cuantitativa de dicha diferencia. La distancia entre imágenes continúa siendo un problema fundamental en el campo de la inteligencia artificial. Una de las primeras medidas utilizadas para comparar dos imágenes es la correlación bidimensional. Esta viene definida en la ecuación 6.4.



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

231



a)



c)



b)



d)

Figura 6.22: a) Imagen omnidireccional obtenida en un barrido del láser. b) Vista original del pasillo. c) Detalle de una reconstrucción utilizando únicamente una unidad. d) Reconstrucción conseguida fusionando dos subunidades. La flecha en la figura 6.19 indica el plano en el cual se han obtenido las tomas.



GENERACIÓN DE MAPAS 3D MEDIANTE FUSIÓN BAYESIANA DE LÁSER Y VISIÓN OMNIDIRECCIONAL

232

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A}) (B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right) \left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}} \quad (6.4)$$

Esta medida funciona correctamente si las dos imágenes a comparar están alineadas (como es nuestro caso). El coeficiente r se mueve en el rango de $[-1, 1]$, siendo 1 el valor de correlación mayor que se obtiene al comparar imágenes idénticas.

Dentro del campo de la visión artificial y la segmentación de imágenes también se han desarrollado métricas para la comparación cuantitativa de imágenes. En [BCS98] se comentan algunas de las técnicas de comparación de imágenes utilizadas en este campo. Nosotros proponemos una versión adaptada de la función de evaluación propuesta por Liu y Yang [LY94] en la ecuación 6.5, donde $m \times n$ especifica el tamaño de la imagen y e determina la distancia euclídea de las intensidades del píxel i . A menor valor de $F(I)$ más parecido entre imágenes.

$$F(I) = \frac{1}{1000(n \times m)} \sum_{i=1}^{n \times m} e_i^2 \quad (6.5)$$

Hemos seleccionado una de las imágenes con más información visual de la reconstrucción y de esta manera hemos comparado las dos reconstrucciones con una imagen real de la misma área del entorno (ver figura 6.23). Los resultados obtenidos se resumen en la siguiente tabla:

Distancia de la imagen real a:	Correlación	Correlación	Modificación	Modificación
	bidimensional en grises	bidimensional en color	Liu Yang en escala de grises	Liu Yang en color
Reconstrucción c (con fusión)	0,7908	0,7335	77,2597	78,0612
Reconstrucción b (sin fusión de unidades de color)	0,7229	0,6646	97,1092	96,0232



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

233

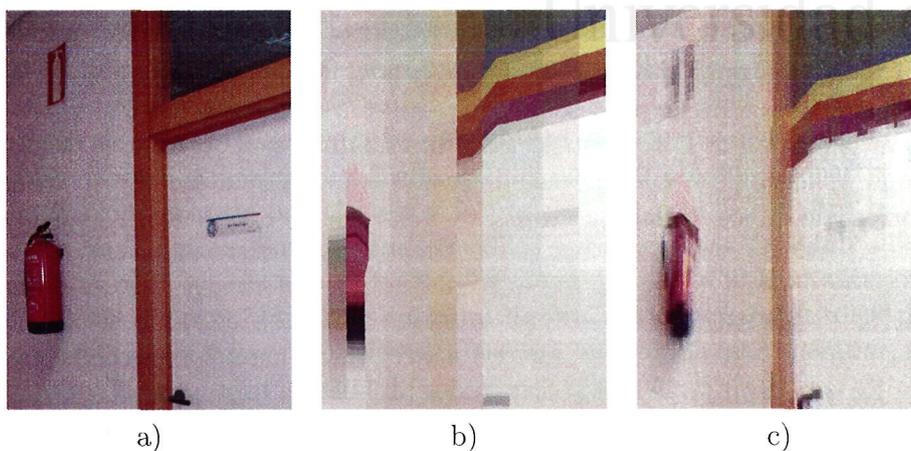


Figura 6.23: a) Imagen real del entorno. b,c) Reconstrucciones obtenidas sin y con fusión de unidades de color.

Tal y como era de esperar los resultados de todos los métodos aquí tratados confirman que se obtiene mejora al fusionar unidades de color para incrementar la calidad de la representación.

6.2.6. Discusión

En esta sección se ha aportado un modelo para la reconstrucción de mapas 3D para un robot autónomo. El modelo del entorno del que dispone el robot puede ser incompleto y por tanto debe poder trabajar teniendo en cuenta la incertidumbre. El sistema propuesto se basa en Unidades Bayesianas. Una unidad Bayesiana es un modelo de fusión, que tiene en cuenta explícitamente la incertidumbre, basado en la programación Bayesiana.

Se ha proporcionado un proceso de reconstrucción de mapas en 3D utilizando un láser y un sistema de omnivisión situados ambos en la parte superior del robot. El proceso de reconstrucción del mapa se divide en distintas fases o tareas, donde cada una de estas tareas corresponde a una unidad de proceso Bayesiana. En una primera fase se genera una malla 3D del entorno obtenida a partir de la información recogida por el láser y la posición del robot. Por otro lado, mediante las imágenes panorámicas obtenidas por el sistema de omnivisión se asigna un color a cada uno de los puntos obtenidos por la



unidad anterior.

Se ha utilizado el operador de fusión competitivo para realizar la fusión de varias unidades de proceso de color. De esta manera se ha conseguido mejorar la calidad de la reconstrucción. Para finalizar se muestran los resultados de la aplicación de los procesos de fusión aquí presentados que verifican la corrección de los mismos.

6.3. Resumen

En este capítulo se han aportado dos pruebas de concepto del operador de fusión competitiva. Más concretamente hemos proporcionado una aplicación del operador de fusión competitiva que permite incluir un sensor, del cual no se tenía información relativa a su uso junto con el robot, a partir de otro sensor de distinto tipo. También se ha aportado un modelo para la reconstrucción de mapas 3D utilizando un robot autónomo.

En la primera prueba de concepto se ha revisado la independencia condicional del operador de fusión competitiva proporcionando de esta manera un mecanismo para aprender el uso de un nuevo sensor mediante aprendizaje por refuerzo. Este método se puede utilizar o bien para aprender como tratar con un nuevo sensor que sustituirá al anterior o bien para aprender un nuevo sensor que complementará las observaciones del primero. Más concretamente se ha diseñado una conducta de evitación de obstáculos mediante ultrasonidos. Aplicando el operador de fusión competitivo se ha conseguido que un sensor visual sea capaz de desarrollar la misma tarea sustituyendo o complementado al sensor de ultrasonidos.

Utilizar el operador de fusión competitiva para aprender un nuevo sensor que complementará las acciones de otro conlleva una serie de ventajas: por una parte se mejora la robustez del sistema dada la redundancia intrínseca que capacita al sistema para proporcionar información incluso en el caso de fallos. Por otra parte, al aumentar la dimensionalidad del espacio de medidas, conseguimos un sistema más inmune al ruido. Al incluir sensores de distinto tipo, el ruido que afecta a uno no tiene por qué afectar al otro disminuyendo de esta manera la sensibilidad del sistema a las interferencias.

En la segunda prueba de concepto se ha proporcionado un modelo para la reconstrucción de mapas 3D utilizando un robot autónomo. El modelo del entorno del que dispone el robot puede ser incompleto y por tanto debe poder trabajar teniendo en cuenta la incertidumbre. Este modelo especifica



CAPÍTULO 6. CASO DE ESTUDIO: FUSIÓN BAYESIANA COMPETITIVA

235

un proceso de reconstrucción de mapas en 3D utilizando un láser y un sistema de omnivisión situados ambos en la parte superior del robot. El proceso de reconstrucción del mapa se divide en distintas fases o tareas, donde cada una de estas tareas corresponde a una unidad de proceso Bayesiana. En una primera fase se genera una malla 3D del entorno obtenida a partir de la información recogida por el láser y la posición del robot. Por otro lado, mediante las imágenes panorámicas obtenidas por el sistema de omnivisión, se asigna un color a cada uno de los puntos obtenidos por la unidad anterior. Para realizar la fusión de varias unidades de proceso de color se utiliza un operador de fusión competitivo. De esta manera, utilizando este operador, se ha conseguido una mejor calidad de reconstrucción.





Universitat d'Alacant
Universidad de Alicante

“La conclusión es que sabemos muy poco y sin embargo es asombroso lo mucho que conocemos.”

BERTRAND RUSSELL

Capítulo 7

Conclusiones y líneas futuras

Actualmente en la robótica los métodos y las arquitecturas de fusión sensorial son muy importantes a la hora de desarrollar sistemas robóticos, siendo uno de los pilares fundamentales para el diseño de robots autónomos. Tanto la cantidad como la diversidad de los sensores utilizados requieren el uso de estas técnicas disfrutando de beneficios respecto a los sistemas tradicionales, como una mayor robustez o fiabilidad.

Esta tesis se enmarca dentro del campo de la fusión sensorial robótica. Su principal objetivo es proporcionar un modelo de fusión capaz de tratar con información incompleta o incierta, que disponga de capacidades de aprendizaje y a ser posible que mantenga una representatividad uniforme para la fusión de bajo y alto nivel.

Como paso previo a la consecución del modelo se ha realizado un recorrido sobre las arquitecturas más utilizadas actualmente para la formalización de sistemas que requieran procesos de fusión sensorial. Se han revisado las arquitecturas centradas en la información, las funcionales y las basadas en comportamientos y se ha constatado que adolecen de diversos inconvenientes. De la misma manera también se han analizado las arquitecturas de fusión híbridas, que surgen de ideas de los modelos de fusión anteriores intentando mejorar o tratar aspectos no contemplados previamente. Estas arquitecturas aunque mejoran ciertos aspectos de las arquitecturas anteriores también adolecen de inconvenientes, como la baja representatividad al tratar con información incompleta e incierta o la necesidad de una unificación de las tareas



de fusión de bajo y alto nivel.

Metodológicamente también se ha analizado uno de los formalismos Bayesianos más completos para tratar con información incompleta o incierta, la programación Bayesiana, revisando sus ventajas al tratar con este tipo de datos. Utilizando este formalismo se han revisado algunos de los modelos probabilísticos más utilizados en el campo de la robótica. El formalismo de la programación Bayesiana se basa en la inferencia y aprendizaje Bayesiano. Se ha visto como un programa Bayesiano es un medio de especificar una familia de distribuciones que requiere dos capacidades, inferencia Bayesiana y toma de decisiones. Se han revisado numerosas técnicas que permiten relajar el problema de inferencia exacta para poder obtener tiempos de computación razonables.

Se ha aportado el modelo para la fusión robótica FSR-BAY, tomando como base el estado del arte de las arquitecturas de fusión y utilizando el formalismo de la programación Bayesiana. Este modelo disfruta principalmente de las siguientes ventajas respecto a los modelos de fusión anteriores:

- **Tiene capacidad para tratar con información incompleta e incierta.** Este modelo está basado en el formalismo de la programación Bayesiana y por tanto tiene en cuenta el tratamiento de la incertidumbre y la no completitud de la información.
- **Dispone de capacidades de aprendizaje.** La inferencia Bayesiana proporciona un mecanismo ideal para desarrollar tareas de aprendizaje utilizando este modelo. Al disponer de mecanismos de aprendizaje se consigue un modelo con más capacidad adaptación y por tanto más robusto.
- **Proporcionar varias herramientas para la especificación formal de los mecanismos de fusión.** El modelo FSR-BAY aporta dos operadores de combinación de unidades, el operador de fusión complementaria y el operador de fusión competitiva. El operador de fusión complementaria permite fusionar unidades que no dependen directamente entre ellas pero se pueden combinar para conseguir una imagen más completa del fenómeno a observar. El operador de fusión competitiva fusiona unidades que obtienen medidas independientes de una misma propiedad.
- **Permite una correcta descomposición de la información uti-**



CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS 239

lizando una representación uniforme independiente del nivel de fusión a utilizar. Este modelo se centra en la descomposición de la información en un conjunto de unidades de procesamiento Bayesiano, permitiendo también la combinación de unidades para formar una nueva unidad. Al utilizar estas unidades de procesamiento como único elemento descriptivo se consigue una representatividad uniforme para la fusión tanto de bajo como de alto nivel.

- **Disfruta de una base matemática robusta.** El modelo aquí presentado disfruta de una base matemática rigurosa, fundamentada en la inferencia Bayesiana.

Se ha aportado un estudio práctico del modelo propuesto mediante diversas pruebas de concepto orientadas al campo de la robótica móvil. Para desarrollar esta labor se han utilizado tanto simuladores (Saphira, Matlab) como robots reales (Pioneer I y Pioneer 3DX). A continuación se enumeran las aportaciones conseguidas en cada uno de estos casos de estudio:

- **Fusión Complementaria**

En el primer caso de estudio se ha definido una tarea de actuación para el robot suponiendo que este se moverá en un pasillo del Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante. El agente robótico ha servido de conexión entre los distintos despachos pudiéndose desplazar de un despacho a otro. Esta tarea se ha dividido convenientemente en subtareas. Para cada una de ellas se ha proporcionado un mapa Bayesiano que se ha encargado de desarrollar las labores iniciales de fusión de la información. Más concretamente se han aportado cuatro mapas Bayesianos (pasillo, puerta, final del pasillo y pared).

Se ha demostrado que un mapa Bayesiano es un tipo especial de unidad Bayesiana, y por tanto los operadores de fusión definidos para las unidades Bayesinas también se pueden aplicar a este tipo de mapas. Ya definidos los mapas anteriores se ha procedido a su fusión aplicando uno de los operadores de fusión complementaria definidos en el capítulo 4. De esta manera se ha obtenido un nuevo mapa que aglutina el conocimiento de los submapas que lo forman. Además se ha proporcionado una representación del mapa de localización resultante del



proceso de fusión utilizado un grafo inducido, que es la forma en la que tradicionalmente se representan los mapas.

Se ha comprobado el funcionamiento del mapa, también llamado unidad de localización, definiendo un ejemplo de conducta simple. Aunque los mapas Bayesianos (la unidad de localización) funcionan correctamente para la mayoría de las tareas, existen situaciones en las cuales el robot es incapaz de desenvolverse de manera correcta. Para aliviar este problema se ha proporcionado una nueva unidad que complementa a los mapas Bayesianos. De esta manera se utilizan dos unidades Bayesianas, la primera será la encargada de representar el espacio y generar comportamientos útiles (la unidad de localización), la segunda obtendrá los datos sensoriales y las salidas de la unidad de localización para retornar la acción a realizar por la capa física del robot (la unidad de ejecución). Esta última unidad proporciona un sistema más fiable completando a los mapas Bayesianos (la unidad de localización) en aquellas situaciones en las cuales el robot puede sufrir, por ejemplo, una colisión.

Dentro de este primer caso de estudio se ha aportado un mecanismo de detección de incoherencias utilizando el mapa Bayesiano propuesto. Al definir la fusión de sensores como un producto de términos más simples, relativos a cada sensor, se obtiene de manera sencilla la lectura esperada de un determinado sensor (de esta manera se puede verificar si un determinado sensor esta emitiendo lecturas incorrectas).

Por último se ha comparado el sistema Bayesiano con un sistema de control basado en lógica difusa. Aunque ambos sistemas se comportan de manera satisfactoria en condiciones normales no ocurre lo mismo cuando hay fallos en el sistema sensorial. El sistema Bayesiano aportado es capaz de detectar dichos fallos y actuar en consecuencia, anulando las lecturas de los sonares que no trabajan de manera coherente con el resto.

■ Fusión Competitiva

El segundo caso de estudio versa en la aplicación del operador de fusión competitiva en dos casos. Más concretamente se ha aportado una aplicación del operador de fusión competitiva que permite incluir un sensor, del cual no se tenía información relativa a su uso junto con el

robot, a partir de otro sensor de tipo distinto. También se ha proporcionado un modelo para la reconstrucción de mapas 3D utilizando un robot autónomo.

En la primera aplicación se ha revisado la independencia condicional del operador de fusión competitiva proporcionando de esta manera un mecanismo para aprender el uso de un nuevo sensor mediante aprendizaje por refuerzo. Este método se puede utilizar o bien para aprender como tratar con un nuevo sensor que sustituirá al anterior o bien para aprender un nuevo sensor que complementará las observaciones del primero. Más concretamente se ha diseñado una conducta de evitación de obstáculos mediante ultrasonidos. Aplicando el operador de fusión competitivo se ha conseguido que un sensor visual sea capaz de desarrollar la misma tarea substituyendo o complementado al sensor de ultrasonidos.

Utilizar el operador de fusión competitiva para aprender un nuevo sensor que complementará las acciones de otro conlleva una serie de ventajas: por una parte se mejora la robustez del sistema dada la redundancia intrínseca que capacita al sistema para proporcionar información incluso en el caso de fallos. Por otra parte, al aumentar la dimensionalidad del espacio de medidas, conseguimos un sistema más inmune al ruido. Al incluir sensores de distinto tipo, el ruido que afecta a uno no tiene porqué afectar al otro disminuyendo de esta manera la sensibilidad del sistema a las interferencias.

En la segunda aplicación del operador competitivo se ha proporcionado un modelo para la reconstrucción de mapas 3D utilizando un robot autónomo. El modelo del entorno del que dispone el robot puede ser incompleto y por tanto debe poder trabajar teniendo en cuenta la incertidumbre. Este modelo especifica un proceso de reconstrucción de mapas en 3D utilizando un láser y un sistema de omnivisión situados ambos en la parte superior del robot. El proceso de reconstrucción del mapa se divide en distintas fases o tareas, donde cada una de estas tareas corresponde a una unidad de proceso Bayesiana. En una primera fase se genera una malla 3D del entorno obtenida a partir de la información recogida por el láser y la posición del robot. Por otro lado, mediante las imágenes panorámicas obtenidas por el sistema de omnivisión se asigna un color a cada uno de los puntos obtenidos por la unidad anterior. Para realizar la fusión de varias unidades de proceso

de color se utiliza un operador de fusión competitivo. Utilizando este operador se ha conseguido una mejor calidad de reconstrucción.

Se han publicado distintos aspectos desarrollados en esta tesis. A continuación se proporcionan los artículos más relevantes:

- F. Aznar, M. Pujol, and R. Rizo, *Robot security and failure detection using bayesian fusion*, AI*IA 2005 - 9th Congress of the Italian Association for Artificial Intelligence. LNAI 3673-518. ISSN 0302-9743 (2005), p. 518–521.
- F. Aznar, M. Pujol, and R. Rizo, *Obtaining a bayesian map for data fusion and failure detection under uncertainty*, In Proceedings of the 18th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems. LNAI 3533-342. ISSN 0302-9743 (2005), p. 342–352.
- F. Aznar, M. Pujol, R. Rizo, and P. Suau, *A new artificial life formalization model: A worm with a bayesian brain*, KELSI 2004, LNAI 3303-124 (2004), p. 124-138.
- F. Aznar, M. Sempere, M. Pujol, R. Rizo, and R. Molina, *Generación de mapas 3d mediante fusión bayesiana de láser y visión omnidireccional.*, En actas de la Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA). CAEPIA'05, ISBN 84-96474-13-5 (2005).
- F. Aznar, M. Sempere, M. Pujol, and R. Rizo, *Bayesian emotions. Developing an interface for robot/human communication*, AI*IA 2005 - 9th Congress of the Italian Association for Artificial Intelligence. LNAI 3673-518. ISSN 0302-9743 (2005).
- F. Aznar, M. Sempere, M. Pujol, and R. Rizo, *A cognitive model for autonomous agents based on bayesian programming*, In Proceedings of Brain, Vision and Artificial intelligence BV&AI. LNAI 3704-277. ISSN 0302-9743 (2005), p. 277–287.
- F. Aznar, M. Sempere, M. Pujol, and R. Rizo, *Emergencia bayesiana en vida artificial: Una manera de tratar problemas complejos*, Simposio de Inteligencia Computacional, SICO 2005 (IEEE Computational Intelligence Society, SC) ISBN 84-9732-444-7 (2005).



CAPÍTULO 7. CONCLUSIONES Y LÍNEAS FUTURAS 243

- F. Aznar, M. Pujol, and R. Rizo, *Information fusion in Alife with Bayesian Programming*, International Transaction on Computer Science and Engineering, ISBN 89-953729-5-8 (2005), p. 47-61.
- F. Aznar, M. Sempere, M. Pujol, and R. Rizo, *Humanización de una interfaz robotica mediante unidades bayesianas*, VI Congreso de Interacción Persona Ordenador (AIPO 2005). ISBN 84-9732-436-6 (2005).
- F. Aznar, M. Pujol, and R. Rizo, *Specifying Complex Systems with Bayesian Programming. An Alife Application*, International Conference on Computational Science. ICCS 2005, LNCS 3514-828 ISSN 0302-9743 (2005), p. 828-836.

El trabajo futuro a desarrollar se enmarca en distintas líneas que se enumeran a continuación:

- **Operadores de fusión.** Una primera línea se basa en la investigación relativa a los operadores fusión. Aunque se han proporcionado dos operadores de fusión que intentan cubrir la mayoría de configuraciones posibles puede ser interesante proporcionar otros operadores, utilizando el mismo modelo, para tratar configuraciones especiales. Actualmente estamos trabajando en el desarrollo de un operador de fusión cooperativo.
- **Estudio de mecanismos para reducir la complejidad en unidades con un número de estados elevado.** Principalmente nos interesa reducir la complejidad de aplicación de los operadores de fusión en unidades con un elevado número de estados. La aplicación de los operadores de fusión puede requerir la marginalización de varias variables; sería deseable encontrar mecanismos para reducir la complejidad de este proceso.
- **Descomposiciones y capacidades de aprendizaje de las unidades Bayesianas.** Puede ser muy interesante el determinar descomposiciones de las variables de entrada que se adapten de manera más apropiada a tareas concretas. Una línea de investigación que surge a partir de esta tesis es buscar un método de selección guiado o incluso autónomo para realizar dichas descomposiciones. Otro aspecto con el



que estamos interesados es la realización de un análisis para determinar cómo afecta el número de estados de una unidad a su capacidad de aprendizaje.

- **Desarrollar sistemas de tiempo real.** También estamos interesados en desarrollar sistemas de tiempo real, en nuestro caso adaptados al campo de la robótica móvil, adaptando el modelo aquí propuesto a requerimientos temporales.
- **Fusión de comportamientos.** Otra línea, con la que estamos trabajando actualmente, versa en la utilización del modelo FSR-BAY para desarrollar tareas de fusión de comportamientos. Dentro de esta línea se situaría el desarrollo de un modelo robótico, que no disponga únicamente de planes y tareas sino que sea capaz de emular emociones humanas tal y como el cansancio o el miedo e incluso tenga cierto nivel de *consciencia*. El dotar a un robot con emociones nos permitirá por una parte planificar comportamientos desde un punto de vista distinto a las arquitecturas actuales robóticas además de proporcionarnos un mecanismo ideal para generar interfaces de interacción hombre/máquina. Como primera aproximación a este modelo cognitivo se ha publicado el artículo [ASPR05].



Universitat d'Alacant
Universidad de Alicante

Bibliografía

- [ACF⁺98] R. ALAMI, R. CHATILA, M. FLEURY, M. CHALLAB Y F. IN-GRAND. An architecture for autonomy. *Journal of Robotics Research (Special Issue on "Integrated Architectures for Robot Control and Programming")* (1998).
- [AJ00] S. AJI Y MCELIECE R. J. The generalized distributive law. *IEEE Transaction on Information Theory* **46**(2) (2000).
- [APRS04] F. AZNAR, M. PUJOL, R. RIZO Y P. SUAU. A new artificial life formalization model: A worm with a bayesian brain. *KELSI 2004, LNAI 3303, pp. 124-138* (2004).
- [Ark89] R.C ARKIN. Motor schema-based mobile robot navigation. *The International Journal of Robotics Research* (1989).
- [Ark97] T.R. ARKIN, R.C.AND BALCH. Aura: principles and practice in review. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)* (1997).
- [ASPR05] F. AZNAR, M. SEMPERE, M. PUJOL Y R. RIZO. A cognitive model for autonomous agents based on bayesian programming. *In Proceedings of Brain, Vision and Artificial intelligence BV&AI* (2005).
- [AZ03] ASHRAF ABOSHOSHA Y ANDREAS ZELL. Robust mapping and path planning for indoor robots based on sensor integration of sonar and a 2d laser range finder. *Proceedings of IEEE 7th International Conference on Intelligent Engineering Systems* (2003).
- [BB03] D. BELLOT Y P. BESSIERE. Approximate discrete probability distribution representation using a multi-resolution binary tree.



- 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03)* página 498 (2003).
- [BCS98] M. BORSOTTI, P. CAMPADELLI Y R. SCHETTINI. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters, Elsevier Science B.V.* **19**, 741–747 (1998).
- [Bed99] J. BEDWORTH, M. D. & O'BRIEN. The omnibus model: A new architecture for data fusion? *Proceedings of the 2nd International Conference on Information Fusion (FUSION '99)* (July 1999).
- [Ber97] A. BERTHOZ. "Le sens du mouvement". Odile Jacob (1997).
- [BFG⁺97] R. P. BONASSO, J. FIRBY, E. GAT, D. KORTENKAMP, D. P. MILLER Y M. G. SLACK. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental & Theoretical Artificial Intelligence* (1997).
- [BG03] PIERRE BESSIÈRE Y INRIA RESEARCH GROUP. Survey: probabilistic methodology and techniques for artefact conception and development. *INRIA* (2003).
- [BM58] G. E. P. BOX Y MULLER. A note on the generation of normal random deviates. *Annals Math. Stat.* **29**, 610 (1958).
- [Boy87] J. R. BOYD. A discourse on winning and losing. *Unpublished set of briefinf slides, Air University Library, Maxwell AFB* (1987).
- [Bro86] R. A. BROOKS. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* (1986).
- [Bro91] R. A. BROOKS. "Architectures for Intelligence". Lawrence Erlbaum Associates, Hillsdale, NJ (1991).
- [BS98] R.R. BROOKS Y IYENGAR S.S. "Multi-Sensor Fusion: Fundamentals and Applications". Prentice Hall, New Jersey (1998).
- [CB01] CHRISTOPHE COUÉ Y PIERRE BESSIÈRE. Chasing an elusive target with a mobile robot. *IEEE/RSJ. Int. Conference on Intelligent Robots and Systems (IROS)* (2001).



BIBLIOGRAFÍA

- [CFBE02] C. COUÉ, TH. FRAICHARD, P. BESSIÈRE Y MAZER E. Using bayesian programming for multi-sensor data fusion in automotive applications. *IEEE Intelligent Vehicle Symposium* (2002).
- [Cha02] J.P. CHANGEUX. "L'Homme de vérité". Odile Jacob (2002).
- [Coo90] GREGORY F. COOPER. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence* **42**(2-3), 393–405 (March 1990).
- [Das97] B. V. DASARATHY. Sensor fusion potential exploitation- innovative architectures and illustrative applications. *Proceedings of the IEEE* **85**(24-38) (1997).
- [DBM03a] J. DIARD, P. BESSIÈRE Y E. MAZER. A survey of probabilistic models using the bayesian programming methodology as a unifying framework. In *Proceedings International Conference on Computational Intelligence, Robotics and Autonomous Systems (IEEE-CIRAS), Singapore* (2003).
- [DBM03b] JULIEN DIARD, PIERRE BESSIÈRE Y EMMANUEL MAZER. Combining probabilistic models of space for mobile robots: the bayesian map and the superposition operator. *Proc. of the Int. Advanced Robotics Programme. Int. Workshop on Service, Assistive and Personal Robots. Technical Challenges and Real World Application Perspectives p. 65-72, Madrid (ES)* (October, 2003).
- [DBM04a] JULIEN DIARD, PIERRE BESSIÈRE Y EMMANUEL MAZER. Hierarchies of probabilistic models of navigation: the bayesian map and the abstraction operator. *Proceedings of the 2004 IEEE, International Conference on Robotics & Automation. New Orleans, LA* (April 2004).
- [DBM04b] JULIEN DIARD, PIERRE BESSIÈRE Y EMMANUEL MAZER. A theoretical comparison of probabilistic and biomimetic models of mobile robot navigation. *Proceedings of the 2004 IEEE, International Conference on Robotics & Automation. New Orleans, LA* (April 2004).



- [DK89] T. DEAN Y K. KANAZAWA. A model for reasoning about persistence and causation. *Computational Intelligence* **5**, 142–150 (1989).
- [DL93] P. DAGUM Y M LUBY. Approximate probabilistic reasoning in bayesian belief networks is np-hard. *Artificial Intelligence* **60**, 141–153 (1993).
- [DLR77] A. DEMPSTER, N. LAIRD Y D. RUBIN. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B.* **39**, 1–38 (1977).
- [DO00] J. DIARD Y LEBELTEL O. Bayesian programming and hierarchical learning in robotics. *SAB2000 Proceedings Supplement Book; Publication of the International Society for Adaptive Behavior, Honolulu* (2000).
- [DW88] H. F. DURRANT-WHYTE. Sensor models and multisensor integration. *The International Journal of Robotics Research* **6**(7), 97–113 (1988).
- [EHH01] W ELMENREICH, W HAIDINGER Y KOPETZM H. Interface design for smart transducers. *IEEE Instrumentation and Measurement Technology Conference* **3**, 1642–1647 (2001).
- [Elm02] WILDRIED ELMENREICH. “Sensor Fusion in Time-Triggered Systems”. Tesis Doctoral, Institut Für Technische Informatik 182 (October 2002).
- [FH95] K. E. FOOTE Y D. J. HUEBNER. Error, accuracy and precision. Informe técnico, Department of Geography, University of Texas (1995).
- [Fil01] D. FILLIAT. “Cartographie et estimation globale de la position pour un robot mobile autonome”. Tesis Doctoral, Universidad de París (2001).
- [For73] G. DAVID FORNEY. The viterbi algorithm. *Proceedings of The IEEE* **61**(3), 268–278 (1973).



BIBLIOGRAFÍA

249

- [Fow79] C. A. FOWLER. Comments on the cost and performance of military systems. *IEEE Transaction on Aerospace and Electronic Systems* (1979).
- [Hub99] MARCUS J. HUBER. Jam: a bdi-theoretic mobile agent architecture. En "AGENTS '99: Proceedings of the third annual conference on Autonomous Agents", páginas 236–243, New York, NY, USA (1999). ACM Press.
- [IC93] F. F. INGRAND Y V. COUTANCE. Real-time reasoning using procedural reasoning (93104). Informe t'ecnico, LAAS (1993).
- [IGR92] F. F. INGRAND, M. P. GEORGEFF Y A. S. RAO. An architecture for real-time reasoning and system control. *IEEE Expert Magazine* páginas 33–44 (1992).
- [Jay95] E. T. JAYNES. Probability theory, the logic of science. Disponible en: <http://bayes.wustl.edu/etj/prob.html> (1995).
- [JGJS99] MICHAEL I. JORDAN, ZOUBIN GHAHRAMANI, TOMMI JAAKKOLA Y LAWRENCE K. SAUL. An introduction to variational methods for graphical models. *Machine Learning* **37**(2), 183–233 (1999).
- [JK95] H. JEON Y B. K. J. KIM. Study on world map building for mobile robots with tri-aural ultrasonic sensor system. *IEEE International Conference on Robotics and Automation* páginas 2907–2912 (1995).
- [JLO90] F. V. JENSEN, S. L. LAURITZEN Y K. G. OLESEN. Bayesian updating in recursive graphical models by local computations. *Computational Statistical Quarterly* páginas 269–282 (1990).
- [JR94] M.I. JORDAN Y JACOBS R.A. Hierarchical mixtures of experts and the em algorithm. *Neural Computation* **6**, 181–214 (1994).
- [JW02] M.I. JORDAN Y Y. WEISS. "Handbook of Neural Networks and Brain Theory", capítulo Graphical models: probabilistic inference. MIT Press (2002).



- [Kal60] R.E. KALMAN. A new approach to linear prediction problems. *Journal of Basic Engineering* **35** (1960).
- [KB90] R. KUC Y BARSHAN B. Differentiating sonar reflections from corners and planes by employing an intelligent sensor. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)* **12**, 560–569 (1990).
- [KMF00] MIECZYSLAW M. KOKAR, MARK D. MEDWORTH Y CARL B. FRANKEL. A reference model for data fusion systems. *SPIE Conference on Sensor Fusion: Architectures, Algorithms, and Applications IV*. páginas 191–202 (July 2000).
- [KMRS97] K. KONOLIGE, K. L. MYERS, E. H. RUSPINI Y A. SAFFIOTTI. The saphira architecture: A design for autonomy. *Journal of experimental & theoretical artificial intelligence (JETAI)* (1997).
- [KPBM03] C. KOIKE, C. PRADALIER, P. BESSIERE Y E. MAZER. Proscriptive bayesian programming application for collision avoidance. *Proc. of the IEEE-RSJ Int. Conf. on Intelligent Robots and Systems (IROS); Las Vegas, USA* (2003).
- [LBDM04] OLIVIER LEBELTEL, PIERE BESSIÈRE, JULIEN DIARD Y EM-MANUEL MAZER. Bayesian robots programming. *Autonomous Robots* **16**(1), 49–79 (2004).
- [LBR+04] JAMES LLINAS, CHRISTOPHER BOWMAN, GALLINA ROGOVA, ALAN STEINBERG, ED WALTZ Y FRANK WHITE. Revisiting the jdl data fusion model ii. *Proceedings of the Seventh International Conference on Information Fusion (2004)* (2004).
- [LH98] J. LLINAS Y D. L. HALL. An introduction to multi-sensor data fusion. *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems* (1998).
- [LHABL04] R. LE HY, A. ARRIGONI, P. BESSIÈRE Y O. LEBELTEL. Teaching bayesian behaviours to video game characters. *Robotics and Autonomous Systems (Elsevier) in press* (2004).



- [LY94] J. LIU Y Y. YANG. Multiresolution color image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **16**, 689–700 (1994).
- [MHB⁺97] M. MARKIN, C. HARRIS, M. BERNHARDT, J. AUSTIN, M. BERWORTH, P. GREENWAY, R. JOHNSTON, A. LITTLE Y D. LOEW. Technology foresight on data fusion and data processing. *The Royal Aeronautical Society* (1997).
- [MJ96] M. MELIA Y M.I. JORDAN. “Advances in Neural Information Processing Systems”, capítulo Learning Fine Motion by Markov Mixtures Experts. MIT Press (1996).
- [MK97] G. MCLACHLAN Y T. KRISHNAN. The em algorithm and extensions. *Wiley series in probability and statistics* (1997).
- [MRR⁺53] N. METROPOLIS, A. W. ROSENBLUTH, M. Ñ. ROSENBLUTH, A. H. TELLER Y E. TELLER. Equations of state calculations by fast computing machine. *J. Chem. Phys.* **21**, 1087–1091 (1953).
- [Mur96] R.R. MURPHY. Biological and cognitive foundations of intelligent sensor fusion. *IEEE Transactions on Aerospace and Electronic Systems* **1**(26), 42–51 (1996).
- [Mur00] R. MURPHY. “An introduction to AI Robotics (intelligent robotics and autonomous agents)”. MIT Press (2000).
- [Mv01] MOREFIELD Y VARIOS. “Handbook of Multisensor Data Fusion”. CRC Press (2001).
- [Nea93] R. M. NEAL. Probabilistic inference using Markov chain Monte Carlo methods. Informe t'ecnico CRG-TR-93-1, University of Toronto, University of Toronto (1993).
- [NP80] P.J. NAHIN Y J. L. POKOSKI. Nctr plus sensor fusion equals iff n or can two plus two equal five? *IEEE Transactions on Aerospace and Electronic Systems* (1980).
- [PCM⁺99] J.A. PÉREZ, J.A. CASTELLANOS, J.M.M. MONTIEL, J. ÑEIRA Y J.D. TARDÓS. Continuous mobile robot localization: Vision vs. laser. *Proceedings of IEEE International Conference on Robotics & Automation* (1999).



- [Pea82] J. PEARL. Reverend bayes on inference engine: A distributed hierarchical approach. En "AAAI National Conference on AI", páginas 133–136 (1982).
- [Pea88] J. PEARL. "Probabilistic reasoning in intelligent systems: Networks of plausible inference". Morgan Kaufmann (1988).
- [Rao98] N. S. V. RAO. A fusion method that performs better than best sensor. *Proceedings of the First International Conference on Multisource-Multisensor Information Fusion* páginas 19–26 (1998).
- [RN96] STUART RUSSELL Y PETER NORVIG. "Inteligencia Artificial, Un enfoque moderno". Prentice Hall (1996).
- [Rob65] J. A. ROBINSON. A machine-oriented logic based on the resolution principle. *J. ACM* **12**(1), 23–41 (1965).
- [Rob79] J. A. ROBINSON. "Logic: Form and Function". Elsevier North-Holland Inc., New York (1979).
- [SBW99] A.Ñ. STEINBERG, C BOWMAN Y F.E. WHITE. Revisions to the jdl data fusion process model. *Proceedings of the 1999 IRIS Unclassified National Sensor and Data Fusion (NSSDF)* (May 1999).
- [SRK97] A. SAFFIOTTI, E. H. RUSPINI Y K. KONOLIGE. "Using fuzzy logic for mobile robot control". Kluwer Academic Publishers Group, Norwell, MA, USA, and Dordrecht, The Netherlands (1997).
- [Sto00] P. STOCKMEYER. "Handbook of Discrete and combinatorial mathematics", capítulo Enumerating graphs, páginas 580–586. CRC Press (2000).
- [TBF00] SEBASTIAN THRUN, WOLFRAM BURGARD Y DIETER FOX. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. *Proceedings of IEEE International Conference on Robotics and Automation* (2000).



- [Thr02] S. THRUN. Particle filters in robotics. En "Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)" (2002).
- [Var99] VARIOS. "Practical applications of fuzzy technologies". Kluwer (1999).
- [VG99] A. VISSER Y F. C. A. GROEN. "Organisation and Design of Autonomous Systems". Faculty of Mathematics, Computer Science, Physics and Astronomy, University of Amsterdam, Kruislaan 403, NL-1098 SJ Amsterdam (1999).
- [Wal90] J. WALTZ, E. & LLINAS. "Multisensor Data Fusion". Artech House (1990).
- [Wal01] L. WALD. The jdl definition of information fusion. Informe t'ecnico, Armines / Ecole des Mines de Paris (see in <http://www.data-fusion.org/article.php?sid=70>) (2001).
- [Wee99] ANDREW WEEB. "Statistical Pattern Recognition". Oxford University Press (1999).
- [YFW02] J. YEDIDIA, W. FREEMAN Y Y. WEISS. Constructing free energy approximations and generalized belief propagation algorithms. *MERL corporation technical report TR2002-35* (2002).
- [Zad65] L. A. ZADEH. Fuzzy sets. *Information and Control* **10**, 338-353 (1965).
- [Zim01] H.-J. ZIMMERMANN. "Fuzzy Set Theory And Its Applications". Kluwer (2001).



Universitat d'Alacant
Universidad de Alicante

Índice alfabético

- 3T, arquitectura, 59
- Arquitecturas Fusión
 - Basadas en el comportamiento, 49–52
 - Centradas en la información, 40–44
 - Funcionales, 44–49
 - Híbridas, 53–63
- AuRA, arquitectura, 54
- Boyd, Modelo, 44
- Cascada, modelo, 43
- Categorización, arquitectura, 49
- Cobertura espacial, 28
- Cobertura temporal, 28
- Conexión Bayesiana de Unidades de Procesamiento, 109
- Continuo, 27
- Correlación, 230
- Detección de incoherencias, 180
- Determinista, 27
- Dinámico, 27
- Distribución, 75
 - Modas, 78
- distributiva General, Ley, 78
- Episódico, 27
- Estimación Bayesiana recursiva, 87
 - filtros Bayesianos, 88
 - filtros de Kalman, 90
 - modelos ocultos de Markov, 89
- FSR-BAY, Modelo, 101–130
- Fusión
 - categorización a tres niveles, 38
 - clasificación basada en la configuración sensorial, 39
 - clasificación basada en la entrada/salida, 38
 - competitiva, 39
 - complementaria, 39
 - cooperativa, 39
 - limites, 31
 - ventajas, 29
- Fusión Competitiva heterogénea, operador, 121
- Fusión Competitiva Homegenea, operador, 127
- Fusión Complementaria, operador, 112
- Gibbs, muestreo, 81
- Imprecisión, 28
- Incertidumbre, 28
- Inferencia Bayesiana, 74–85
- Innacesible, 27
- Integración
 - analítica, 83
 - Monte Carlo, numérica, 83



ÍNDICE ALFABÉTICO

- Monte Carlo, simple (perfecta),
 84
- JDL, modelo, 41
- JLO, Algoritmo, 77
- láser, Sick LMS 220, 215
- LAAS, arquitectura, 53
- Mapas Bayesianos, 93–99
 abstracción, operador, 99
 superposición, operador, 99
- Mapas, Construcción, 214
- Marginalización de variables, 72, 76,
 82
- Matlab, 135
- Metrópolis, muestreo, 81
- Modularidad, 105
- Monte Carlo, métodos, 80
- Motora, arquitectura, 52
- Muestreo
 directo, 78
 indirecto, 78
- Omnibus, modelo, 48
- Perceptual Aliasing, 209
- Pioneer 3DX, 197
- Pioneer I, 134
- Privación Sensorial, 28
- Programación Bayesiana, 67–99
 conocimiento previo (π), 71
 datos experimentales (δ), 73
 descripción, 72
 preguntas, 74
 proposiciones lógicas, 70
 variables discretas, 70
- Redes Bayesianas, 85
- Redes Bayesianas dinámicas, 86
- Robustez, 30
- Saphira, 135
- Saphira, arquitectura, 61
- Saphira, simulador visual, 197
- SFX, arquitectura, 56
- Unidad Bayesiana de rango, 199
- Unidad Bayesiana para la obten-
 ción de puntos 3D, 217
- Unidad Bayesiana para la obten-
 ción del color de un punto,
 218
- Unidad Bayesiana Visual, 201
- Unidad de ejecución, 177
- Unidad de localización, 136
- Unidad de Procesamiento Bayesia-
 no, 107