

# Un sistema multi-agente que emplea visión activa y ultrasonidos aplicado a navegación con comportamientos difusos \*

Eugenio Aguirre, Moisés Gómez, Rafael Muñoz, Carmen Ruiz  
Departamento de Ciencias de la Computación e Inteligencia Artificial  
E.T.S. de Ingeniería Informática. Universidad de Granada  
18071-Granada tfno.: +34.958.240807 fax: +34.958.243317  
e-mail: eaguirre@decsai.ugr.es, moi\_gomez@hotmail.com

## Resumen

En los enfoques de navegación de robots que emplean mapas topológicos y llevan a cabo una navegación basada en comportamientos, la generación del plan de navegación puede llegar a estar muy relacionada con las habilidades que posea el robot, ya que el éxito del plan depende en gran parte de dichas habilidades. En este trabajo presentamos un sistema multi-agente que utiliza tanto información visual como datos de los sensores de rango para asistir a un sistema de navegación basado en comportamientos difusos, proporcionándole la habilidad de encontrar las puertas de las habitaciones. Para ello se utiliza una marca artificial situada junto a la puerta, de modo que se pretende “fijar” la cámara en dicha marca y dirigir así el robot hacia ella, ya sea para salir o entrar en una habitación en un entorno de interiores. El sistema se ha implementado en un robot móvil Nomad 200 y se han realizado numerosos experimentos en un entorno real de interiores que avalan la utilidad del mismo.

## 1 Introducción

Dentro del ámbito de la navegación de robots móviles, los enfoques fundamentados en modelos topológicos del entorno (Kortenkamp y Weymouth 1994) cuentan con la ventaja de manipular la información a más alto nivel que los fundamentados en modelos geométricos (Elfes 1989) que, normalmente, precisan de una mayor complejidad computacional y de un mayor tiempo de reacción en función del modelo de representación utilizado. También los modelos geométricos suelen ser muy dependientes del sistema de localización empleado ya que necesitan conocer con cierta exactitud la posición del robot para tomar las decisiones oportunas en la navegación. Con respecto a los enfoques topológicos, la semántica asociada a los nodos y arcos del grafo topológico que representa el entorno no es siempre la misma. Mientras que en (Thrun 1998) el mapa topológico se obtiene a partir de la división en regiones de un mapa de celdas de ocupación, en (Kuipers y Byun 1991) los nodos representan lugares distinguidos desde el punto de vista sensorial y los arcos caminos entre estos lugares a los que se asocian unas estrategias de control concretas. En relación con la navegación basada en comportamientos (Arkin 1998), podemos encontrar enfoques que usan mapas topológicos orientados a la navegación basada en comportamientos, en unos casos (Mataric 1992) los nodos representan lugares distinguidos asociados a comportamientos mientras que en otros trabajos (Aguirre y González 2002) también los arcos indican el comportamiento a utilizar para navegar entre los nodos. En estos casos la navegación entre dos puntos se plantea como la ejecución de un plan compuesto por una serie de nodos o lugares unidos por arcos o comportamientos a ejecutar siendo los nodos o lugares sub-objetivos a alcanzar en el plan. Las habilidades del robot, como el seguimiento de paredes, pasillos, cruce de puertas y la evitación de obstáculos son fundamentales para conseguir el éxito del plan generado ya que éste, al no ser un plan de alto nivel de detalle, cuenta con las capacidades del robot para alcanzar cada sub-objetivo o lugar establecido en el plan. En estos últimos modelos el papel del sistema sensorial es muy importante ya que cada comportamiento relaciona percepción con acción siguiendo su propio objetivo de control por lo que en anteriores trabajos hemos desarrollado un modelo perceptual basado en ultrasonidos que permite distinguir objetos como paredes, pasillos, esquinas y los huecos de las puertas entre otros tipos de objetos (Aguirre et al. 2002).

Para mejorar las habilidades de nuestro sistema es necesario contar con la incorporación de información visual, la cual ha sido ampliamente utilizada en el mundo de la robótica como sensor muy importante en la

---

\*Este trabajo ha sido parcialmente financiado por el proyecto TIC2002-04146-C05-02 del MCYT

navegación de robots. Por ejemplo Kabuka y Arenas (Kabuka y Arenas 1987) utilizan un patrón circular para facilitar la tarea de que el robot se acerque a un lugar determinado. Zhang et al. (Zhang et al. 1993) utilizan los puntos de fuga de dos líneas paralelas cercanas y su orientación para ayudar a la navegación del robot. Otros enfoques abordan el problema de construir y mantener una representación en tres dimensiones del entorno a partir de imágenes planas, ámbito de la visión estereoscópica, de manera que existen técnicas y propuestas muy elaboradas en esta materia como se puede observar en el libro de Ayache (Ayache 1991) y en otros trabajos (Kriegman et al. 1989).

En este trabajo presentamos un sistema multi-agente que incorpora información visual a un sistema de navegación basado en comportamientos difusos y le proporciona una nueva habilidad consistente en la capacidad de encontrar las puertas de las habitaciones y dirigirse a ellas, ya sea para salir o entrar en la habitación. Esta nueva habilidad surge de la interacción de comportamientos que toman sus datos de entrada de los ultrasonidos con otros que utilizan solamente información extraída a partir del sistema de visión. Tantos unos como otros, se han diseñado utilizando reglas difusas lo que nos permite una manipulación adecuada de la imprecisión e incertidumbre subyacente en los datos sensoriales. La información visual se extrae de las imágenes tomadas con una sola cámara situada encima de una cabeza robotizada o unidad de pan-tilt (PTU) que permite giros en sentido horizontal y vertical de la cámara a la vez que el robot sigue su propio movimiento, por lo que la visión adopta un papel activo en el sistema a la hora de establecer la relación entre percepción y acción. Para facilitar la detección de las puertas se ha utilizado una marca artificial con dígitos numéricos que es reconocida por el robot después de aplicar un proceso de aprendizaje mediante redes neuronales. Una vez que el robot detecta la marca en la puerta, debe mantener la cámara mirando fijamente hacia la marca para indicar así la dirección de avance. De esta forma el robot es capaz de dirigirse hacia ella, evitando posibles obstáculos no esperados, llegar a la puerta, cruzarla y seguir su navegación por el entorno. Con respecto a las puertas se va a suponer en todo momento que éstas están abiertas, ya que nuestro robot no tendría la posibilidad de abrirlas por él mismo, además, en esta primera versión del trabajo se suministra al robot un mapa topológico del entorno para relacionar las marcas numéricas con las correspondientes habitaciones, pero en próximas versiones se plantea construir este mapa en una fase previa de exploración del entorno.

A continuación se describe la arquitectura del sistema multi-agente, realizando primero una breve descripción del hardware utilizado y luego una visión general de los agentes del sistema. Las siguientes secciones se dedican a profundizar en tres agentes concretos: El agente que realiza el procesamiento de las imágenes y control de la cámara, el que dirige la navegación y el que controla todo el sistema. En la sección de resultados experimentales se muestra el funcionamiento del sistema en una tarea de navegación en un entorno real y finalmente, se ofrecen algunas conclusiones y líneas de actuación para el futuro.

## 2 Arquitectura del Sistema Multi-Agente

### 2.1 Descripción del hardware

El problema planteado suponía la actualización hardware de un robot móvil de tipo Nomad 200 que contaba con un sistema de visión muy básico no apto para aplicaciones en tiempo real. Realmente, para realizar el procesamiento de las imágenes a cierta velocidad se necesita emplear una máquina de mayor potencia que la presente en el Nomad 200. En nuestro caso dicha máquina es un ordenador portátil Pentium III a 1.100 Mhz que se comunica por red ethernet a 10 Mbps vía TCP/IP con la CPU del robot y que se coloca encima de la torreta del Nomad 200. La CPU del robot se utiliza tan sólo para ejecutar un demonio sobre Linux que recibe los datos de los sensores y envía las órdenes de movimiento a los motores, mientras que todo el sistema de agentes se ejecutará en el portátil sobre windows 98, al que se ha conectado tanto la unidad PTU como una cámara de vídeo Sony EVI-401. Con respecto a la unidad PTU en horizontal puede realizar un giro de 139 grados para cada lado, mientras que en vertical puede girar 47 grados hacia abajo y 31 grados hacia arriba. Este nuevo sistema de visión nos permite realizar un procesamiento de imágenes a mayor velocidad que el que nos permitía el sistema básico del Nomad 200, pero a pesar de la mejora del hardware, es importante el uso de algoritmos rápidos en el procesamiento de las imágenes para conseguir que el robot sea capaz de mantener la cámara mirando a la marca. Por ello como veremos en la sección 3 se han utilizado técnicas de procesamiento de imágenes sencillas, pero rápidas y efectivas.

### 2.2 Descripción del Sistema Multi-Agente

El objetivo del sistema multi-agente es poder ayudar a la navegación en un entorno de interiores del cual se posee un mapa topológico como el mostrado en la figura 1 en el que existen una serie de habitaciones

conectadas por pasillos o puertas. En las puertas se ha situado convenientemente una marca artificial para facilitar su detección por el sistema de visión como se verá más adelante. Esta marca puede estar situada junto al lado derecho o izquierdo de la puerta, siempre relativo al sentido de avance para cruzar dicha puerta. Esta información aparece también en la figura 1 con los términos *Der* (lado derecho) e *Izq* (lado izquierdo) mientras que las puertas se representan con los  $P_i$ .

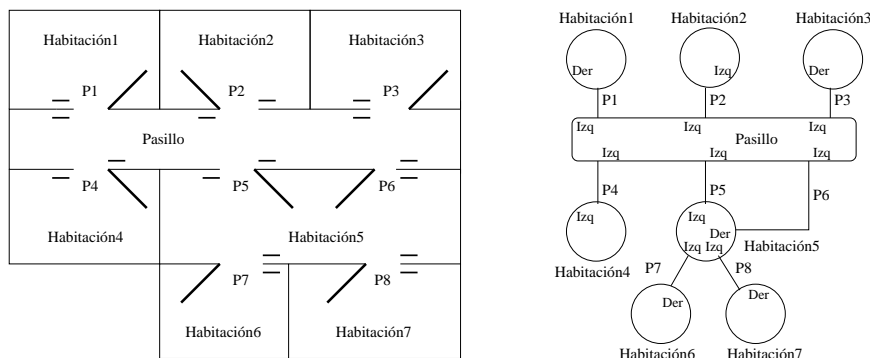


Figura 1: Entorno de interiores y su mapa topológico

En este contexto debemos aclarar que entendemos por agente un proceso software dirigido a alcanzar o mantener un objetivo implícito en su propio diseño, entendiendo este objetivo en el sentido más general del término. Hemos diseñado un sistema distribuido basado en agentes, entre los cuales encontramos agentes de tipo reactivo y también deliberativo. El sistema se ha diseñado de la forma más modular posible para facilitar su posterior expansión ya sea mediante la adición de nuevas competencias a un agente concreto o bien mediante la adición de nuevos agentes. El conjunto de agentes diseñado es el siguiente:

- **Planner:** Su objetivo es encontrar el camino hacia la meta final para lo cual utiliza el mapa topológico y un algoritmo de búsqueda de caminos mínimos ya que los arcos del grafo tiene pesos asociados a la dificultad del camino. También se encarga de la ejecución y monitorización del plan lo que implica la activación del resto de agentes para realizar la búsqueda de la marca situada en la puerta, aproximarse a ella, cruzarla y navegar por el entorno.
- **Navigation:** Este agente cuenta con diferentes comportamientos difusos cuya activación le permite navegar por el entorno siguiendo las paredes, evitar obstáculos, dirigirse hacia la marca de la puerta, cruzarla y navegar por un pasillo, entre otras habilidades.
- **Nomad200:** Este agente establece una comunicación via TCP/IP con el demonio que se ejecuta en el robot, de manera que cualquier otro agente que desee enviar alguna orden al robot deberá hacerlo a través de éste. El agente permite de esta forma el uso concurrente de los diferentes dispositivos del robot al resto de la comunidad y supone una abstracción de los detalles de programación del robot. Las órdenes que se pueden enviar al robot permiten configurar su velocidad y aceleración para desplazarlo por el entorno así como obtener los datos de los sensores de ultrasonidos e infrarrojos, entre otros.
- **PTU:** Este agente permite el uso de la PTU al resto de la comunidad de agentes. La ventaja de usar este agente es la posibilidad de encolar las órdenes de movimiento que se envían a la PTU por parte de otros agentes y controlar así el uso concurrente de la misma. El agente además permite conocer la posición de cada eje del instrumento mediante paso de mensajes.
- **Tracker:** Es el encargado de realizar el procesamiento de las imágenes para detectar la marca en el entorno y mover la PTU para mantener, siempre que sea posible, la cámara enfocando la marca, lo que va a permitir al robot dirigirse hacia la puerta.

En la figura 2, se muestra el conjunto de agentes empleado. Las líneas que unen los agentes implican que hay algún tipo de comunicación entre ellos o prestación de servicios de unos a otros. A continuación, nos centraremos en describir solamente en profundidad los agentes Tracker, Navigation y Planner a lo que dedicaremos las siguientes secciones.

### 3 Agente Tracker

Este agente está al cargo de detectar y hacer seguimiento de la marca concreta que se le indique por parte del agente Planner, en función del lugar donde se encuentre el robot en un momento dado. Este agente es capaz de detectar el conjunto de marcas que hay en una imagen concreta e informar sobre si una determinada marca se encuentra en dicha imagen. Cuando la marca ha sido identificada, puede realizar un seguimiento de la marca indicada valiéndose de la PTU, manteniendo la marca dentro del campo visual de la cámara. También puede informar sobre la distancia existente entre la marca y la cámara a fin de determinar si se está cerca de la puerta a cruzar. En resumen, este agente realiza tres funciones: Detección de la marca, seguimiento de la misma y medición de la distancia entre marca-cámara, las cuales se comentan a continuación.

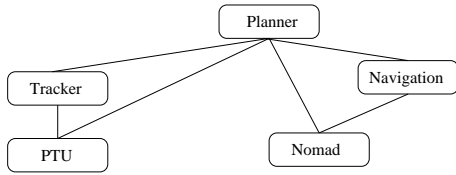


Figura 2: Estructura del sistema multi-agente.



Figura 3: Marca diseñada.

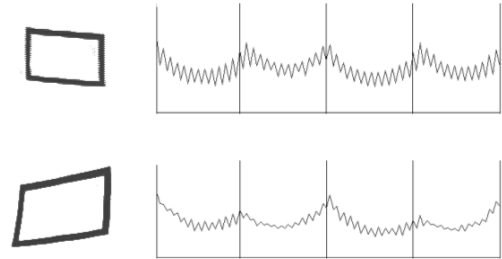


Figura 4: Ejemplos de la firma doble del borde rectangular.

#### 3.1 Detección de la marca

Para el diseño de la marca se han tenido en cuenta una serie de requisitos que son los siguientes: Debe ser fácilmente distinguible del entorno, debe permitir algún tipo de representación de información para identificar la puerta junto a la que se encuentra y debe ser fácil de construir con bajo coste. Cumpliendo estas condiciones, se decidió usar el modelo de marca que se muestra en la figura 3.

La marca tiene un borde rectangular externo negro para ser fácilmente distinguible del fondo y se han colocado números en el interior para identificar la puerta sobre la que está situada la marca. Las marcas han sido desarrolladas usando un editor simple y se ajustan al formato de folio A3.

El proceso de identificación de la marca puede ser dividido en dos partes. En primer lugar hay que realizar la detección de la misma, lo que se lleva a cabo identificando el borde rectangular de color negro, y en segundo lugar, identificando el número que hay en el interior del rectángulo. Para detectar la marca, el agente toma una imagen en niveles de gris y realiza un proceso de binarización. Para ello hemos usado el algoritmo de umbralización propuesto por (Otsu 1979) basado en el histograma. En primera instancia se probó a usar un método de umbralización estático (a valor fijo), pero debido a los cambios en la iluminación el proceso se hacía inestable. El método finalmente elegido selecciona un umbral basándose en la iluminación y tratando de minimizar la entropía de la imagen resultante y hacer así el proceso más robusto.

El siguiente paso es la identificación del borde rectangular externo de la marca. En primer lugar se utilizó para ello la técnica de la *firma*. La firma de un objeto es una función  $f(\phi)$  que indica la distancia desde el centro de un objeto en la imagen, hasta el punto más externo del mismo en una línea trazada desde el centro con un ángulo  $\phi$  (González 1996), para ángulos en un rango comprendido entre  $[0, 2\Pi]$  equiespaciados.

Pero debido a que la marca necesitaba ser reconocida desde diferentes puntos de vista y a la imposibilidad de que la firma simple representara la frontera más interna del rectángulo, se decidió emplear una variante. En lugar de la firma, se ha desarrollado en este trabajo una adaptación de este método, que llamaremos *firma doble* y que consiste en tomar tanto la distancia del centro al punto más lejano en la imagen, como la distancia al más cercano para ángulos equiespaciados en el intervalo  $[0, 2\Pi]$ . La figura 4 muestra la firma doble aplicada al borde rectangular en imágenes tomadas desde puntos de vista distintos.

Estas distancias se almacenan de forma consecutiva en un array para representar a los objetos que aparecen en la imagen tras la segmentación. Para aprender la firma doble del borde rectangular a partir de ejemplos, se ha probado tanto el uso de árboles de clasificación creados usando algoritmos genéticos, como redes neuronales.

En el primer método se crean árboles binarios de decisión, en que se toma como predicado de decisión inecuaciones que tienen como argumentos celdas del array,  $c_i$ , que contienen las distancias obtenidas mediante la firma doble. De esta forma podría tenerse como inecuación  $c_1 * 3 + 5 > c_{13}/8 + c_3$ . Si el objeto cumple la

inecuación se profundiza en una rama del árbol y si no, en la otra. De esta forma se llega finalmente a un nodo hoja donde está especificado si el objeto es un borde rectangular válido o no. Tanto las inequaciones como la estructura del árbol se crea aleatoriamente utilizando árboles genéticos. En el segundo método se ha usado una red neuronal con una capa oculta entrenada por el algoritmo de *backpropagation*. La tabla 1 muestra los resultados de las pruebas de clasificación del borde rectangular usando tanto los árboles de clasificación, como redes neuronales.

Método	Éxito en entrenamiento	Éxito en validación
Árbol de decisión	98.14	98.30
Red Neuronal	100	99.65

Tabla 1: Porcentajes de éxito de los métodos de aprendizaje.

Finalmente se ha optado por emplear la red neuronal al haber obtenido los mejores resultados.

Una vez identificado el borde rectangular, el siguiente paso es determinar el número que hay en el interior. Para ello creamos una imagen con cada dígito y extraemos las siguientes características:

1. **Derivadas horizontal y vertical.** El método consiste en construir una función con el número de pixels del dígito en cada fila y columna de la imagen creada. A esta función le aplicaremos la derivada y los valores obtenidos se escalan en el intervalo  $[-1, 1]$ .
2. **Suma por bloques.** Debido a que el método anterior se demostró insuficiente para la detección de los dígitos en imágenes tomadas a diferentes ángulos, ha sido necesario, además, el uso de otro método para la extracción de características adicionales. Hemos creado bloques 2x2 y se ha contado el número de pixels del dígito en cada bloque. Finalmente trasladamos los valores al intervalo  $[0, 1]$ .

Usando ambos métodos se obtiene al final un conjunto de características que han sido aprendidas mediante redes neuronales. De esta forma, se puede aprender cada posible dígito por separado y uniendo los dígitos reconocidos se reconstruye el identificador de la marca que consta de dos dígitos.

### 3.2 Seguimiento de la marca

Una vez que la marca deseada ha sido localizada se procede a su seguimiento, es decir, el objetivo es tener la marca siempre dentro de la siguiente imagen pese al movimiento del robot. Para ello el agente mueve la PTU para que la marca quede en el centro de la imagen. La información que se necesita para centrar la marca es el número de pixels que existe entre el centro de la marca y el centro de la imagen y a partir de ahí, habrá que establecer la correspondencia necesaria con el movimiento de la PTU que lleva el centro de la marca al centro de la imagen.

El tratamiento que se hace de la imagen en este caso difiere del necesario para la identificación de la marca ya que la rapidez es un elemento muy importante en este caso. Ahora, podemos suponer que la marca que seguimos está en el centro de la imagen actual y que en la siguiente imagen, la marca más cercana al centro será la que estamos siguiendo. Asumiendo esto, podemos reducir el procesamiento necesario para detectar la marca en la imagen y centrarnos sólo en la detección del borde externo de la marca. Ya que para este análisis se requiere menor precisión que para el caso de la identificación de los dígitos, podemos usar una imagen de menor tamaño aumentando así la velocidad de procesamiento. Esto nos permitirá hacer más ajustes de la posición de la cámara por unidad de tiempo y reducir la probabilidad de perder la marca del campo de visión.

Para determinar el movimiento en los ejes de la PTU se han creado dos rectas de regresión, una para cada eje, utilizando la técnica de mínimos cuadrados a partir de valores tomados experimentalmente. Las rectas de regresión son:

$$pan\_degrees = -25.99 \cdot \frac{2IncPixX}{Width} + 0.05; \quad tilt\_degrees = 17.63 \cdot \frac{2IncPixY}{Height} + 0.29$$

Donde  $IncPixX$ ,  $IncPixY$  son la distancia en pixels desde el centro de la marca al centro de la imagen en el eje  $X$  e  $Y$  respectivamente, y  $Width$ ,  $Height$  son el ancho y alto de la imagen.

### 3.3 Medición de la distancia marca-cámara

El agente Planner necesita conocer la distancia entre el robot y la marca para determinar si está lo suficientemente cerca de la puerta como para comenzar a cruzarla. Sería posible realizar esta medición usando los sensores de ultrasonidos, pero estaríamos sometidos a los problemas típicos como posibles falsas reflexiones en función del mobiliario cercano a la puerta, por lo que se ha optado por usar visión en este caso.

Asumiendo que el tamaño de la marca es conocido y fijo, podemos aproximar la distancia de la cámara a la marca contando el número de pixels que ocupa la proyección vertical de la marca en la imagen. A mayor distancia a la marca, menos pixels tendrá su proyección en la imagen y viceversa. Para ello, hemos creado una curva de regresión por mínimos cuadrados con mediciones reales que se puede observar en la figura 5.

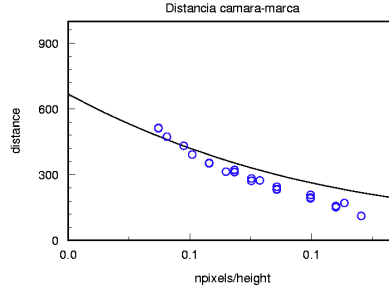


Figura 5: Curva de regresión distancia a la marca.

La curva finalmente obtenida es la siguiente:

$$D = 763.02 - 6652.21x + 23403.03x^2 - 29177.50x^3$$

siendo  $x = \frac{NoPixels}{Height}$ , donde *NoPixels* es el número de pixels de la proyección vertical de la marca en la imagen y *Height*, el número de pixels verticales total que tiene la imagen que se ha tomado con la cámara, es decir, el alto de la imagen. De esta forma,  $x$  queda acotada al intervalo  $[0, 1]$  y podemos usar la misma ecuación para imágenes de varios tamaños.

## 4 Agente Navigation

Como se comentó en la sección 2.2 este agente se ocupa de llevar a cabo la navegación del robot por el entorno, relacionando la percepción del robot con las órdenes que se deben enviar a los actuadores en función del objetivo que debe cumplir en cada momento. Para ello, hace uso de diferentes comportamientos que se explican a continuación, los cuales se han diseñado utilizando técnicas tomadas de la teoría de Control Difuso (Driankov et al. 1993) y siguiendo una metodología propia de construcción de las bases de reglas difusas que ha sido empleada con éxito en anteriores trabajos (Aguirre y González 2000). La utilización de estos comportamientos difusos permiten al robot el control de los actuadores a pesar de la presencia de ruidos en los datos de los sensores lo que provoca imprecisión en las medidas e incertidumbre sobre la certeza de la información manejada. A continuación, detallamos cómo se ha diseñado el comportamiento que utiliza la información suministrada por el agente Tracker para guiar al robot hacia la puerta, mientras que en el resto de comportamientos simplemente comentaremos brevemente su función. Para más detalles sobre la metodología de construcción de éstos y otros comportamientos difusos remitimos a (Aguirre y González 2000).

1. Dirigirse hacia la marca de la puerta. Este comportamiento permite que el robot se dirija hacia la marca de la puerta indicada, orientando las ruedas y la torreta hacia la dirección que le indica la PTU. Es necesario indicar que el hardware del robot nos permite mover independientemente la torreta por un lado y las ruedas por otro, lo cual nos da la posibilidad de poder mantener la torreta en una situación más favorable para no perder la visión de la marca de la puerta, cuando por ejemplo, debe evitarse algún obstáculo en el camino, lo que puede implicar girar las ruedas para evitar el choque. Por tanto, necesitamos realmente diseñar un conjunto de bases de reglas para orientar la torreta y otro para determinar la orientación de las ruedas, es decir, el movimiento del robot.
  - Base de reglas para orientar la torreta. La idea es mantener la torreta orientada hacia la marca, para ello, en función del ángulo de la PTU y la situación actual de la torreta, se calcula el ángulo

que debe girar la torreta para mantenerse alineada con la PTU. Llamaremos a este ángulo *Ángulo Mover Torreta* y se toma como variable de entrada en las reglas, mientras que la variable de control es la velocidad de rotación de la torreta, *VR Torreta*. La figura 6 muestra las etiquetas de la variable *Ángulo Mover Torreta* mientras que la figura 7 muestra las de *VR Torreta*. Asimismo, la tabla 2 muestra el conjunto de reglas.

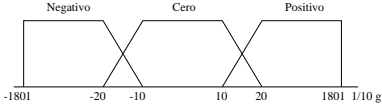


Figura 6: Ángulo Mover Torreta.

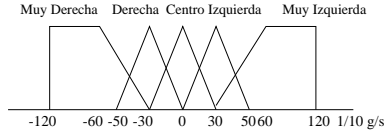


Figura 7: VR Torreta.

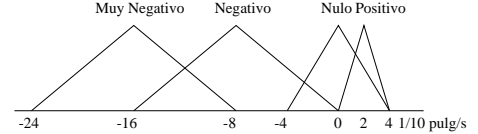


Figura 8:  $\Delta VT$ .

Ángulo Mover Torreta	VR Torreta
Negativo	Girar derecha
Cero	Centrado
Positivo	Girar izquierda

Tabla 2: Base de reglas para orientar la torreta

- Base de reglas para orientar las ruedas. La idea es semejante a la anterior pero ahora las variables a controlar serán la velocidad de rotación (*VR*) y traslación (*VT*) del robot con la finalidad de mantener orientadas las ruedas hacia la marca. Igual que antes, a partir del ángulo de la PTU y el ángulo actual de las ruedas se calcula el ángulo que deben moverse las ruedas para mantenerlas alineadas con la PTU, llamamos a este ángulo *Ángulo Mover Ruedas*. Para facilitar la alineación, la velocidad de traslación se disminuye cuando las ruedas y PTU están desalineadas y se incrementa una vez que están alineadas, siempre dentro de un rango determinado. A estos incrementos positivos o negativos los denominamos  $\Delta VT$ . Con respecto a las etiquetas de las variables, en el caso de *Ángulo Mover Ruedas* y *VR* coinciden con las mostradas en las figuras 6 y 7 respectivamente, mientras que la figura 8 muestra las etiquetas de  $\Delta VT$ .

La tabla 3 muestra las bases de reglas para el control de *VR* (tabla 3(a)) y  $\Delta VT$  (tabla 3(b)) en función del *Ángulo Mover Ruedas*.

Ángulo Mover Ruedas	VR
Negativo	Girar derecha
Cero	Centrado
Positivo	Girar izquierda

Tabla 3: (a) Base de reglas para VR

Ángulo Mover Ruedas	$\Delta VT$
Negativo	Incremento negativo
Cero	Incremento positivo
Positivo	Incremento negativo

(b) Base de reglas para  $\Delta VT$

2. Evitar obstáculo en una habitación. Este comportamiento permite al robot sortear un obstáculo que esté situado en su trayectoria. Las variables de entrada que se han considerado son: *Distancia Frontal* y *Distancia Derecha e Izquierda* detectadas por los ultrasonidos, junto a la *Diferencia entre Distancia Derecha e Izquierda* y velocidad de traslación *VT*. Las variables de control son la velocidad de rotación *VR* y traslación *VT*. La idea que gobierna el diseño de este comportamiento consiste en que si el obstáculo no está justo delante del robot entonces éste tiende a girar un poco a uno u otro lado según interese. Cuando el obstáculo está justo delante o el robot se ha acercado demasiado, se girará hacia donde haya mayor espacio libre. En cualquier caso, cuando se detecta un obstáculo se disminuye la velocidad de traslación.
3. Seguir la pared de una habitación. El objetivo de este comportamiento es permitir que el robot se mueva alrededor de la habitación siguiendo el contorno de la pared, mientras realiza una búsqueda en movimiento de la marca de la puerta. Las variables de entrada para el seguimiento, por ejemplo de la pared derecha, serían *Distancia a la Pared Derecha* y *Ángulo de Incidencia con la Pared Derecha*. La variable de salida es la velocidad de rotación de las ruedas *VR*.

4. Atravesar una puerta. Este comportamiento utiliza otros dos comportamiento simples: *cruzar puerta* y *evitar obstáculo en puerta*, ya que en este caso ha sido necesario especializar la evitación de obstáculos debido a las cortas distancias que se detectan respecto de los marcos de la puerta. Comentamos solamente *Cruzar puerta* ya que la filosofía de evitación de obstáculos es parecida a la anterior pero con otras etiquetas más adecuadas a este caso. El comportamiento *Cruzar puerta* permite al robot atravesar una puerta para entrar o salir de una habitación. Cuando el robot está muy cerca de los marcos de una puerta los datos de los ultrasonidos son muy inexactos, por lo que se trata de situar los marcos de la puerta cuando el robot está cerca de la misma pero las medidas son todavía fiables. De esta forma, se ha utilizado un modelo temporal de los marcos de la puerta con el objetivo de calcular una recta que pase por el centro del hueco. Luego, el robot simplemente tendrá que tomar como referencia dicha trayectoria. Como este proceso no está exento de ruido, es posible que la trayectoria deba ser reajustada según el acierto o desacierto de la actuación del robot. Las variables de entrada son la *Distancia a la Trayectoria* calculada y el *Ángulo de Desviación* respecto de la trayectoria. Como variable de salida se ha considerado la velocidad de rotación de las ruedas *VR*, mientras que la velocidad de traslación *VT* se disminuye a valores muy bajos.
5. Orientar las ruedas y la torreta en el pasillo. Cuando el robot ha terminado de cruzar la puerta, las ruedas y la torreta se mueven hasta alcanzar un determinado ángulo para alinear el robot con el pasillo. Las variables de entrada que se han considerado han sido los *Ángulos a Mover Ruedas y Torreta* que indican lo que se debe girar para alcanzar la posición deseada, y como variables de salida, las velocidades de rotación de las ruedas *VR* y de la torreta *VR Torreta*.
6. Mover el robot por un pasillo. Este comportamiento permite al robot recorrer un pasillo, avanzando por el centro del mismo. Las variables de entrada que se han considerados son la *Diferencia entre la Distancia Derecha y Distancia Izquierda* relativas a las paredes del pasillo y *Diferencia del Ángulo de Incidencia Derecho y Ángulo de Incidencia Izquierdo* con el objeto de controlar *VR*. Con respecto a *VT* se aumenta si el robot va alineado por el centro del pasillo y se disminuye en el resto de los casos.

Por último, destacar que este agente está constantemente esperando a recibir ordenes del agente Planner para activar el comportamiento correspondiente.

## 4.1 Agente Planner

Este agente se sitúa en el mayor nivel de abstracción del sistema y se encarga de ir activando el resto de agentes monitorizando la ejecución del plan, a fin de asegurar el correcto funcionamiento de todo el sistema. Para generar el plan, se utiliza un mapa topológico del estilo anteriormente mostrado en la figura 1. Este plan es realmente un plan de alto nivel de abstracción ya que no especifica los movimientos del robot con detalle sino que consiste en una secuencia de lugares por los que hay que pasar, que se convierten en sub-objetivos del plan, y se deja en manos de los comportamientos que poseen el resto de agentes la responsabilidad de la consecución de dicha secuencia de sub-objetivos.

En este trabajo, nos queremos centrar en la habilidad del robot para salir de una habitación y poder llegar a otra, por lo que se ha simplificado al máximo la representación del entorno, de manera que sólo se han representado las habitaciones, puertas y pasillos en el mapa topológico. Así, los nodos representan las habitaciones o pasillos mientras que los arcos representan las puertas. Cada una de las puertas puede tener un peso asignado indicando la dificultad de cruzar esa puerta y además posee un número identificativo único, que será el número de la marca situada junto al marco de esa puerta. Las marcas pueden estar situadas en cualquier lado del marco de la puerta y por ello, es necesario conocer su posición para que el robot se pueda orientar correctamente dentro del mapa topológico.

El camino calculado vendrá determinado por las habitaciones y puertas que se han de recorrer para ir desde una habitación origen a otra destino. La visión nos permitirá la búsqueda de las puertas que se han de cruzar para cumplir dicha tarea. Una vez conocido el camino, este agente será el encargado de la ejecución y monitorización del plan. A continuación se describen brevemente las tareas asignadas a este agente.

1. Cálculo del camino. El primer paso es determinar el camino, es decir la sucesión de habitaciones, pasillos y puertas para ir de una habitación origen a otra destino, utilizando para ello el mapa topológico ya comentado.
2. Búsqueda de puerta. Una vez calculado el camino, el primer paso es buscar la primera puerta a cruzar. El método de búsqueda de la puerta variará si el tipo de nodo en el que nos encontramos es un pasillo o una habitación.



- (a) Búsqueda de la marca en una habitación. Para resolver este problema se ha optado, en primer lugar, por realizar una búsqueda por la habitación con el robot parado. La idea es que el robot irá girando la torreta hasta encontrar la marca deseada. En caso de no encontrarla en parado, el robot tendrá que iniciar un proceso de exploración de la habitación en busca de la marca. Este proceso se realizará manteniendo el robot moviéndose en paralelo a la pared con la cámara mirando hacia la misma, activando el comportamiento *Seguir pared*. En cualquier caso, para poder encontrar la marca es importante que la inclinación de la cámara sea la adecuada para que la marca quede dentro de la imagen. Este ángulo de inclinación de la cámara, que llamaremos  $\alpha$ , depende de la distancia  $D$  de la cámara a la pared en la que suponemos puede estar la marca y de las alturas de la marca  $h_m$  y la cámara  $h_c$ . La figura 9 muestra gráficamente la relación entre estas variables, mientras que la siguiente ecuación indica cómo obtener  $\alpha$  en función del resto de valores:

$$\alpha = \text{atan}\left(\frac{h_m - h_c}{D}\right).$$

Para detectar la distancia  $D$ , se usará el promedio de los valores proporcionados por los tres sonars delanteros del robot y aún siendo un método aproximado y sometido a imprecisión, se ha comprobado experimentalmente que los valores obtenidos de  $\alpha$  son suficientemente válidos como para encontrar la marca.

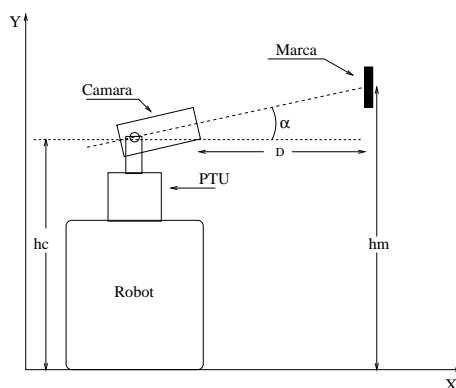


Figura 9: Ángulo apropiado para detectar la marca

- (b) Búsqueda de la marca en un pasillo. Para realizar la búsqueda de la marca cuando el robot tiene que navegar por un pasillo se hace uso de la información del mapa topológico, es decir, el robot puede consultar el mapa topológico para saber en qué sentido tiene que recorrer el pasillo a fin de dirigirse a la habitación destino. Se activa el comportamiento *Mover el robot por un pasillo* para mantener al robot centrado en el pasillo y el comportamiento *Orientar las ruedas y torreta en el pasillo* para que la cámara este dirigida hacia la pared en la que se encuentra la puerta destino con el objeto de facilitar la detección visual de la marca en la puerta de la habitación destino. Una vez detectada la marca, se procederá a orientar el robot de manera adecuada hacia la puerta para proceder a cruzarla mediante el comportamiento *Atravesar una puerta*.
3. Aproximación a la marca en una habitación. Una vez localizada la marca en una habitación, el robot tendrá que dirigirse hacia ella hasta alcanzar una determinada distancia. Para ello, se utiliza el comportamiento *Dirigirse hacia la marca de la puerta*. Esta circunstancia solamente se dará cuando se localice la marca en parado, ya que si se localiza realizando una búsqueda en movimiento, el robot se encontrará ya situado en las cercanías de la puerta. Puede ocurrir que, mientras se realiza el movimiento, la cámara pierda la marca debido a la evitación de un obstáculo. En caso de pérdida, se realizará un barrido por la zona en la que se encontró en última instancia la marca, y en caso de no encontrarla, se comenzaría de nuevo el proceso de búsqueda en la habitación. La búsqueda en habitación se realizará primero en parado y posteriormente en movimiento, tal y como se describió anteriormente.

## 5 Resultados experimentales

Este sistema multi-agente se ha implementado y experimentado utilizando un robot móvil Nomad 200 actualizado con el sistema de visión, situado en un entorno de interiores real representado en la figura 10. A

continuación se comenta un ejemplo en el que el robot debe salir de la *Habitación 1*, recorrer el pasillo y entrar en la *Habitación 3*. En primer lugar el agente Planner calcula el siguiente camino a partir de mapa topológico: Habitación1 → P00 → Pasillo → P02 → Habitación3. A continuación, se ejecuta el plan y se explican las fases por las que ha ido pasando el robot hasta completar el plan.

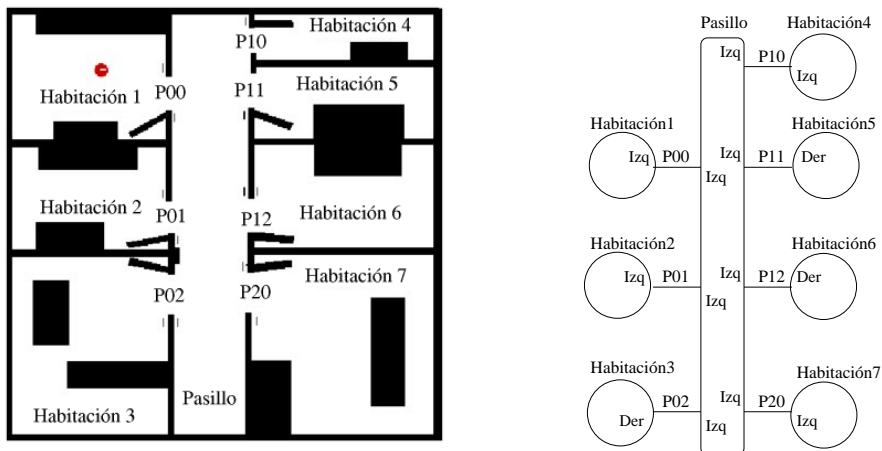


Figura 10: Mapa de habitaciones y mapa topológico del entorno real

1. **Búsqueda de la puerta 00:** Según el camino obtenido anteriormente, lo primero que realiza el robot es buscar la marca que identifica a la puerta 00 mediante una búsqueda en parado. La figura 11 muestra dos imágenes tomadas con el robot del proceso de búsqueda en parado, de manera que en la segunda imagen se puede apreciar la marca con los dígitos 00 a la izquierda del marco de la puerta. En otra situación, puede ocurrir que la marca no se encuentre visible debido a que exista un obstáculo que le impida ver la marca. En ese caso, tras completar la búsqueda en parado sin éxito, se comenzaría la búsqueda en movimiento por la habitación. La figura 12 muestra una traza de la ejecución del robot en esta situación.



Figura 11: Búsqueda en parado de la marca 00.

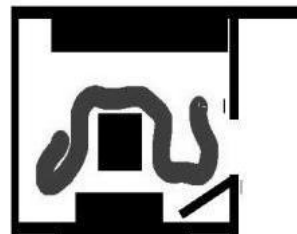


Figura 12: Búsqueda en movimiento de la marca 00.

2. **Aproximación a la marca número 00:** Suponiendo que el robot ha localizado la marca 00 mediante la búsqueda en parado, a continuación debe aproximarse a ella. Para conseguirlo, el robot intenta orientar las ruedas y la torreta con la dirección que le indica la PTU alineada con la cámara, como se muestra en la figura 13, en la cual se observan también varias imágenes tomadas con la cámara a medida que el robot se acerca a la marca. Esta fase termina cuando el robot se encuentra a una determinada distancia de la puerta. Se debe resaltar que esta fase no tiene lugar cuando la marca se localiza mediante el proceso de búsqueda en movimiento, ya que en ese caso el robot se aproximará a la marca en el propio proceso de búsqueda.
3. **Cruzar la puerta número 00:** Cuando el robot se encuentra próximo a la puerta, se orienta hacia el hueco de la misma con la finalidad de centrarse con respecto al marco de la puerta y poder cruzarla. Una vez que el robot se encuentra situado convenientemente, se activa el comportamiento *Atravesar puerta* que se encarga de dicha función. El recorrido del robot y algunas imágenes tomadas mientras el robot cruza la puerta se muestra en la figura 14.

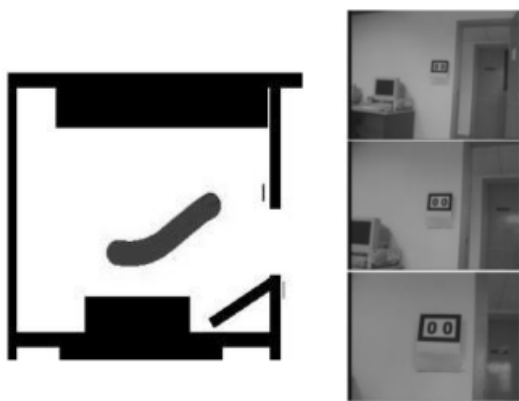


Figura 13: Aproximación a la marca 00.

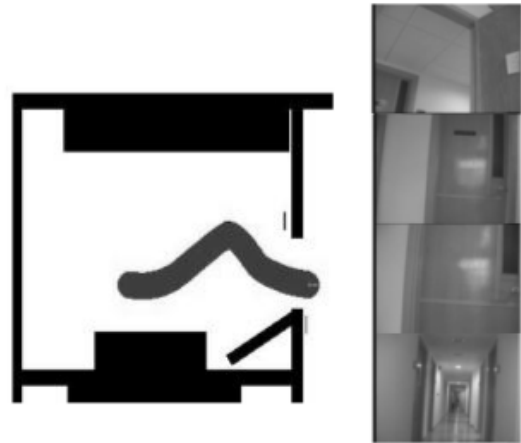


Figura 14: Cruzando la puerta 00.

4. **Recorrer el pasillo buscando la habitación destino:** Al terminar de cruzar la puerta, el robot se encuentra en el pasillo y debe recorrerlo alineado por el centro, hasta encontrar la habitación destino. Para ello el robot debe situarse correctamente para facilitar la búsqueda visual de la puerta destino. Esto es, debe poner las ruedas paralelas a las paredes del pasillo y la torreta perpendicular para que la cámara esté dirigida a la pared donde se encuentra la puerta que se va a buscar. La figura 15 muestra cómo el robot busca y encuentra la marca de la puerta 02 tras recorrer el pasillo.
5. **Cruzar la puerta 02:** Una vez localizada la marca de la puerta 02, el robot puede consultar en el mapa la posición relativa de la puerta respecto a la marca y ayudado de esta información y de los sensores de ultrasonidos, se centra respecto al hueco de la puerta y comienza a cruzarla. Así finalmente se consigue que el robot llegue a la habitación destino. El recorrido que sigue el robot y algunas imágenes tomadas en el proceso se muestran en la figura 16.

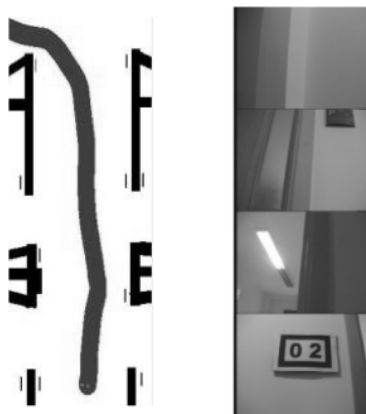


Figura 15: Recorriendo el pasillo en búsqueda de la puerta 02.



Figura 16: Cruzando la puerta 02.

## 6 Conclusiones y Trabajo Futuro

En este trabajo, hemos presentado un sistema multi-agente que emplea tanto información visual como información proveniente de los sensores de rango con la finalidad de incorporar la habilidad de detectar las puertas de las habitaciones y dirigir el robot hacia ellas, ya sea para entrar o salir de ciertas habitaciones de un entorno de interiores. El entorno está representado mediante un mapa topológico formado por nodos, que representan las habitaciones y los pasillos, y arcos que representan las puertas. En cada puerta se sitúa una marca artificial formada por un borde rectangular y dos dígitos interiores. Dicha marca es aprendida por el sistema utilizando una red neuronal que es entrenada a partir de las imágenes, de modo que la marca puede ser identificada

posteriormente y utilizada para que el robot se aproxime a las puertas especificadas en el plan. El agente Tracker es capaz de mantener la cámara enfocada hacia la marca utilizando para ello la PTU, mientras que el agente Navigation cuenta con los comportamientos difusos que le permiten dirigir el robot hacia la marca de la puerta, evitar los obstáculos, seguir las paredes, atravesar las puertas y mover el robot por el pasillo. Dichos comportamientos han sido diseñados utilizando reglas difusas lo que nos permite manejar la imprecisión e incertidumbre subyacente en los datos de los sensores de manera adecuada. Finalmente, el agente Planner es el que genera el plan y da las órdenes al resto de agentes para ejecutar el plan monitorizando su funcionamiento. Los resultados experimentales llevados a cabo en un entorno de interiores demuestran que el robot es capaz de identificar y llevar a cabo el seguimiento de la marca indicada en tiempo real a pesar del propio movimiento del robot; es capaz de aproximarse a ella, realizando evitación de obstáculos mediante los ultrasonidos y nuevas búsquedas de la marca en aquellos momentos en que se pierde del campo visual. En el futuro se pretende que el robot construya el mapa topológico realizando una exploración de la planta de oficinas, describiendo también en el mapa el contenido de las habitaciones mediante otros modelos de percepción desarrollados anteriormente por el equipo.

## Referencias

- Aguirre, E. y González, A. 2000. Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion. *International Journal of Approximate Reasoning*, 25:255–289.
- Aguirre, E. y González, A. 2002. Integrating fuzzy topological maps and fuzzy geometric maps for behavior-based robots. *International Journal of Intelligent Systems*, 17(3):333–368.
- Aguirre, E., González, A. y Pérez, R. 2002. A fuzzy perceptual model for ultrasound sensors applied to intelligent navigation of mobile robots. En *Programs & Proceedings of the Third Int. NAISO Symposium on Engineering of Intelligent Systems*, pág. In CD–Rom, Málaga.
- Arkin, R. 1998. *Behavior-Based Robotics*. The MIT Press.
- Ayache, N. 1991. *Artificial Vision for Mobile Robots*. MIT Press.
- Driankov, D., Hellerdoorn, H. y Reinfrank, M. 1993. *An Introduction to Fuzzy Control*. Kluwer Academic Publishers.
- Elfes, A. 1989. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer Magazine, Special Issue on Autonomous Intelligent Machines*, 22(6):46–57.
- González, R. 1996. *Tratamiento Digital de Imágenes*. Addison-Wesley Iberoamericana, SA.
- Kabuka, M. y Arenas, A. 1987. Position verification of a mobile robot using standard pattern. *IEEE Journal on Robotics and Automation*, 13(6):505–516.
- Kortenkamp, D. y Weymouth, T. 1994. Topological mapping for mobile robots using a combination of sonar and vision sensing. En *Proc. of the Twelfth National Conf. on AI (AAAI-94)*, pág. 979–984, Menlo Park, Calif.
- Kriegman, D., Triendl, E. y Binford, T. 1989. Stereo vision and navigation in buildings for mobile robots. *IEEE Transactions on Robotics and Automation*, 5(6):792–803.
- Kuipers, B. y Byun, Y. 1991. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63.
- Mataric, M. 1992. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304–312.
- Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions On-Systems, Man, and Cybernetics*, SMC-9(1):66–69.
- Thrun, S. 1998. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71.
- Zhang, Z., Weiss, R. y Hanson, A. 1993. Automatic calibration and visual servoing for a robot navigation system. En *Proc. of the IEEE Conf. on Robotics and Automation*, pág. 14–19, Atlanta.