

Arquitectura Multiagentes para un Equipo de Pequeños Robots (2007)

Maritza Bracho de Rodríguez [18], José Ali Moreno [19]
Universidad Centroccidental Lisandro Alvarado - Universidad Central de Venezuela

ABSTRACT:

Using the Distributed Artificial Intelligence and the Distributed Robotics as a frame of reference, particularly the RoboCup World Championship, in this work an architecture for multiple mobile, autonomous, rational and coordinated agents is proposed. This architecture is composed of a mechanism of reasoning or automatic decision making, a functional and control structure, an artificial vision system, a navigation system, and a robotic architecture for a team of small minibots with capacity of acting at the Small Size League. In the design, development and construction of these components are coupled methods and techniques of Emergent Computation with models based on the traditional knowledge representation of Artificial Intelligence, producing algorithms that combine properties as adaptability, robustness, uniformity with representation, inference and universality. The Agents are conformed by mechanisms that integrate sensors out of the body of the robot with effectors embedded in the robot and processes that are executed jointly over different computing devices, providing them with cooperative and competitive behaviors which are exhibited by the minibots while they operate in real time. The execution and performance of the architecture proposed had been experimented and evaluated through the implementation of a Multiagent System that allows the operation of one or more robots on a testbed conformed by a football field made to comply with RoboCup rules.

Key Words - Multiagent System, Distributed Robotics, RoboCup

RESUMEN:

Tomando como marco de referencia a la Inteligencia Artificial Distribuida y la Robótica Distribuida, particularmente a la Copa Mundial de Fútbol de Robots RoboCup, en este trabajo se propone una arquitectura para múltiples agentes móviles, autónomos, racionales, coordinados, la cual comprende un modelo de organización para los agentes, una estructura funcional y de control para cada uno de los agentes, un mecanismo de razonamiento o toma de decisiones automáticas, un sistema de visión artificial, un sistema de navegación y una arquitectura para un equipo de pequeños minibots con capacidad de actuación en la Liga de Pequeños Robots. En el diseño, desarrollo y construcción de estos componentes se acoplan métodos y técnicas clasificadas dentro de Computación Emergente con modelos basados en la representación tradicional de la Inteligencia Artificial, dando lugar a algoritmos que combinan las propiedades de adaptabilidad, robustez y uniformidad con la representación, inferencia y la universalidad. Los Agentes están conformados por mecanismos que integran sensores ubicados fuera del cuerpo del robot con efectores instalados en el robot y procesos que se ejecutan en forma conjunta sobre diferentes dispositivos de cómputo, dotándolos de comportamientos competitivos y cooperativos los cuales son exhibidos por los minibots al operar en tiempo real. La ejecución y rendimiento de la arquitectura propuesta ha sido experimentada y evaluada mediante la implementación de un Sistema Multiagentes que permite la operación de uno o más robots en un ambiente de trabajo constituido por un campo de fútbol construido de acuerdo a las normativas de Robocup.

Palabras Claves - Sistemas Multiagentes, Robótica Distribuida, Robocup.

I. INTRODUCCIÓN

Disenar y construir agentes para desarrollar su actividad en el mundo real, en entornos cambiantes y poco estructurados, sometidos a restricciones como imprecisión, incertidumbre, variabilidad en el tiempo, resalta el papel fundamental que desempeña el cuerpo físico porque debe suministrar una interacción significativa del agente con ese

entorno regido por un conjunto de restricciones. Si en el desarrollo de su actividad, el ente se interrelaciona con otros agentes de su misma especie, se introduce entonces la necesidad de manejar tanto la racionalidad colectiva, como la forma de organización para esta interrelación. Si adicionalmente, el agente interactúa con otros de diferente género, se requiere entonces de mecanismos de comunicación que permitan la coordinación.

Todos estos aspectos han promovido la investigación en un área multidisciplinaria denominada Sistemas Multiagentes, con dos intereses fundamentales: el análisis teórico y experimental de los mecanismos de organización necesarios para que varias entidades autónomas interactúen y la creación de artefactos distribuidos capaces de acometer tareas complejas haciendo uso de la cooperación e interacción entre entidades racionales. Se espera que de estas investigaciones puedan generarse los métodos y técnicas necesarios para la creación de complejos sistemas de computación basados en los conceptos de agentes, comunicación, cooperación y coordinación de acciones, [1].

Partiendo de estos hechos, en este trabajo se desarrolla una arquitectura para agentes con cuerpos físicos, autónomos, móviles, racionales y coordinados, en la cual la funcionalidad global emerge de la interacción paralela de múltiples procesos cooperativos, capaces de generar comportamientos coherentes, utilizando métodos y técnicas clasificados dentro de la Computación Emergente integrados con la perspectiva de la representación del conocimiento.

Desarrollar una arquitectura para múltiples agentes, es un proyecto que abarca un amplio espectro de aspectos a considerar y de decisiones a tomar, tales como: enfoque y estructura de la programación del agente, arquitectura del robot, organización, dinámica, lenguajes, especificaciones, entre otras. Ante esta complejidad, el dominio del problema ha sido definido dentro de la copa de robots jugadores de fútbol, RoboCup, la Copa Mundial de Fútbol para Robots [2]. Esta copa es una iniciativa mundial propuesta para promover la investigación en Inteligencia Artificial y Robótica. Su objetivo primario es la competencia entre equipos de robots jugadores de fútbol, siendo su característica principal la definición de un problema estándar para la evaluación de teorías, algoritmos y arquitecturas robóticas.

RoboCup ofrece un conjunto de ligas clasificadas en dos grandes áreas, Fútbol RoboCup y Rescate RoboCup. En estas ligas los investigadores encuentran una oportunidad única de aprender y compartir soluciones. Fútbol RoboCup, se encuentra organizado en varias competencias. En la Liga de Robots Reales se usan los agentes físicos para jugar fútbol en dos categorías: Pequeños Robots [3] y Robots de Tamaño Mediano. La liga de Pequeños Robots, denominada también F-180, ha sido establecida para el juego de fútbol entre robots de cualquier forma, con un área de ocupación de a lo sumo 180cm^2 definida por un casco convexo bidimensional proyectado sobre un plano horizontal y con un cuerpo que pueda ser contenido en un cilindro de 18cm de diámetro por 15cm de alto. El aspecto más importante de la liga de Robots Reales es que proporciona un grupo de normas y reglas claramente establecidas que definen conjuntos de objetivos a ser alcanzados y niveles de complejidad permitiendo evaluar los adelantos que se logren en la investigación y desarrollo tecnológico, en el diseño, programación e implementación de robots físicos, de agentes de software y de organizaciones de múltiples agentes directamente relacionados con el problema definido.

Fundamentándose en las cualidades de RoboCup, las cuales lo distinguen como un dominio para la investigación en Inteligencia Artificial y Robótica que no solamente permite modelar la complejidad del mundo dinámico y real, sino que además ofrece la oportunidad de evaluar las propuestas de estructuras de agentes y de arquitecturas de sistemas robóticos, en este proyecto se ha seleccionado a la liga de Pequeños Robots, F-180, como dominio para el desarrollo de esta arquitectura de múltiples agentes. Dentro de este marco de referencia, existen varias metas parciales que deben ser alcanzadas para lograr el objetivo principal: 1) Modelado y construcción de un cuerpo físico para el robot, tomando en consideración las limitaciones impuestas por el tamaño que restringen la capacidad de procesamiento a bordo del agente físico. 2) Modelado y desarrollo de un sistema de visión global que permita la eficiencia y robustez en la percepción. 3) Modelado de un sistema de múltiples agentes. 4) Diseño de algoritmos de reducida complejidad que permitan manejar en tiempo real la planificación de rutas y de movimientos para los robots. 5) Diseño e implementación de sistemas de comunicación eficientes para el intercambio de información entre los procesos ejecutados por los agentes fuera y dentro del cuerpo del robot.

II. ARQUITECTURA MULTIAGENTES PARA LOS MINIBOTS

Se inicia la descripción de esta Arquitectura para Múltiples Agentes, con dos aspectos fundamentales: 1) la forma como están organizados los agentes que pertenecen a la colección y las interrelaciones existentes entre ellos; 2) la estructura de cada uno de los elementos individuales de cómputo o agentes.

A. Organización de la Colección de Agentes

En esta arquitectura existen múltiples agentes activos, A_1, \dots, A_n , en tiempo real y en forma simultánea. Estos elementos están organizados en una estructura predefinida que no contiene una jerarquía, sino que es igualitaria. Cada miembro tiene el mismo nivel de autoridad: los agentes participan en forma uniforme e independiente en la toma de

decisiones. No hay control central.

Cada agente puede ejecutar un conjunto de actividades o comportamientos previamente definidos y agrupados bajo el término rol. Los agentes, a su vez, pueden integrar conjuntos de dos o más elementos denominados grupos o equipos. En cada equipo, un elemento tendrá un rol asignado, pero no necesariamente deben estar todos los roles designados a uno de los agentes. Los elementos de la organización son heterogéneos, especializados y pueden ser redundantes, - un rol puede ser desempeñado por varios agentes a la vez -. La asignación de roles es fija y previa a la ejecución, sin embargo dada una situación, - particularmente cuando ocurren excepciones dentro del proceso -, un agente puede asumir, de manera temporal, comportamientos que han sido asignados a otros roles.

Las relaciones entre los agentes están definidas de las siguientes formas: cada agente conoce de la existencia de otros agentes pertenecientes a su organización y están relacionados entre sí de forma operativa. En consecuencia, existe una dependencia asociada a las tareas: El agente A_1 , para completar la ejecución de la acción op_1 puede necesitar, en determinadas circunstancias, de los resultados de una tarea po_2 que debe ser completada por el agente A_2 . Como la relación es dinámica, existe una dependencia comprometida entre A_1 y A_2 : A_2 permite que otros agentes, al momento de realizar sus planificaciones, cuenten con que él cumplirá con sus obligaciones.

No existe comunicación en forma directa: los agentes ni envían, ni reciben mensajes, entre sí. La comunicación es indirecta, específicamente, en la percepción que los agentes tienen unos de otros y en la interacción que se puede presentar producto de la ejecución de acciones. Los agentes tienen la capacidad de distinguir tanto a otros agentes del grupo, como a otros objetos en el ambiente.

La percepción es realizada por un mecanismo computacional único, perteneciente al sistema que capta los cambios en el entorno. Sin embargo, los agentes procesan de manera asíncrona en el tiempo e interpretan de forma diferente.

1) Modelos y Representación del Conocimiento

Existe un modelo del mundo que les permite a los agentes determinar la situación de su espacio de trabajo en cada percepción. Este modelo contiene representaciones en varios niveles de abstracción: 1) en el nivel inferior se encuentra la representación más detallada y específica, la cual es utilizada para producir el movimiento del robot. 2) En el nivel superior, una abstracción más general es usada para resaltar aspectos importantes del mundo y proporcionar la información necesaria en el proceso de toma de decisiones.

Como la estructura de la colección de agentes está previamente definida, existe un modelo de la organización. En él se definen: 1) equipos o grupos existentes en la organización; 2) miembros y roles desempeñados por los diferentes elementos dentro del equipo; 3) las interrelaciones entre los elementos de la organización; 4) el orden que debe existir entre los elementos; 5) las normas de la organización. Este modelo, incluye una base de conocimientos, construida sobre escenarios preprogramados, en la cual se definen posibles acciones o comportamientos a ser ejecutados, para diferentes estados del mundo y de la organización. Estos escenarios están modelados en reglas de condición acción y están representados en árboles de decisión.

Finalmente, existe un modelo de comportamientos para los agentes en el cual se definen el nombre y significado de sus actuaciones. Como éstas son específicas al dominio, en este caso están representados por jugadas a ser ejecutadas por los agentes. Para exhibir sus comportamientos, los robots deben desplazarse físicamente en un espacio real, por lo tanto están dotados de las rutinas clásicas de la robótica móvil: navegar, evadir obstáculos, evadir paredes. Esto implica que cada agente, una vez que ha determinado el comportamiento a exhibir, ejecuta en un ciclo de control, las siguientes primitivas funcionales: planificar el movimiento, traducir movimiento en comandos del robot, controlar ejecución de comandos. Estas primitivas no forman parte de la representación de comportamientos, son dinámicas y están dentro de la parte funcional del agente

2) Descripción Funcional

Los componentes funcionales del Sistema Multiagentes, presentados en la Fig. 1, son descritos a continuación:

Administrador de Percepción: su función es percibir a través de los sensores el entorno y suministrar una representación primaria del ambiente de trabajo al proceso denominado coordinador. Este administrador consta de dos elementos: 1) los sensores y su controlador, los cuales forman el mecanismo usado para percibir el ambiente; 2) el servidor de percepción, proceso que se encarga de convertir los datos sensoriales en medidas cuantitativas y cualitativas, de aspectos importantes y con significado, esto es, en una representación primaria del mundo.

El servidor de percepción y el controlador del sensor, coordinan el trabajo de uno respecto al otro, bajo un mecanismo cliente servidor. El sensor y el controlador, están percibiendo continuamente el ambiente de trabajo, proporcionándole al servidor los datos de la percepción, cuando éste los solicita.

El administrador de percepción y el coordinador funcionan bajo un mecanismo cliente servidor. Cuando el coordinador solicita una percepción, el administrador le proporciona la última representación obtenida del mundo.

or: **proceso tiene como único**

ularmente en esta organización existe solamente un dispositivo de percepción, el coordinador solicita la percepción al administrador correspondiente y la distribuye a todo el equipo de agentes, asegurando de esta forma que en un instante de tiempo t , todos los elementos de la organización están procesando la misma percepción. Además, si cada agente, solicitara directamente la percepción al administrador, éste estaría en la obligación de atender solicitudes simultáneas, repartiendo su tiempo de ejecución y sus recursos en dos funciones básicas: percibir el mundo y atender a tantos clientes como agentes existan en la organización.

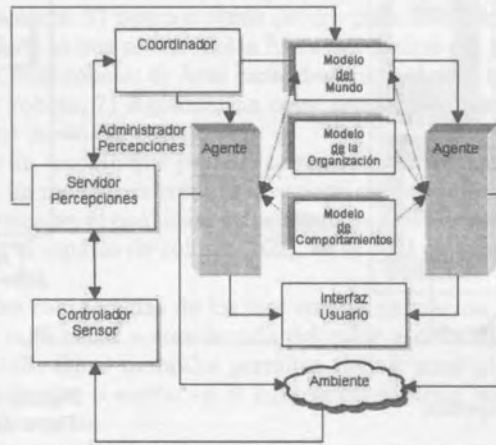


Figura 1: Organización Funcional de los Múltiples Agentes.

De la forma como esta diseñado, el administrador de percepciones solo atiende a un cliente, el coordinador, empleando la mayor parte de su tiempo de ejecución y de sus recursos en procesar la percepción.

Agentes: cada agente es un proceso independiente. Tiene una arquitectura propia, que usando como entrada la representación primaria del ambiente, en una percepción, produce como salida el movimiento significativo del robot, ejecutando una acción.

Interfaz Gráfica para el Usuario: este proceso se encarga de recibir de cada agente, la información concerniente al estado del mundo y las acciones que está ejecutando, para mostrarla al usuario a través de una interfaz gráfica. Funciona bajo un mecanismo de cliente servidor: cuando un agente invoca el servicio, éste se lo proporciona. Además, le permite al usuario solicitar la detención de la ejecución de los procesos activos en el sistema.

B. Estructura Organizativa y de Control para Cada Agente Individual.

Cada agente consta de una arquitectura híbrida de dos capas, Fig. 2: en la capa superior se encuentra el agente remoto o fuera de borda, - ejecutándose en el PC -, y, en la inferior el agente a bordo, - encarnado en el cuerpo del robot -. Entre ambas capas existe un proceso denominado Comunicador cuya función es administrar las comunicaciones que se producen entre el agente fuera de borda y el agente a bordo.

El agente fuera de borda, tiene una arquitectura deliberativa, organizada en módulos horizontales, con control vertical: partiendo de la percepción, los datos fluyen en forma ascendente, aumentando su nivel de abstracción, hasta alcanzar el módulo de toma de decisiones, el cual decide como actuar en función de la información recibida y en concordancia con los objetivos planteados. Luego en la fase descendente, el control fluye hasta alcanzar los módulos de menor nivel: el planificador, generador y controlador de la ejecución de la acción.

Desde el punto de vista funcional, la arquitectura del agente esta compuesto de los elementos ilustrados en la Fig. 3, los cuales son descritos a continuación.

Modelo del Mundo, de la Organización y de Comportamientos: los modelos previamente definidos son suministrados a todos los agentes del sistema.

Módulo para definir la situación del mundo y de la organización: una percepción está compuesta por el valor cuantitativo y cualitativo de las variables más importantes, usadas para representar el ambiente y su estado. Este módulo le permite al agente entender lo que está sucediendo en el mundo: partiendo de la percepción y haciendo uso del modelo del ambiente y de la organización, el agente identifica su situación, la de otros elementos ubicados en el entorno y el estado del mundo.

Módulo para definir el conjunto de opciones: habiendo identificado la situación del mundo y la de la organización, este módulo le permite al agente determinar cuáles serían las posibles acciones a emprender para lograr ciertos estado ideales o cuáles son los estados que se deben evitar, es decir, aquellos que no se deberían alcanzar.

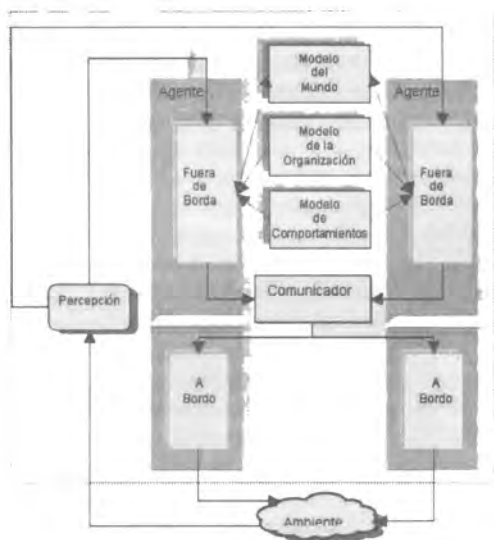


Figura 2: Arquitectura de los Agentes.

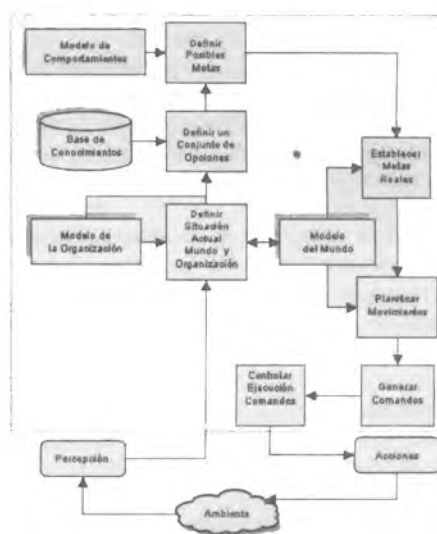


Figura 3: Arquitectura Deliberativa del Agente Fuera de Borda.

Este proceso, soportado por el modelo de la organización, parte de la situación del mundo y de la organización y establece una relación con algunas de las situaciones predefinidas en la base de conocimientos para determinar los posibles comportamientos a ser ejecutados.

Módulo para definir posibles metas: los posibles comportamientos a ser ejecutados son traducidos por este módulo en términos de metas o configuraciones del espacio de trabajo, a ser alcanzadas por el agente.

Módulo para definir metas reales: hasta este momento el agente ha determinado una situación conceptual. Este módulo debe traducir las metas conceptuales en metas reales y concretas, validando que efectivamente existan en el mundo del agente y que puedan ser utilizadas por el módulo planificador en el desempeño de sus funciones. **Planificador de movimientos:** definida la meta concreta, el planificador establece cuál es la ruta que debe trazar para alcanzar esta meta y cuales son los movimientos a realizar para seguir la ruta.

Generador de comandos: identificados los movimientos, este módulo los traduce en función de comandos a ser ejecutados por el robot: mover hacia delante l pasos, mover hacia atrás l pasos, girar m grados a la derecha, girar m grados a la izquierda, parar.

Controlador de ejecución de comandos: para lograr el control sobre los movimientos del robot, - por ejemplo, ubicarlo en la dirección y sentido calculado por el agente -, este módulo envía el comando y asigna un período de tiempo, compuesto por instantes o ciclos, para que el robot ejecute el movimiento correspondiente. En cada instante o ciclo, el módulo verifica si el último comando ha sido completado o aún se está desarrollando. Un nuevo comando solamente es enviado al robot, cuando se completa el período o la ejecución del movimiento.

La arquitectura reactiva del agente a bordo está organizada en módulos verticales con control horizontal: los módulos son ejecutados en forma concurrente y cuando ocurre un conflicto entre ellos, el módulo de mayor jerarquía suprime la ejecución del de menor jerarquía. Cada agente empotrado es una máquina de estímulo respuesta que ejecuta un comportamiento en el robot, dado un comando de entrada.

C. Sistema de Visión Artificial

Aunque la visión es un sensor muy poderoso, es difícil de implementar debido a los altos requerimientos de procesamiento: es necesario revisar un largo flujo de datos de entrada para generar información sensorial de alto nivel en el sistema. El problema puede ser resumido, por lo tanto, en proyectar un flujo de video de entrada a una representación sensorial más prominente, que pueda ser utilizada por otras partes del agente.

El método de aceptación general para el procesamiento visual en tiempo real, y particularmente en el dominio de RoboCup, es el de la segmentación por color. Este método resulta útil en múltiples aplicaciones y su eficacia simplifica notablemente los procesos en las tareas de reconocimiento automático de formas visuales.

El sistema de visión debe contestar a la pregunta: ¿qué hay en el campo de trabajo

cámara es colocada sobre el campo de juego para percibir el estado global del sistema y en las imágenes capturadas por ésta, se identifican los objetos presentes en la escena. Cada robot tiene una marca que identifica el equipo al cual pertenece. De **acherado** a las reglas de RoboCup esta marca puede ser de color Amarillo o Azul. Aparte de esta identificación, otras marcas pueden ser colocadas sobre el robot. En este caso, se coloca una segunda señal para identificar el robot y para determinar su orientación.

Cada imagen capturada por la cámara de video, es procesada por un algoritmo de segmentación, **propuesto** por Bracho, Castro y Moreno en [4] y [5], el cual partiendo de las propiedades cromáticas de los objetos en la escena, los segmenta considerando los siguientes colores: 0) negro o verde oscuro para identificar el campo de juego y el cuerpo de los robots; 1) Gris para identificar todo lo que se encuentra fuera del campo del juego; 2) Amarillo para identificar uno de los equipos; 3) Verde para identificar robots; 4) Azul para identificar el otro equipo; 5) Naranja para identificar la pelota; 6) Magenta, para identificar robots; 7) Aguamarina color usado para identificar otros objetos o robots; 8) Blanco para identificar las paredes y las líneas del campo.

Para reconocer un color es necesario un modelo que permita la representación a través de una forma normalizada y aceptada genéricamente. Básicamente, un modelo de color es la especificación de un sistema de coordenadas tridimensionales y de un subespacio de este sistema en el cual cada color queda representado por un único punto. El algoritmo aplicado en este Sistema de Visión, usa el espacio de colores RGB, en el cual el color es definido por sus componentes espectrales primarias de rojo, verde y azul.

Como cada color esta definido por las coordenadas de los tres colores primarios, se aplica un algoritmo basado en umbrales para su identificación. Para cada canal o coordenada del color a definir, un umbral máximo y un umbral mínimo son colocados en el espacio RGB. Estos umbrales permiten definir unos planos, de tal forma que el espacio contenido entre los planos forma un subcubo o región en el cual se encuentran las combinaciones posibles para las coordenadas RGB referidas al color en cuestión.

La imagen es tomada por una cámara de video y es capturada por una tarjeta multimedia de video, gráficos y TV. El Sistema de Visión se encarga de capturar las imágenes desde el dispositivo de video, efectuando inicialmente tres tareas principales: 1) apertura e inicialización del dispositivo de video; 2) lectura de la información en el dispositivo de video; 3) transformación del formato de la imagen desde el espacio YCbCr hasta el espacio RGB.

Una vez que la imagen ha sido obtenida en formato RGB 32bits se procede a identificar los objetos en la escena, aplicando el algoritmo de segmentación basado en las propiedades cromáticas del los mismo. Este algoritmo requiere ser desarrollado con especial atención a la optimización de la velocidad, eficiencia y exactitud. Cada imagen de datos contiene 320bytes x 240bytes x 4bytes, por lo tanto, si un sistema debe procesar al menos 10 imágenes por segundo, requiere de un procesamiento de 3 Mbytes por segundo. El algoritmo requiere de seis umbrales para definir el subcubo contenedor del color y necesitaría de dos operaciones relacionales y una operación lógica, por cada coordenada RGB, para poder establecer la pertenencia de un punto a cualquier subcubo del color. Para mejor la eficiencia del algoritmo, las operaciones relacionales y la operación lógica por cada coordenada RGB son reemplazadas por cómputos efectuados a nivel de bits, - específicamente tres operaciones lógicas -, haciendo uso de las facilidades de representación interna para los números ofrecidas por los lenguajes de programación, de la versatilidad de las bases de los sistemas de representación y del acarreo, o préstamo, que se produce en una operación de sustracción, cuando el sustraendo es mayor al minuendo.

A medida que se va identificando el color de cada uno de los puntos o píxeles de la imagen, se genera un autómata celular, [6], [7], [8], con espacio celular,

$$C = \{(x, y) | x \in \{0, \dots, x_{max}\}, y \in \{0, \dots, y_{max}\}\} \quad (1)$$

Los autómatas celulares son sistemas dinámicos discretos, descentralizados, compuestos por un gran número de componentes idénticos, caracterizados por una conectividad local, organizados geoméricamente según la conveniencia del problema. Su evolución en el tiempo ocurre de forma sincrónica y paralela, y en ella cada celda cambia de acuerdo a reglas definidas sobre una relación local de vecindad. Un autómata celular contiene los siguientes componentes: 1) Un espacio celular formado por una rejilla o reticulado de n máquinas de estado finito idénticas, cada una de ellas con el mismo patrón de conectividad local y con condiciones de borde. El conjunto de estados de las máquinas de estado finito está representado por \sum y $|\sum|$ denota el número de estados por celda. Cada celda está denotada por un par de índices (x, y) y su estado en el tiempo t está representada por $s_{(x,y)}^t$. La vecindad η_k^t de una celda (x, y) está definida por el conjunto de celdas k y sus correspondientes estados que están conectadas o interactúan con la celda (x, y) . 2) Una regla de transición $\phi(\eta_k^t)$ que establece las operaciones a ser aplicadas para la actualización $s_{(x,y)}^{t+1}$ de cada celda del autómata. Cada celda actualiza su estado de manera sincrónica y en paralelo con las otras. Existen diferentes tipos de vecindad y el tamaño $|\eta|$ está definido por $|\eta| = 2r + 1$, donde r es el radio.

Cada celda del espacio tiene un estado que representa el color del píxel en la imagen. El primer autómata celular **generado** tiene los siguientes estados iniciales: (0) espacio libre en el campo de fútbol, color negro o verde oscuro; (1) **espacio ocupado por objetos fuera del ambiente de trabajo**, color gris; (2) **color amarillo**; (3) **color verde**; (4) **color azul**; (5) **pelota, color naranja**, (6) **color magenta**; (7) **color aguamarina**, (8) **color blanco o color no identificado**. Para

identificar los estados espurios, - por ejemplo, un píxel azul rodeado de píxeles verdes -, el autómata celular original es evolucionado durante 1 periodo de tiempo, al aplicar sobre una vecindad de von Neumann, [9], [10], la siguiente regla de transición,

$$s_{(i,j)}^{t+1} = \begin{cases} s_{(i,j)}^t & \text{si } s_{(i,j)}^t \in \{0, 1, 8\} \\ s_{(i,j)}^t & \text{si } \forall (x, y) \in \eta_k^t, s_{(x,y)}^t = s_{(i,j)}^t \\ 8 & \text{si } \exists (x, y) \in \eta_k^t | s_{(x,y)}^t \neq s_{(i,j)}^t \end{cases} \quad (2)$$

Para eliminar el ruido ocasionado por las líneas blancas o por aquellos autómatas celulares cuyos estados no han sido identificados, la siguiente regla de transición es aplicada el segundo autómata celular, sobre una vecindad de Moore, de radio 1, , [11], [12]:

Sea $G = \{1, 2, 3, 4, 5, 6, 7\} \subseteq \Sigma$

$$s_{(i,j)}^{t+1} = \begin{cases} s_{(i,j)}^t & \text{si } s_{(i,j)}^t \in G \\ g & \text{si } s_{(x,y)}^t \in \{8\}; \text{ donde} \\ & g = s_{(i-r,j-r)}^t \vee \dots \vee s_{(i,j-r)}^t \vee \dots \vee s_{(i+r,j-r)}^t \vee \dots \vee s_{(i-r,j)}^t \vee \dots \vee s_{(i+r,j)}^t \\ & \vee \dots \vee s_{(i-r,j+r)}^t \vee \dots \vee s_{(i,j+r)}^t \vee \dots \vee s_{(i+r,j+r)}^t \quad \forall (x, y) \in \eta_k^t (x, y) \neq (i, j) \\ 0 & \text{en otro caso} \end{cases} \quad (3)$$

Finalmente, usando como entrada el autómata celular obtenido en la etapa anterior, se puede aumentar el tamaño de los objetos identificados con un color en particular, aplicando la siguiente regla de transición sobre una vecindad de Moore, de radio 1, durante m periodos de tiempo:

Sea z el estado que representa el color de los objetos a aumentar de tamaño

$$s_{(i,j)}^{t+1} = \begin{cases} s_{(i,j)}^t & \text{si } s_{(i,j)}^t \in \{1, 2, 3, 4, 5, 6, 7\} \\ z & \text{si } s_{(i,j)}^t \in \{0\} \wedge \exists (x, y) \in \eta_k^t \wedge s_{(x,y)}^t \neq (i, j) | s_{(i,j)}^t = z \end{cases} \quad (4)$$

Como resultado de este mecanismo de visión, se obtiene el ambiente de trabajo representado en un autómata celular bidimensional, de tamaño 320 x 240, en el cual todos los puntos de la imagen han sido clasificados con un color en particular.

D. Sistema de Navegación.

Cuando se construyen robots móviles, uno de los principales problemas a resolver es cómo dotar al robot de la capacidad de planificar sus propios movimientos, de tal forma que pueda decidir automáticamente las actividades a ejecutar, para lograr una tarea especificada por arreglos espaciales de objetos físicos.

Para resolver este problema, se realiza en primer lugar una abstracción, asumiendo que el robot es el único objeto en movimiento en el ambiente de trabajo, se ignoran las propiedades dinámicas del robot y aspectos relacionados con las interacciones mecánicas entre objetos físicos en contacto, transformando el problema físico de la planificación de movimientos en un problema geométrico de planeación de trayectorias. Además, se asume que el robot es un objeto rígido cuyos puntos están fijos unos respecto a los otros y cuyos movimientos están restringidos tan solo por los obstáculos, u otros objetos en el espacio de trabajo.

Una vez que la planeación de trayectorias es efectuada obteniendo posibles rutas de movimiento, se consideran las otras restricciones físicas, temporales y de interacción de los comandos de movimiento con la percepción, a fin de dotar al robot con un sistema de planificación y ejecución de movimientos.

En el algoritmo del Planificador de Rutas, propuesto por Behring, Bracho, Castro y Moreno en [4], [5], se parten de las siguientes consideraciones: 1) El espacio de trabajo, un espacio en , es discretizado en una rejilla fina y regular formando el espacio celular de un autómata celular. Asumiendo un tamaño dado para la rejilla, el espacio de configuraciones puede ser definido como,

$$C = \{(x, y) | x \in \{0, \dots, x_{max}\}, y \in \{0, \dots, y_{max}\}\}$$

2) Cada configuración corresponde a una celda (x, y) . 3) El robot, un objeto libre en movimiento, sin restricciones cinemáticas y dinámicas, está representado por un punto que ocupa una celda (x, y) . 4) Cada objeto presente en la escena, corresponde con la definición del conjunto,

$$C_{region\ obstaculo}, \bigcup_{i=1}^n C(O_i), \text{ donde}$$

$$C(O_i) = \{(x, y) \in C \mid A((x, y)) \cap O_i = \emptyset\} \quad (5)$$

5) El espacio libre o de navegación es el conjunto

$$C_{libre} = C \setminus \bigcup_{i=1}^n C(O_i) \quad (6)$$

6) Una ruta libre de colisiones es una secuencia de configuraciones continuas $q_{inicial}, q_1, \dots, q_{meta}$, tal que

$$A(q) \cap O_i = \emptyset \quad i = 1, \dots, n \quad (7)$$

El espacio de configuraciones discretizado, se obtiene en el Agente fuera de borda y está representado en un autómata celular, donde cada celda del espacio tiene un estado que pertenece al conjunto $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, con el significado siguiente: (0) espacio libre dentro del campo de fútbol, (1) espacios del entorno ubicados fuera del campo de fútbol, (2) color amarillo para robot del equipo o arquería B, (3) color verde claro en cualquier robot, (4) color azul para robot del equipo o arquería A, (5) color magenta en cualquier robot, (6) color aguamarina en cualquier objeto o robot, (7) color naranja de la pelota, (8) color amarillo para robot reconocido, (9) color azul para robot reconocido.

Para encontrar la ruta, se establece la posición del robot propio como *configuración inicial* y está definida por la celda ocupada por el centroide de su marca de identificación. Se calcula la orientación del robot, la cual está determinada por el ángulo que forma la recta que une el centroide de la marca del color identificador del robot y el centroide de la marca del color del equipo, con el eje de las abscisas del sistema de coordenadas cartesianas F_W . Cualquier *configuración meta* dentro del espacio, está definida por una posición, o celda (x, y) , a ser alcanzada en el movimiento. Adicionalmente se considera un ángulo de orientación Θ , el cual representa la dirección que debe tener el robot al aproximarse a la meta.

Tomando como entrada el autómata celular, [4], [5], el algoritmo identifica la configuración meta y aplica una regla de transición sobre una vecindad de Moore, de radio 1, con el objetivo de encontrar la distancia Manhattan, que existe entre la meta y cualquier configuración. La dinámica es aplicada en forma consecutiva, partiendo desde la meta, hasta encontrar la configuración inicial, o, hasta que haya recorrido todo el espacio celular:

El conjunto de estados resultantes posibles para el autómata celular, obtenido al aplicar la dinámica es el siguiente: (0) espacio libre dentro del campo de fútbol, (1) espacios del entorno ubicados fuera del campo de fútbol, (2) color amarillo, para robot el equipo o arquería B, (3) color verde claro en cualquier robot, (4) color azul, para robot del equipo o arquería A, (5) color magenta en cualquier robot, (6) color aguamarina para obstáculos, (7) color naranja de la pelota, (8) color amarillo, para robot reconocido, (9) color azul, para robot reconocido, (10) configuración meta, (10 + 1) distancia Manhattan 1 a la meta, (10 + 2) distancia Manhattan 2 a la meta, ..., (10 + h) distancia Maniatan h a la meta. (Ver Fig. 4).

Este algoritmo es completo y tiene una complejidad lineal al número de celdas en la rejilla n , $O(n)$. Partiendo del autómata celular obtenido se construye la lista de configuraciones que forman la ruta de movimientos, siguiendo el descenso del gradiente, desde la configuración inicial hasta la meta, Fig. 4. Como para algunas celdas existen vecinos que se encuentran a la misma distancia de la meta, se aplica una de dos heurísticas en la selección de la celda vecina, dependiendo del tipo de jugada que debe ejecutar el robot. 1) Cuando el robot se dirige a la pelota, se trata de conservar la dirección del movimiento, por lo tanto aquella celda vecina que se encuentra en esta misma dirección tendrá prioridad en la selección. 2) Cuando el robot se dirige a otras metas, o cuando debe cumplir movimientos paralelos al eje de coordenadas y de F_W , se asigna mayor prioridad a las celdas que se encuentran en las direcciones Norte, Sur, Este y Oeste.

Una vez que la ruta ha sido construida, el generador de comandos la transforma en una lista de funciones del tipo: mover hacia delante l pasos, mover hacia atrás l pasos, girar m grados a la derecha, girar m grados a la izquierda, parar. Para la ruta en celdas con puntos mostrada en la Fig. 4, la orientación del robot es 270° y la lista de comandos es la siguiente: girar 45°, mover hacia adelante 5 celdas, girar 45°, mover hacia adelante 1, girar 45°, mover hacia adelante 9, girar - 45°, mover hacia adelante 5.

El *Controlador de ejecución de comandos* envía la función al robot y asigna un período de tiempo mínimo, compuesto por n instantes o ciclos, para que el robot ejecute el movimiento correspondiente. En cada instante o ciclo,

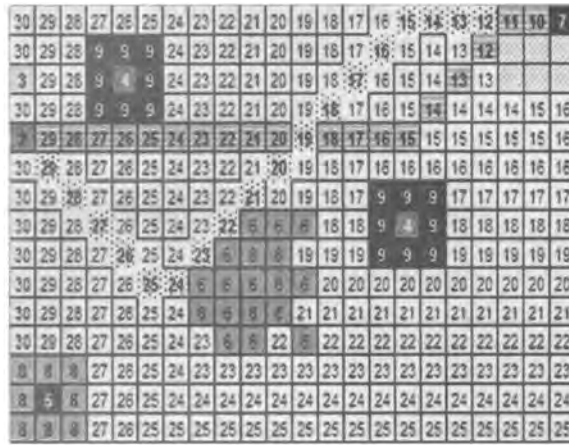


Figura 4: Autómata Celular con Rutas de Movimiento. Al aplicar el algoritmo, la configuración meta (7) se encuentra a una distancia Manhattan 20 de la configuración inicial (2). Ruta en celdas con puntos corresponde a heurística mantener dirección del movimiento. Ruta en celdas con líneas horizontales corresponde a heurística Norte, Sur, Este, Oeste.

el módulo verifica que el comando haya sido completado o que se esté desarrollando. Una nueva función solamente es enviada al robot, cuando se completa el período o la ejecución del movimiento

E. Diseño de los Minibots.

Dentro de la robótica, el término *robótica distribuida* se refiere a sistemas de múltiples robots móviles comprometidos en un comportamiento colectivo [13], [14], [15]. Se caracteriza porque los robots pueden llevar a cabo tareas que un solo robot no puede lograr, debido a que en última instancia un solo robot, - no importa lo capaz que sea -, es espacialmente limitado. La robótica distribuida comprende el estudio, construcción y aplicación de sistemas de múltiples robots, sistemas en los cuales, varios robots interactúan persiguiendo algún conjunto de metas o ejecutando algún conjunto de tareas.

Los robots desarrollados para la aplicación de los algoritmos de este proyecto, son del tipo denominado **minibots**: pequeños robots dotados con microcontroladores, de capacidad de procesamiento y memoria limitados. Los minibots son móviles, normalmente carecen de una estructura en forma de brazo y generalmente su imagen se parece más a la de un vehículo con ruedas. Su tamaño los hace menos costosos. Como están gobernados por un microcontrolador, su capacidad de programación es limitada. No están diseñados para llevar a cabo operaciones que exijan elevadas potencialidades y complejos algoritmos, deben colaborar con otras máquinas para ejecutar tareas globales. Están destinados a realizar tareas pequeñas con rapidez y precisión. Las aplicaciones de los minibots no están pensadas para sustituir a los robots industriales, más bien se enfocan a la investigación, educación, entretenimiento e implementación en tareas sencillas, en lugares donde el espacio para trabajar resulta pequeño.

El **Minibot UCV**, presentado por Bracho, Castro y Moreno en [16], robot cuyas medidas se expresan en centímetros, es una plataforma para la investigación en robótica distribuida. El gran tamaño de muchas de las plataformas disponibles y su alto costo hacen que la experimentación en robótica distribuida, a gran escala, sea poco factible. Este robot, de bajo costo, suministra la funcionalidad requerida por experimentaciones en serie. Es lo suficientemente pequeño como para permitir que experimentos que requieren grupos de robots sean ejecutados de una manera estándar. Este tipo de plataforma robótica es flexible y robusta, permitiendo probar resultados teóricos y suministrando bases para exploraciones futuras.

El robot, cuyo diseño y prototipo fue presentado y construido por Castro en [17], está compuesto de dos plataformas simples de aluminio, en la cuales se soportan sus componentes esenciales, unidas por medio de cuatro pernos y unos tubos de aluminio que hacen de separadores. Sus dimensiones son las siguientes: 12cm de ancho, 12cm de largo y 11cm de altura, - sin incluir altura de la antena-, Fig. 5. Utiliza tecnología electrónica disponible en el mercado y solamente sus componentes mecánicos han sido construidos expresamente para el robot. En la plataforma básica se encuentran dos ruedas, cada una de ellas impulsada de forma independiente por un servomotor, cuya velocidad es controlada por modulación de ancho de pulso, En la segunda plataforma, se encuentra un micro controlador PIC16C73A de la empresa MicroChip. También están ubicados en esta plataforma: 1 batería de 9 voltios que alimenta el micro controlador, 4 baterías de 1.2 voltios para alimentar a los motores del robot, y además, dos módulos de comunicaciones fabricados por Glolab Inc., uno para la recepción - RM1V - y otro para la transmisión - TM1V -.

Estos módulos pueden transmitir y recibir datos a una tasa de hasta 4800 baudios, operan a una frecuencia de portadora de 418Mhz y tienen un alcance de hasta 90 metros, de acuerdo a las especificaciones del fabricante.

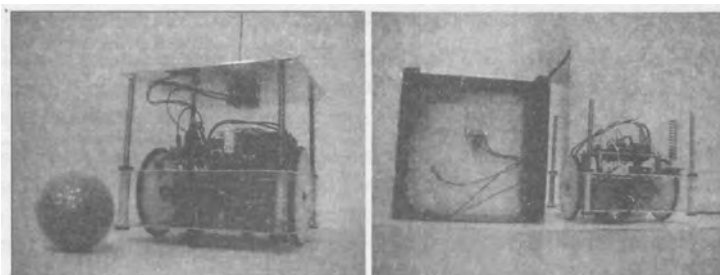


Figura 5: Dos fotografías del minibot UCV.

El robot está gobernado por un programa que controla las interrupciones que ocurren en el micro controlador. Las comunicaciones entre el computador y el robot funcionan bajo el modelo Maestro Esclavo: sólo el computador inicia la comunicación y envía un mensaje. Cada vez que el robot recibe un mensaje, identifica si es para él y si esto es así, lo procesa. La estructura de la trama del mensaje se compone de los siguientes elementos: 1) Preámbulo: Toda transmisión se inicia con dos bytes iguales a cero, de esta manera el robot puede detectar el inicio de un nuevo mensaje y sincronizarse para recibirlo. 2) Dirección: Cada robot tiene una única dirección, en consecuencia, sólo responde al mensaje el robot indicado por este campo. 3) Función: Comando a ser ejecutado por el robot: avanzar, retroceder, girar, parar. 4) Secuencia: contador incrementado con cada mensaje nuevo recibido lo que hace a cada mensaje único. Este campo le permite, al robot, determinar si se trata de un mensaje nuevo o si es un mensaje anterior, retransmitido por que hubo algún error de comunicaciones. 5 y 6) Datos: proporciona la duración del movimiento, medido en cantidad de pulsos o cuentas para el codificador de engranaje. 7) CRC: El valor de este byte es determinado a partir de los campos anteriores y se calcula haciendo el OR exclusivo de todos los bytes que componen el mensaje. Con él se asegura la ausencia de errores de comunicación debido a alteraciones en la información durante la transmisión del mensaje.

F. Implementación.

La arquitectura del Sistema Multiagentes descrito es compleja porque integra sensores ubicados fuera del cuerpo del robot con efectores instalados en el robot, y contiene, además, procesos que se ejecutan en tiempo real y en forma conjunta sobre diferentes dispositivos.

Para lograr la interacción de múltiples agentes cooperativos y competitivos, el Sistema Multiagentes funciona sobre una estructura de procesos sincrónicos y asíncronos ejecutados de forma concurrente o distribuida en diferentes procesadores, que se comunican entre sí a través del pase de mensajes. Con el objetivo fundamental de que máquinas heterogéneas interconectadas en una red de área local, con protocolos TCP/IP, proporcionen un entorno de cómputo en el cual las acciones puedan ocurrir en tiempo real y en paralelo, la arquitectura propuesta para este equipo de pequeños robots, ha sido implementada en un multiprocesador de área local, LAM, Fig. 6. Los multiprocesadores de área local permiten resolver problemas en redes de estaciones de trabajo heterogéneas creando la ilusión de una sola máquina paralela en la cual el sistema es percibido como un sistema unificado, sin que los detalles de la red de área local y de las máquinas individuales que conforman la red sean visibles directamente al usuario.

La configuración del hardware para este proyecto consta de los siguientes elementos: 1) Un procesador con una tarjeta multimedia capturadora de video y una cámara de video instalada sobre el campo de trabajo, a una altura de aproximadamente 2,5m, conformando el Administrador de Visión, Fig 7. 2) Un o más procesadores en el cual se encontrará el multiprocesador de área local instalado, el Coordinador del Sistema Multiagentes los Agentes fuera de borda y la interfaz para el módulo transmisor de comandos vía radio a los robots. 3) Uno, dos, más minibots, por equipo, según el caso particular a implementar. 4) Un campo de fútbol construido de acuerdo a las normas y reglamentaciones de RoboCup, Liga de Pequeños Robots.

III. CONCLUSIONES

Este trabajo se desarrolló en el contexto de la Inteligencia Artificial Distribuida y de la Robótica Distribuida, particularmente en el dominio de la Copa Mundial de Fútbol de Robots, RoboCup. Su aporte fundamental es la proposición de una Arquitectura de Múltiples Agentes para el control de un grupo de Minibots. Esta arquitectura es

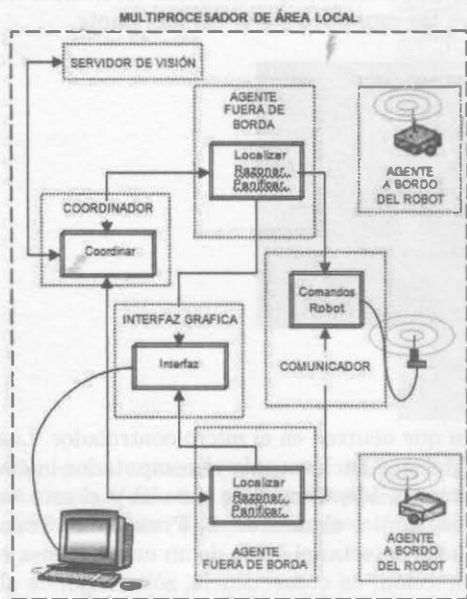


Figura 6: Colección de Procesos Distribuidos sobre los Diferentes Dispositivos de Cómputo

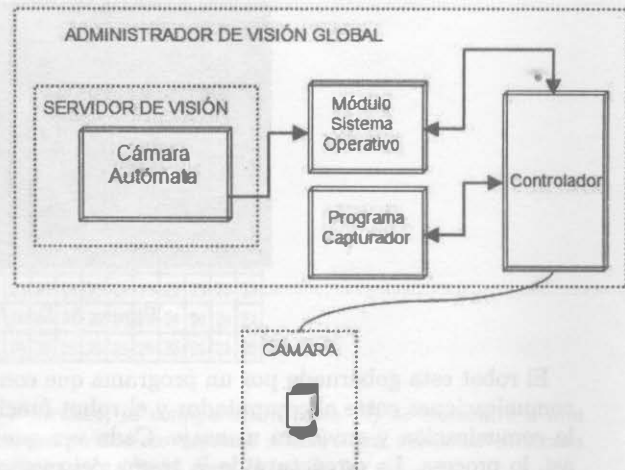


Figura 7: . Administrador de Visión.

implementada y evaluada en un Sistema Multiagentes mediante el cual un equipo de pequeños robots puede exhibir dos tipos de comportamientos: uno cooperativo previamente definido y especificado, el otro competitivo emergente. La arquitectura propuesta comprende:

- Un modelo de organización propuesto para los agentes con las siguientes propiedades: 1) La estructura es igualitaria, cada agente participa en forma uniforme e independiente en la toma de decisiones. No hay control central. 2) Las relaciones entre los miembros es la siguiente: cada elemento conoce de la existencia de otros elementos pertenecientes a su organización y existe una dependencia comprometida entre ellos. 3) Cada miembro de la organización puede ejecutar un conjunto de actividades agrupadas bajo el término *rol* e integrar conglomerados de dos o más elementos denominados *grupos* o *equipos*. 4) Los agentes de la organización son heterogéneos, especializados y pueden ser redundantes. 5) No existe comunicación en forma directa: los agentes ni envían, ni reciben mensajes, entre sí. La comunicación es a través del ambiente, específicamente, en la percepción que los agentes tienen unos de otros y en la interacción que se puede presentar producto de la ejecución de acciones. 6) El comportamiento cooperativo se logra a través de un mecanismo de planeación distribuida, en el cual cada agente considera un conjunto de acciones a ejecutar para lograr el objetivo común obteniendo un conjunto de posibles metas a ser alcanzadas.
- Una arquitectura híbrida de dos capas para cada uno de los Agentes: 1) En la capa superior, se encuentra el agente fuera del cuerpo del robot, cuyo razonamiento es deliberativo. 2) En la capa inferior, se encuentra el agente empotrado en el minibot, con razonamiento reactivo. 3) Entre ambas capas se encuentra un proceso comunicador encargado de administrar las comunicaciones entre ambos agentes.
- Un sistema de visión artificial para el cual se propone y evalúa un nuevo algoritmo de clasificación de colores basado en umbrales que utiliza a un autómata celular en la representación del espacio de trabajo de los robots al cual le son aplicadas diferentes dinámicas con el fin de lograr la segmentación e identificación de los diferentes objetos presentes en el ambiente. Otros resultados obtenidos en el desarrollo del trabajo incluyen algoritmos de localización y seguimiento de objetos en escenas de trabajo para robots móviles, basados en sus propiedades cromáticas.
- Un Sistema de Navegación basado en un nuevo Planificador de Rutas de Movimiento, de complejidad $O(n)$, que calcula la distancia Manhattan entre dos puntos, sobre el espacio celular de un autómata contenedor del ambiente de trabajo del robot, con el fin de planear y obtener rutas de movimientos. Otros resultados obtenidos

en el área de robótica móvil incluyen algoritmos para generación de comandos a partir de rutas y de control de ejecución de movimientos para robots en tiempo real.

- Un minibot construido de acuerdo a las restricciones de tamaño impuestas por la Liga Pequeña de Robots.

La arquitectura propuesta funciona bajo el paradigma del cómputo distribuido: integra sensores ubicados fuera del cuerpo del robot con efectores instalados en el minibot, y contiene además, procesos que se ejecutan en tiempo real y en forma conjunta sobre diferentes dispositivos de cómputo.

Todos los componentes de la Arquitectura propuesta se desarrollaron, integraron, implementaron y evaluaron en un Sistema Multiagentes que permite la operación en tiempo real de uno, dos o más minibots con una capacidad de procesamiento 16 a 24 imágenes cada segundo, mientras que la capacidad de procesamiento del Sistema de Visión es de 59 imágenes cada segundo, para imágenes de dimensión 320bytes x 240bytes x 4bytes, en formato RGB de 32bits.

REFERENCIAS

-
- [1] Ferber, J. (1999). *MultiAgent Systems. An Introduction to Distributed Artificial Intelligence*. Harlow, England: Addison Wesley.
 - [2] Kitano, H., Asada M., Kuniyoshi, Y., Noda, I., Osawa, E., Matsubara, H. (1997). RoboCup a Challenge Problem for AI and Robotics. En K. Hiroaki (Eds). *Lectures Notes in Computer Science, Lectures Notes in Artificial Intelligence: Vol. 1395. RoboCup-97: Robot Soccer World Cup I*. (pp. 1-19). Berlin: Springer Verlag.
 - [3] RoboCup Small Size League. Disponible en: <http://small-size.informatik.uni-bremen.de/> , <http://www-2.cs.cmu.edu/%7Ebrettb/robocup/>
 - [4] Behring, C., Bracho, M., Castro, M., Moreno, J. A. (2000). An Algorithm for Robot Path Planning with Cellular Automata. En S. Bandini, T. Worsch (Eds.). *Theoretical and Practical Issues on Cellular Automata, Proceedings of the Fourth International Conference on Cellular Automata for Research and Industry, Karlsruhe, 4-6 October 2000*, (pp. 11-19). Berlin: Springer Verlag.
 - [5] Bracho, M., Castro, M., Moreno, J. A. (2000). Robot Motion Planning with Cellular Automata. Proceedings of The 4th World Multiconference on Systemics, Cybernetics and Informatics, The 6th International Conference on Information Systems, Analysis and Synthesis. USA, 9, 90-95.
 - [6] Mitchell. M. (1996). Computation in Cellular Automata: A Selected Review. En T. Gramss, S. Bornholdt, M. Gross, M. Mitchell, T. Pellizzari (Eds). *Nonstandard Computation*. Weinheim: VCH Verlagsgesellschaft.
 - [7] Toffoli, T., Margolus, M. (1987). *Cellular Automata Machine*. Cambridge, MA: MIT Press.
 - [8] Wolfram, S. (1984). Computation Theory of Cellular Automata. *Communications in Mathematical Physics*, 96, 15-57.
 - [9] Von Neumann, J. (1966). *Theory of Self-reproducing Automata*. Champaign: University of Illinois Press
 - [10] Weisstein, E. von Neumann Neighborhood. Disponible en MathWorld—A Wolfram Web Resource, <http://mathworld.wolfram.com/vonNeumannNeighborhood.html>
 - [11] Moore, E. (1964). Mathematics in the Biological Sciences. *Scientific American*, 211, 148-164.
 - [12] Weisstein, E. Moore Neighborhood. Disponible en MathWorld—A Wolfram Web Resource <http://mathworld.wolfram.com/MooreNeighborhood.html>
 - [13] Cao, Y., Fukunaga, A., Kahng, A. (1997). Cooperative Mobile Robotics: Antecedents and Directions. *Autonomous Robots*, 4, 1-23.
 - [14] Dudek, G., Jenkin, M., Milios, E. (2002). A Taxonomy of MultiRobot Systems. En T. Balch, L. Parker (Eds.), *Robot Teams: From Diversity to Polymorphism*. Natick, MA: A K Peters.
 - [15] Parker, L., (2000). Current State of the Art in Distributed Robot Systems. En L. Parker, G. Bekey, J. Barhen (Eds.). *Distributed Autonomous Robotic Systems 4*. (pp. 3-12). Heilderberg: Springer Verlag.
 - [16] Bracho, M., Castro, M., Moreno, J. A. (2001). A Robotic Architecture for RoboCup. *Actas de IX Conferencia de la Asociación Española para la Inteligencia Artificial , IV Jornadas de Transferencia Tecnológica de Inteligencia Artificial*, España, 1, 675-684.
 - [17] Castro, M. (2000). *Control en Tiempo Real de un Mini Robot Físico por Medio de Técnicas de Computación Emergente*. Trabajo de Grado aprobado para obtener el Título Magister Scientiarum en Investigación de Operaciones. Universidad Central de Venezuela. Caracas, Venezuela.
 - [18] Unidad de Inteligencia Artificial, Decanato de Ciencias y Tecnología, Universidad Centroccidental Lisandro Alvarado, Barquisimeto, Venezuela.
 - [19] Laboratorio de Computación Emergente, Facultad de Ciencias, Universidad Central de Venezuela, Caracas, Venezuela.