ALGORITMO PARA LA GENERACIÓN DE LABERINTOS DE CONEXIÓN MÚLTIPLE EN 2D.

Víctor Tomás Tomás Mariano[1], Jorge Hernández Camacho[1]

[1]Lic. en Sistemas Computacionales, Universidad Autónoma del Estado de Hidalgo - Escuela Superior Huejutla. Corredor Industrial S/N, Parque de Poblamiento. Municipio de Huejutla, Hidalgo. CP. 43000. Tel. 771-7172000, ext. 5880,5881, e-mail: victor_tomasm@hotmail.com.mx, jhcjorge@hotmail.com

RESUMEN

En el presente trabajo se hace el análisis de los algoritmos para la construcción de Laberintos de Conexión Múltiple (LCM). Estos algoritmos construyen *partes del laberinto con áreas inaccesibles* ocasionando desperdicio de espacio dentro del cuerpo en el laberinto. Partiendo de la inaccesibilidad, se aprovecha el resultado de los algoritmos de construcción de *Laberintos de Conexión Simple* (LCS) para crear un nuevo algoritmo. El algoritmo propuesto se llama "*LCM-Mascaras*", principalmente consiste en buscar una serie de *"mascaras"* o *"patrones"* en la representación matricial de un laberinto. Con este algoritmo, se generan: *Laberintos de Conexión Múltiple (LCM) y Laberintos de Conexión Múltiple Mixto (LCM Mixto)*. Los resultados obtenidos dependen del número de máscaras aplicadas, y estás modifican de forma inmediata la representación matricial del laberinto, además, el manejo de matrices facilita el proceso de construcción.

Palabras Clave: Construcción, Laberintos, Patrones, Mascaras.

INTRODUCCIÓN

Los laberintos rectangulares son de los más utilizados en diferentes campos como lo es la robótica, educación, medicina y entretenimiento [1]-[6], [7], [8]. Existen varios algoritmos para generarlos [2] [6], [13], en este documento se realiza una descripción de los algoritmos más utilizados para la construcción de Laberintos de Conexión Múltiple (LCM). A continuación se presentan los tipos de laberintos rectangular que existen y sus principales características ya que en base a estos laberintos se presenta la contribución del algoritmo "LCM-Mascaras" propuesto.

ROC&C'2011 – CP-18 PONENCIA RECOMENDADA POR EL COMITÉ DE COMPUTACIÓN DEL IEEE SECCIÓN MÉXICO Y PRESENTADA EN LA REUNIÓN DE OTOÑO, ROC&C'2011, ACAPULCO, GRO., DEL 27 DE NOVIEMBRE AL 3 DE DICIEMBRE DEL 2011.

Tipos de Laberintos

Laberinto de Conexión Simple (LCS)

Son aquellos que tienen puntos muertos o callejones sin salida, por lo tanto, solo tienen una solución entre la entrada y la salida, en el proceso de buscar tal solución, en ocasiones, se recorre gran parte del laberinto, causando frustración en el espectador, ver fig. 1a.

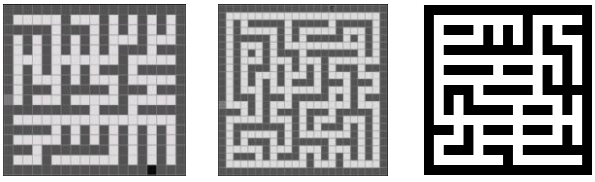
Laberinto de Conexión Múltiple (LCM)

Son aquellos que tienen más de una *solución* desde la entrada a la salida. En este tipo de laberintos, además de encontrar una solución, se requiere que sea la más corta. Tienen ciclos internos, lo que dificulta aún más el proceso de buscar la solución, ya que se debe tomar en cuenta en que secciones del laberinto ya se ha transitado, evitando dar giros en un mismo segmento del laberinto, ver fig. 1b. Sin embargo, en ocasiones, llegan a tener *callejones sin salida*, en ese caso, se les llaman LCM Mixto, ver fig. 1c.

La construcción de un laberinto puede ser tan complicado como el resolverlo, los laberintos a realizar son de tipo cuadriculado o *rejilla*, éste debe ser perfecto y completamente conectado, es decir, se debe poder elegir cualquier punto del laberinto como entrada y cualquier otro punto como salida [2], [6], [11], [12].

Construcción de Laberintos de Conexión Múltiple

La construcción al azar permite generar gran cantidad de laberintos conteniendo corredores como las ramas de un árbol, persiguiendo la desorientación del explorador, siendo a su vez ésta la forma más fácil de implementarse en un programa de computadora.



a) **b)** **c)**

Fig. 1. Tipos de Laberintos a) LCS. b) LCM. c) LCM Mixto.

La construcción al azar se basa en tres enfoques, los cuales se enuncian a continuación:

1. **Cavar túneles**. Como su nombre lo indica, se refiere a cavar túneles a lo largo y ancho del laberinto hasta

cubrirlo en su totalidad, como la explotación de una mina. A este tipo de construcción, pertenecen los algoritmos de construcción para LCS: *Kruscal, Aldous-Broder, Prim's, Recursivo Bactracker* y *Anderson* [1][2][6].

2. **Levantar muros.** Consiste en elevar los muros de los corredores por todo el cuerpo del laberinto, como en la construcción de un edificio.

3. **Levantar o cavar muros.** El algoritmo que se propone, pertenece a este tipo de construcción para la generación de LCM o LCM Mixto, y este consiste en cavar o levantar muros según sea el caso de la máscara aplicada.

Los algoritmos explicados a continuación, generan laberintos LCM completamente conectados [2] [6].

Algoritmo de Construcción de Anderson Modificado

1. Inicie el laberinto extremadamente conectado (laberinto sin paredes, excepto el exterior del laberinto).
2. Elegir dos esquinas adjuntas y levantar un muro entre ellas.
3. Elija en forma aleatoria cualquier esquina de una celda a la que ninguna pared sea adjunto. Si no hay esquinas, finalizar, el laberinto está terminado.
4. Si es posible, construir una pared de la esquina a una esquina cercana, al azar, siempre que la esquina cercana sea parte de una pared –conecta una pared a una ya existente–. Ir al paso 3.
5. Indique la entrada y salida del laberinto.

Algoritmo para Construir Laberinto de Conexión Múltiple (ACLCM)

Está técnica evita la colocación de muros en secciones que formen callejones sin salida. Los pasos a seguir en esta técnica son los siguientes:

1. Especificar el tamaño o las dimensiones del laberinto de acuerdo con las habitaciones que se desea que contenga.
2. Se marcan los vértices de las habitaciones cubriendo todo el cuerpo del laberinto.
3. Marcar el muro exterior o el perímetro del cuerpo del laberinto.
4. Elegir al azar un punto libre y levantar un muro hacia un punto adyacente, siempre y cuando no forme ningún callejón sin salida o áreas inaccesibles, (en caso de existir más de uno, elegir al azar), repitiendo este paso hasta que ya no quede ningún punto libre en el laberinto.
5. Marcar la ubicación de la entrada o del punto inicial de la exploración y la ubicación de la salida.

En los algoritmos para construir LCM previos, es necesario el uso de grafos para la representación de los puntos internos de la matriz, además, en el ACLCM, en el punto 4, no se aclara de manera precisa cuando se forma un callejón sin salida o área inaccesible. En ambos algoritmos, los laberintos resultantes, tienen callejones sin salida y en laberintos de gran tamaño, tienen errores, ya que se construyen partes del laberinto que son inaccesibles ocasionando desperdicio de espacio [2][6].

DESARROLLO

En el algoritmo propuesto para construir un LCM o LCM Mixto, el resultado depende del número de máscaras aplicadas, y la ejecución de alguna de ellas, se refleja de forma inmediata en la representación matricial del laberinto, además, el manejo de matrices facilita el proceso de construcción**.**

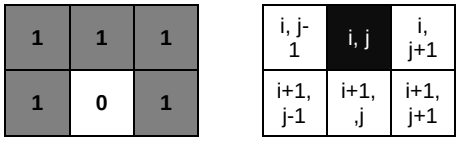
Algoritmo para construir Laberintos de Conexión Múltiple LCM-Mascaras

Este algoritmo genera LCM o LCM Mixtos, dependiendo del número mascaras o patrones se apliquen en el proceso de construcción.

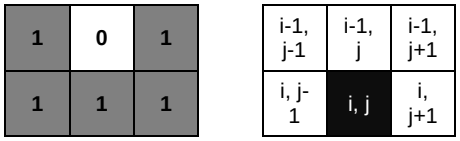
1. Generar un laberinto de conexión simple (LCS) aplicando cualquier algoritmo de construcción [1].
2. Hacer una representación matricial del laberinto anterior en "0" y "1" [1].
3. Buscar el patrón de las máscaras definidas en las figs. 2, 3, 4, y 5 dentro de la matriz que representa al laberinto en el orden en que están presentadas. Se hace con la finalidad de abrir o levantar muros. Repetir este paso, mientras, se puedan seguir aplicando patrones de máscara. *La máscara se aplica siempre y cuando se cumpla la condición de no modificar las paredes externas del laberinto*.
4. Terminar. Designar entrada y salida del laberinto.

Mascaras o patrones de búsqueda

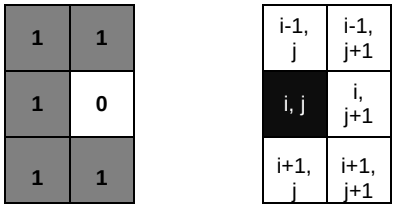
Las figs. 2, 4 y 5, están divididas en dos secciones, las máscaras y la representación de posiciones con respecto a la matriz, estas se utilizan para cavar muros en las celdas matriciales (i, j) representada en color negro.



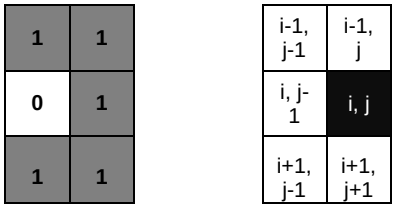
Mascara 1



Mascara 2



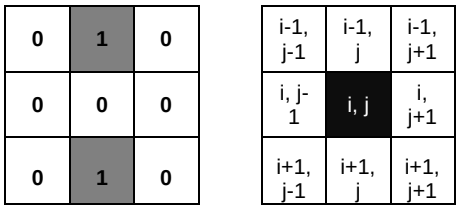
Mascara 3



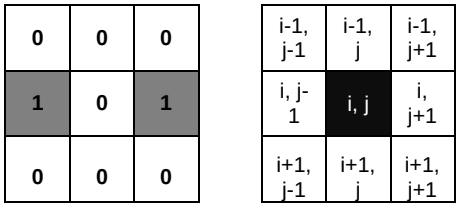
Mascara 4

Fig. 2. Patrones de Mascaras a buscar en el laberinto matricial.

Las máscaras de la fig. 2 permiten eliminar los callejones sin salida, hacer posición (i, j) igual a 0.



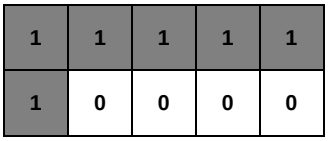
Mascara 5

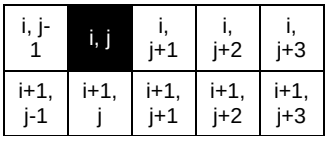


Mascara 6

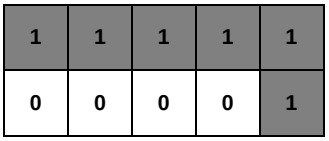
Fig. 3. Patrones de Mascaras a buscar en el laberinto matricial.

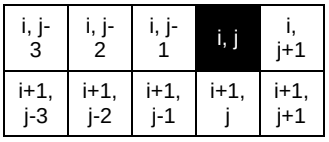
Como se observa en la fig. 3. estos permiten levantar muros, pero estas máscaras pueden generar callejones sin salida, hacer posición (i, j) igual a 1.



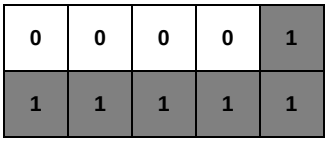


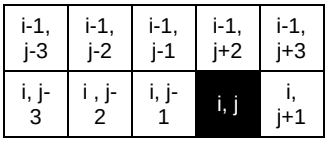
Mascara 7



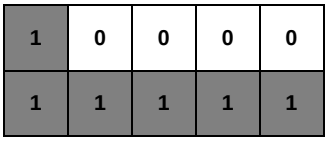


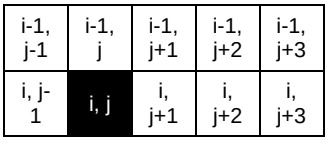
Mascara 8





Mascara 9





Mascara 10

Fig. 4. Patrones de Mascaras a buscar en el laberinto matricial.

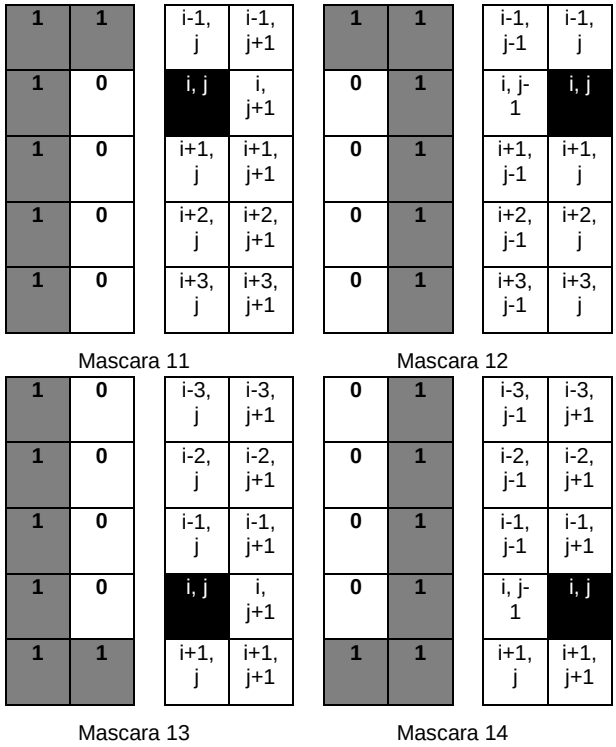


Fig. 5. Patrones de Mascaras a buscar en el laberinto matricial.

Las máscaras de las figs. 4 y 5, permiten eliminar callejones sin salida en secciones cercanas a la pared exterior del laberinto, hacer posición (i, j) igual a 0.

Ejemplo de construcción de un LCM con el algoritmo propuesto.

1. Paso 1: se presenta un caso tomando como ejemplo el laberinto de la fig. 6.

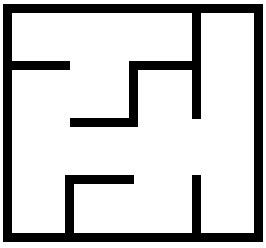


Fig. 6. Representación grafica de LCS.

2. Paso 2: representación matricial:

1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	1
1	1	1	0	1	1	1	0	1
1	0	0	0	1	0	1	0	1
1	0	1	1	1	0	1	0	1
1	0	0	0	0	0	0	0	1
1	0	1	1	1	0	1	0	1
1	0	1	0	0	0	1	0	1
1	1	1	1	1	1	1	1	1

Fig. 7. Representación matricial "0" y "1" de la Figura 6.

Los patrones a buscar, deben estar dentro la representación matricial y hacerse en el orden en que están enumeradas.

3. Paso 3: se encontraron las máscaras 1 y 3. Cavar muro en la posición [i, j]: observe fig. 8.

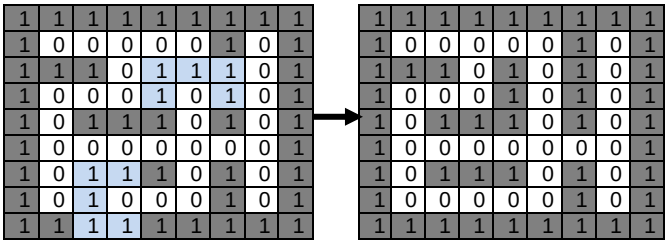


Fig. 8. Localización de máscaras y ejecución.

4. Paso 3: se encontró la máscara 5. Levantar un muro en la posición [i, j]: observe fig. 9.

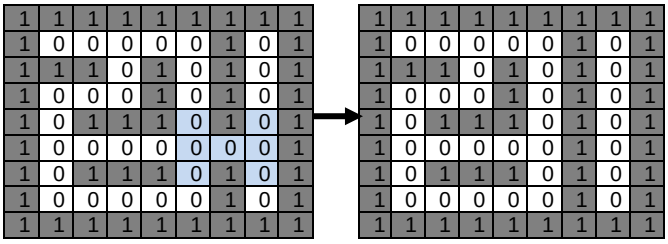


Fig. 9. Localización de máscara y ejecución.

5. Paso 3: se encontró la máscara 11. Cavar un muro en la posición [i, j]: observe fig. 10.

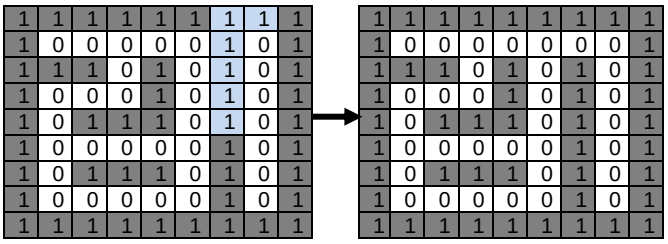


Fig. 10. Localización de máscara y ejecución.

6. Paso 3: se encontró la máscara 13. Cavar un muro en la posición [i, j]: observe fig.11.

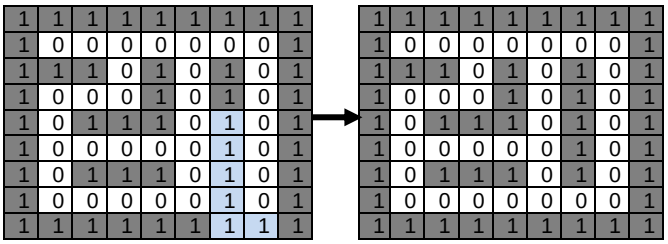


Fig. 11. Localización de máscara y ejecución.

7. Paso 4: finalizada la localización y ejecución de las máscaras, el laberinto queda como se observa en la fig. 12.

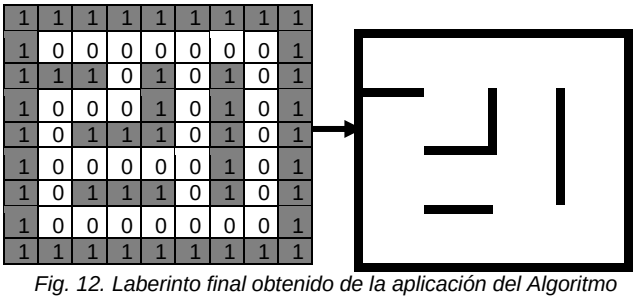


Fig. 12. Laberinto final obtenido de la aplicación del Algoritmo LCM-Mascaras.

Una vez terminado el laberinto, se designa la entrada y su respectiva salida, y posteriormente se aplica el algoritmo de Lee o Moore [11][12] para encontrar la solución del laberinto de tipo rejilla, estos localizan la ruta más corta entre dos celdas conectadas, vea fig. 13.

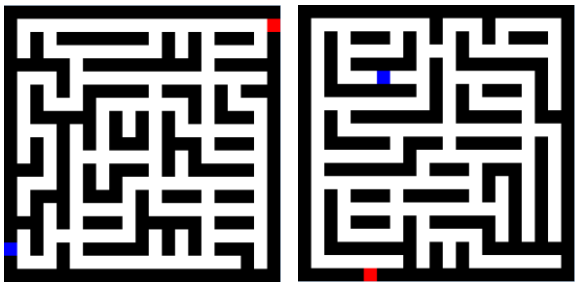
1	1	1	1	1	1	1	1	1
2	3	4	5	6	7	8	9	1
1	1	1	6	1	8	1	10	1
1	9	8	7	1	9	1	11	1
1	10	1	1	1	10	1	12	1
1	11	12	13	12	11	1	13	1
1	12	1	1	1	12	1	14	1
14	13	14	0	14	13	14	0	1
1	1	1	1	1	1	1	1	1

Fig. 13. Laberinto y solución etiquetada.

RESULTADOS

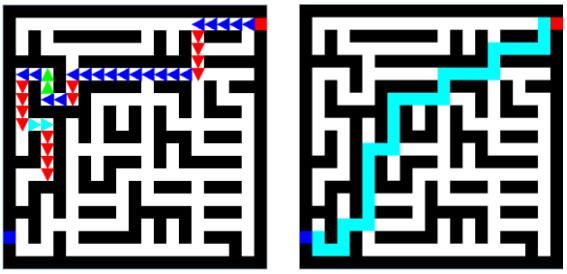
Como se observa en la fig. 14, la salida –punto azul- no necesariamente debe encontrarse en la pared externa del laberinto.

En la fig. 15 a), se observa el recorrido ejecutado por el usuario para encontrar la salida, los movimientos básicos realizados son: izquierda, abajo, arriba y derecha. En la fig. 15 b) se presenta la solución del laberinto, observe que es la más corta.



a) *b)*

Fig. 14. Salida de un laberinto. a) la entrada y salida están en un el extremo del laberinto. b) la salida se encuentra en la parte interna del laberinto.



a) *b)*

Fig. 15. Características de un Laberinto. a) recorrido realizado por el usuario. b) ruta solución de laberinto.

POSIBLES APLICACIONES

Una de las aplicaciones de los laberintos es en la rehabilitación neuropsicológica, los laberintos son pruebas de gran utilidad para la detección de déficit de atención y síndrome frontal. Permiten evaluar diferentes niveles de integración nerviosa y trastornos específicos como secuela de una lesión cerebral [5] - [4].

La robótica es una las áreas en donde el problema de planeación de trayectorias en ambientes o espacios abiertos dinámicos tiene gran aplicación, en ocasiones se cuenta con obstáculos extras en el camino, en estos casos, el robot se auxilia de sensores para su detección y con la implementación de algoritmos que brindan información local, se puede encontrar una solución [7]-[10].

El conocimiento de la estructura del laberinto por donde se mueve el robot es de gran importancia ya que se

debe tomar en cuenta la forma del propio robot para la planeación de trayectorias [7] [9] [10].

CONCLUSIÓNES

En el algoritmo propuesto "LCM-Mascaras" para construir un LCM o LCM Mixto, el resultado depende del número de máscaras aplicadas, y la ejecución de alguna de ellas, se refleja de forma inmediata en la representación matricial del laberinto, además, el manejo de matrices facilita el proceso de construcción.

Los resultados arrojados por el algoritmo propuesto permiten generar LCM o LCM Mixtos de diferentes tamaños y complejidades, totalmente conectados y perfectos.

BIBLIOGRAFÍA

[1] V.T. Tomas Mariano, M. Pozas Cárdenas, J. Hernández Camacho "Propuesta para la generación de laberintos ampliados en 2D", Artículo, Simposio Iberoamericano Multidisciplinario de Ciencias e Ingenierías (SIMCI-2011). ISBN: 978-607-95344 Zempoala, Hidalgo, 2011.

[2] Tomás M.V.T. "Generación y Ampliación de Laberintos", Tesis Maestría, Pachuca de Soto, CITIS, Hidalgo, México, 2007.

[3] Pozas C. M. J., L. H. Roberto, "Generación de Laberintos para Aplicaciones Neuropsicológicas". CITIS, Hidalgo, México, 2006.

[4] Pozas C. M. J., L. Rojas V., Domínguez O., Tomás M. V. T., "Los Juegos Abstractos Como Instrumentos de Evaluación de Habilidades Espaciales", CITIS, Hidalgo, México, 2006.

[5] Domínguez R. O. A., López M. V., Samperio L. R., "Resultados Preliminares sobre Interacción Háptica en Laberintos Virtuales, con Propósitos de Diagnóstico en Pacientes con Discapacidades Neuropsicológicas", UAEH, CITIS, Hidalgo México, 2005.

[6] Bernabe S. E. "Construcción y Recorrido de Laberintos", Tesis Licenciatura, UAEH, CITIS, Hidalgo, México, Julio 2004.

[7] Jan G. E., Chang K-Y., Parberry I. "A New Maze Routing Approach for Path Planning of a Mobile Robot", IEEE International Conference on Advanced Intelligent Mechatronics (AIM 2003)

[8] Dracopoulos. D. C., "Robot Path Planning for Maze Navigation", Brunel University, Department of Computer Science, IEEE, 1998.

[9] Gordon V. S., Matley Z., "Envolving Sparse Direction Maps for Maze Pathfinding", California State University, Sacramento and Digital Eclipse Software, Inc. Vancouver, 1992.

[10] Fraenkel A. S., "Economic Traversal of Labyrinths", in Mathematics Magazine, vol. 43, pp. 125-130, 1970.

[11] Lee C. Y., "An Algorithm for Path Connections and its Applications", IRE Transactions on Electronic Computers, vol. EC-10, pp. 346-365, 1961.

[12] Moore E. F., "The Shortest Path Through a Maze", Annals of Computation Laboratory of Harvard University, Harvard University Press, vol. 30, pp. 285-292,1959.

[13] Pullen D. W, 15/08/2011, http://www.astrolog.org/labyrinth /algorithm.htm.

CURRICULUM

Víctor Tomas Tomás Mariano obtuvo el grado de Maestro en Ciencias Computacionales con especialidad en procesamiento digital de imágenes y señales en la Universidad Autónoma del Estado de Hidalgo, en Pachuca de soto, Hidalgo, México en el 2007. Profesor investigador titular y está incorporado a la Licenciatura en Sistemas Computacionales de la Escuela Superior Huejutla de la Universidad Autónoma del Estado de Hidalgo, en Huejutla de Reyes, Hidalgo, desde el año 2008. Su interés en áreas de investigación incluye: planeación de trayectorias, *maze search problem*, laberintos virtuales 2D y 3D, computación inteligente, Entre las publicaciones más recientes están: *Propuesta para la generación de laberintos ampliados en 2D,* en el 2011, *Los Juegos Abstractos Como Instrumentos de Evaluación de Habilidades Espaciales* en el 2006.

Jorge Hernández Camacho obtuvo el grado de Maestro en Ciencias Computacionales con especialidad en computación educativa en la Universidad Autónoma del Estado de Hidalgo, en Pachuca de soto, Hidalgo, México en el 2008. Profesor investigador titular y está incorporado a la Licenciatura en Sistemas Computacionales de la Escuela Superior Huejutla de la Universidad Autónoma del Estado de Hidalgo, en Huejutla de Reyes, Hidalgo, desde el año 2008. Su interés en áreas de investigación incluye Sistemas SCADA's, diseño de instrumentación virtual para invernaderos automatizados vía internet, sistemas educativos, sistemas de realidad virtual, algoritmos genéticos, sistemas multiagentes, inteligencia artificial y laberintos. Entre las publicaciones más recientes están: *Propuesta para la generación de laberintos ampliados en 2D, A Generalised Semantics for Belief Updates —A Consistency-based Approach* en el 2011, *Virtual distributed supervision and control for an automated greenhouse*, y *Virtual Greenhouse – Virtual tour in a real automated greenhouse* en el 2006.