



Tecnura

ISSN: 0123-921X

tecnura@udistrital.edu.co

Universidad Distrital Francisco José de Caldas
Colombia

GALÁN MUNÉVAR, JONATHAN; ROBLES SÁNCHEZ, EDISSON LEANDRO; MORENO
MOSQUERA, HENRY

La robótica aplicada a la lúdica

Tecnura, vol. 15, núm. 30, julio-diciembre, 2011, pp. 52-63

Universidad Distrital Francisco José de Caldas
Bogotá, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=257020895006>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

La robótica aplicada a la lúdica

The robotics applied to leisure

JONATHAN GALÁN MUNÉVAR

Ingeniero electrónico, investigador de la Universidad Nacional de Colombia. Bogotá, Colombia. jgalanm@unal.edu.co

EDISSON LEANDRO ROBLES SÁNCHEZ

Ingeniero electrónico, investigador de la Universidad Nacional de Colombia. Bogotá, Colombia. elrobless@unal.edu.co

HENRY MORENO MOSQUERA

Ingeniero electrónico, magíster en Ingeniería de Telecomunicaciones. Docente de la Escuela Colombiana de Ingeniería. Bogotá, Colombia. henry.moreno@escolaing.edu.co

Clasificación del artículo: Investigación (Conciencias)

Fecha de recepción: 28 de febrero de 2011

Fecha de aceptación: 30 de mayo de 2011

Palabras clave: Interacción humano-robot, procesamiento de imágenes, programación de robots, visión de máquina.

Key words: Human-robot interaction, image processing, programming robots, machine vision.

RESUMEN

Dentro del contexto social colombiano se tiene la idea de que la electrónica, especialmente aplicada a la robótica, es un tema de gran complejidad y en el que estamos rezagados. Sin embargo, esto no es del todo cierto, por lo cual se pretende mostrar que con algo inicialmente complejo para la gran mayoría como lo es un brazo robótico, se pueden realizar ciertas actividades y tareas programadas previamente, obteniendo un sistema experto.

Para el caso particular de este trabajo, se diseñó e implementó un sistema con fines lúdicos capaz de desempeñar el juego de “Tres en raya” contra un oponente humano. Para tal fin, se tuvieron en cuenta varios aspectos tales como el discernimiento de la disposición de las fichas, tanto las que están dentro del tablero de juego como las que están por fuera, el planteamiento de un algoritmo capaz de establecer la mejor jugada que debe realizar el sistema artificial de forma tal que tenga la capacidad de ganarle al humano y la ejecución de los movimientos de las fichas por par-

te del sistema artificial en el juego. Todas estas consideraciones conllevan a la integración de tres áreas fundamentales: procesamiento de imágenes, algoritmia y robótica.

ABSTRACT

Within the social context Colombia has the idea that electronics, especially applied to robotics, is a very complex issue and we are behind. However, this is not entirely true, which is intended to show that something initially for most complex such as a robotic arm can perform certain activities and/or pre-programmed tasks, resulting in an expert system.

For the particular case of this study was designed and implemented a system able to play for fun game of Tic Tac Toe against a human opponent. For this purpose, took into account various aspects such as the perception of the willingness of both those cards in the game board as they are out, the approach of an algorithm capable of establishing the best play to be performed the artificial system in a way that has the ability to beat the man and the execution of the movements of the chips by the artificial system in the game. All these considerations lead to the integration of three key areas: image processing, algorithms and robotics.

* * *

1. INTRODUCCIÓN

EL trabajo realizado se basa en la integración de tres áreas de investigación de la electrónica: la robótica, el procesamiento digital de imágenes y la inteligencia artificial.

La idea es mostrar un trabajo integral y trasversal en estas 3 ramas del conocimiento, para dar lugar a un sistema experto capaz de participar en una partida de “Tres en raya”. Para tal fin, se diseñaron y desarrollaron tres módulos bajo la plataforma de programación gráfica LabVIEW®, los cuales conforman un sistema conjunto que se comunica con dos dispositivos periféricos para la adquisición de datos (cámara de video) y la ejecución de respuestas (brazo robótico).

El planteamiento del sistema conjunto se puede apreciar en la Fig. 1. En el módulo de adquisición y procesamiento de la imagen; se desea que la imagen sea tomada por la cámara y procesada con el fin de definir el área de juego e identificar la ubicación de las fichas involucradas en la partida, para crear una matriz virtual en la que se

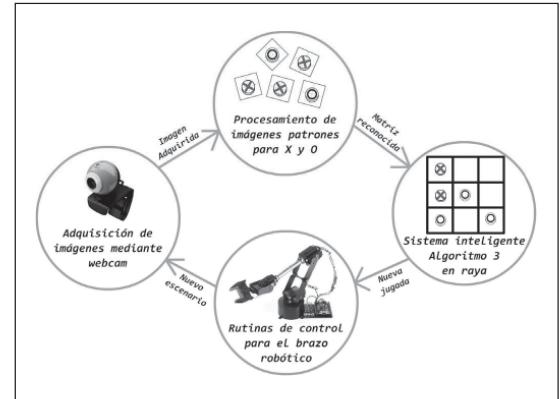


Fig. 1. Estructura general del sistema.

establezcan los tres estados posibles: casilla vacía, casilla con equis (X) y casilla con círculo (O). Luego, esta matriz virtual se envía al módulo de algoritmo para resolución del “Tres en raya”, el cual se encarga de analizarla para plantear la mejor jugada posible a favor del sistema artificial, dependiendo del estado actual del juego. Finalmente, esta respuesta es procesada por el módulo de control de movimiento del brazo robótico, el cual establece la configuración de los respectivos ángulos para cada articulación del brazo robótico,

de manera tal que se puedan desempeñar movimientos de recolección y colocación de la ficha en el lugar determinado por el módulo de algoritmo para resolución del juego.

2. MÓDULOS DEL SISTEMA

2.1 Adquisición y procesamiento de la imagen

La función de este módulo consiste en adquirir la señal del entorno en tiempo real para su posterior análisis, de tal forma que se pueda visualizar la matriz que sirve como escenario de juego y las fichas que se ven involucradas en el desarrollo del mismo. Una vez adquirida la señal, se entra a procesar la información obtenida, logrando de esta manera determinar la ubicación de la matriz del juego, la naturaleza de las fichas (círculo o equis), la posición de las fichas que se encuentran dentro del tablero para definir cuáles casillas están vacías o llenas y la posición de las fichas que se encuentran fuera del tablero para futuras jugadas que se realizarán por medio de la manipulación del brazo robótico.

2.2 Algoritmo para resolución del “Tres en raya”

Una vez procesada la imagen se realiza un procedimiento en el cual se asocia la información proveniente del medio contenida en la matriz real del juego, a una matriz virtual en donde se plasma exactamente lo que está ocurriendo en el ambiente real.

Posterior a este proceso, se emplea un módulo llamado *algoritmo de resolución del “Tres en raya”*, el cual tiene como objetivo usar la información plasmada en la matriz virtual para aplicar un algoritmo que sea capaz de responder a una jugada propuesta o plantear una jugada; es decir, se propone un juego entre usuario y máquina de forma tal que la máquina tenga inteligencia y pueda res-

ponder o establecer una jugada dependiendo del turno de juego. En otras palabras, es el cerebro del proyecto.

2.3 Control de movimiento del brazo robótico

La tarea del módulo de control de movimiento del brazo robótico radica, en usar la información suministrada por el algoritmo de juego que se ve representada en términos de coordenadas de una matriz virtual, para que funcionen como entradas de selección de unas tramas previamente definidas en el módulo que le dan movimientos complejos al brazo robótico a todos los sitios a donde tenga que desplazarse para llevar a cabo el juego del “Tres en raya” satisfactoriamente. Además, dicho módulo incorpora toda la configuración de la comunicación serial RS-232 usada para la transmisión de las tramas creadas en el sistema artificial hacia el brazo robótico, dando así vida al proyecto.

3. METODOLOGÍA

Como primera medida, se buscó un algoritmo eficiente en la solución del juego, que no implicara un gran uso de recursos en máquina. Por otro lado, se efectuó el tratamiento de imágenes basado en el tablero de juego y la posición de las fichas para sincronizar las acciones efectuadas por el brazo. En este caso, se realizaron pruebas para integrar el algoritmo con el tratamiento de imágenes del tablero de juego. Posteriormente, se definieron las ecuaciones de la cinemática inversa, que facilitarían la gestión de los movimientos fijos por programar para el autómata. Teniendo esto, se guardaron rutinas para cada uno de los movimientos que el brazo ejecutaría en el juego. Estas tramas se incorporaron al instrumento virtual principal, de manera que los movimientos ejecutados por el brazo fueran consecuentes con los otros dos módulos del sistema (procesamiento

de imágenes y algoritmo de juego). Para finalizar, teniendo ya estos desarrollos se llevaron a cabo las últimas pruebas y se complementó el sistema con una interfaz gráfica final para facilitar la experiencia del usuario.

3.1 Equipos y materiales

3.1.1 Cámara USB (*Universal Serial Bus*)

La cámara USB utilizada fue la referencia QuickCam Connect E 2500 Series Logitech®, la cual tiene características comunes para estos dispositivos: compatibilidad con USB 2,0, captura de imágenes a 30 muestras por segundo y captura en tiempo real a una resolución de 640 x 480 píxeles [1].

No obstante, para la aplicación en LabVIEW las imágenes fueron capturadas a 15 muestras por segundo y se utilizó una resolución de 320 x 240 píxeles. Dicha configuración fue debida a que no se necesitó gran velocidad de video al capturar imágenes estáticas (captura de los patrones en juego), y en cuanto a resolución, la idea era emplear cualquier dispositivo webcam del mercado. Las pruebas realizadas mostraron que las características propuestas fueron apropiadas para el diseño del proyecto.

3.1.2 Tarjeta de control SSC-32

Esta tarjeta que se adquirió al fabricante Lynxmotion®, es un dispositivo que permite controlar los servos pre ensamblados Fig. 2. Posee una alta resolución para lograr un posicionamiento preciso y permite movimientos muy suaves. Tiene un rango de 0,50 ms a 2,50 ms para lograr un alcance alrededor de 180°. El control de movimiento puede ser de respuesta inmediata, con control de velocidad, tiempo de movimiento, o una combinación de éstos. Además, se pueden realizar combinaciones de los servos para empezar y terminar

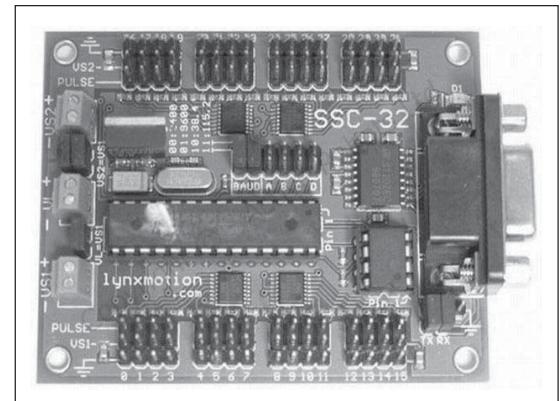


Fig. 2. Tarjeta SSC32.

Fuente: http://www.adrirobot.it/Braccio_robot/braccio_robot_luca/immagini/braccio_robot_mini-ssc-32.jpg

el movimiento al mismo tiempo, incluso si los servos tienen que moverse a diferentes distancias. A continuación se mostrarán algunas de las especificaciones del servo controlador:

- Microcontrolador: Atmel ATMEGA168-20-PU
- EEPROM (ROM programable y borrable eléctricamente): 24LC32P
- Frecuencia de operación: 14,75 MHz
- Entrada serial: True RS-232 o TTL, 2400, 9600, 38.4k, 115,2k, N81 [2], [3]
- Resolución de servo: 1uS, 0,09°
- Resolución de velocidad de servo: 1uS / Segundo

3.1.3 Comunicación del brazo robótico al ordenador

Para lograr la comunicación entre el brazo robótico y el ordenador se empleó el protocolo RS-232. Las tramas a transmitir están definidas por la programación del microcontrolador. Estas tramas tienen un formato llamado “comando de grupo” cuya estructura se describe a continuación:

con-ciencias |

<ch> P <pw> S <spd> ... # <ch> P <pw> S <spd> T <time> <cr>

Donde los parámetros se encuentran definidos en la tabla 1.

Tabla 1. Comandos de trama.

<ch>	Número de canal en decimal, 0-31
<pw>	Ancho de pulso en microsegundos, 500-2500
<spd>	Velocidad de movimiento en μ s por segundo para un canal (Opcional)
<time>	Tiempo en mS para el movimiento completo (afecta todos los canales)
<cr>	Carácter de retorno, ASCII (Código Estadounidense Estándar para el Intercambio de Información) 13 (Necesario para iniciar acción)
<esc>	Cancelar la acción actual, ASCII 27

3.1.4 Brazo robótico Lynxmotion® AL5D

El brazo robótico Lynxmotion®AL5D proporciona un movimiento rápido, preciso y repetible. Cuenta con características propias como la rotación de la base, hombro y codo, movimiento de la muñeca, pinza funcional, y por último, la posibilidad de rotación en la muñeca.

A continuación se muestran las características dimensionales del brazo robótico:

- Número de ejes: 4 + Pinza; Muñeca giratoria (opcional)
- Distancia del hombro al codo: 14,605 cm
- Distancia del codo a la muñeca: 18,7325 cm
- Control de movimiento del servo: Lazo cerrado local
- Altura (brazo en posición inicial): 18,415 cm
- Altura (alcance máximo): 48,26 cm
- Alcance medio: 26,035 cm
- Apertura de la pinza: 3,175 cm
- Rango de movimiento por eje: 180°

- Precisión de movimiento por eje: depende del servo controlador (SSC32 = 0,09°)

4. DISEÑO Y DESARROLLO EN LABVIEW

Cada uno de los respectivos módulos se realizó en LabVIEW 8,5. En adición, fue necesario usar el NI Vision Development Module 8,5, como también, el software NI Vision Assistant 8,5 que fue de gran utilidad a la hora de poder inspeccionar el correcto funcionamiento del diseño propuesto para llevar a cabo el procesamiento de la imagen.

4.1 Adquisición de la imagen

En el proceso de adquisición de la imagen se busca obtener información del medio que luego será llevada a una etapa de procesamiento. En este caso, se usó una webcam para lo cual es necesario usar el software NI-IMAQ for USB Cameras, el cual contiene una librería llamada NI-IMAQ for USB Cameras Virtual Instrument que ofrece la posibilidad de utilizar cámaras USB con LabVIEW.

Para el procedimiento de la adquisición de la imagen se deben seguir ciertos pasos y usar determinados VI's (Instrumentos Virtuales). En primer lugar, se tienen que establecer las cámaras USB disponibles en el equipo con el bloque llamado IMAQ USB enumerate cameras.vi y con un matrix index se selecciona la cámara a utilizar. Posteriormente, se emplean los bloques IMAQ USB init.vi e IMAQ USB grab setup.vi para inicializar la cámara y para permitir la grabación respectivamente. Despues, se utiliza el bloque IMAQ create que asigna una posición de memoria temporal para la imagen. En este bloque se asigna el nombre de la imagen y el tipo, que para este caso sería una imagen a color RGB (U32).

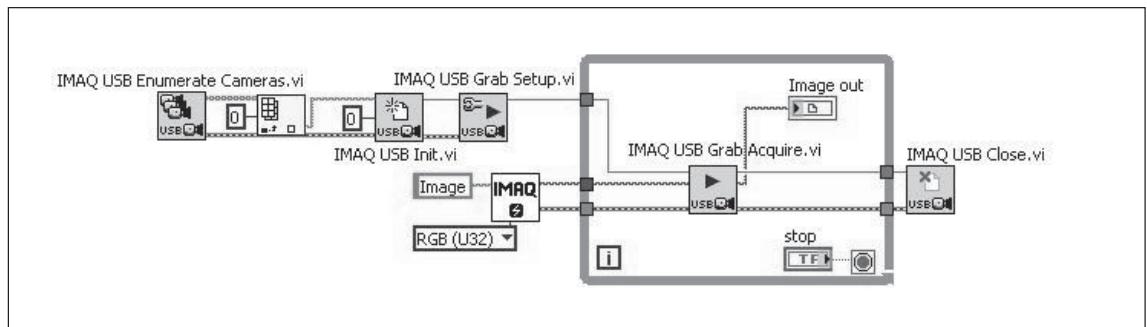


Fig. 3. Etapa de adquisición de la imagen.

Luego, se debe asegurar la adquisición continua de la imagen, colocando el bloque IMAQ USB Grab Acquire.vi dentro de una estructura while para que se mantenga activada la visualización de la imagen, de tal forma que si se desea finalizar dicha adquisición se oprime el botón STOP [4, 5]. Luego, se hace uso del bloque IMAQ USB Close.vi que cierra la sesión de la cámara USB abierta con el IMAQ USB Init.vi, completando de esta manera toda la etapa de adquisición la cual se puede visualizar en la Fig. 3.

4.2 Procesamiento de la imagen

Enseguida de la adquisición de la imagen, se realiza su procesamiento para poder obtener información acerca de ésta. Para tal fin, se procede a obtener el atributo más usado en procesamiento de imágenes considerado así, porque es el que mayor información aporta de una imagen y al cual se le denomina plano de luminancia [6]. Este componente posee la información de luminosidad de la imagen, es decir, es la versión en blanco y negro de la imagen original [4, 5].

Al extraer el mencionado plano de la imagen original, se obtiene como resultado lo que se muestra en la Fig. 4.a.

Para acentuar los patrones de las fichas, se requirió emplear la función morfológica de erosión, que

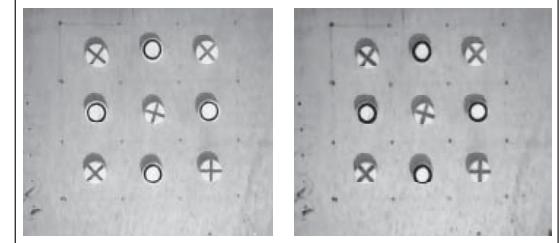


Fig. 4. Métodos usados para el procesamiento de la imagen: a) extracción del plano de luminancia, b) aplicación de la morfología de escala de grises al tablero de juego.

compara cada píxel de la imagen con los píxeles alrededor, para determinar cuáles tienen el valor más pequeño. Esto permite incrementar el grosor aparente de las piezas tal y como se aprecia en la Fig. 4.b [7].

Luego, se utilizó un método llamado coincidencia de patrones para comparar dos imágenes y ver la similitud entre éstas. Con este método se pueden localizar las regiones de una imagen en escala de grises que contienen un patrón de referencia conocido como plantilla, de tal forma que para efectos prácticos del trabajo, se pudiera establecer la naturaleza de las fichas de juego. Para tal fin, se diseñaron dos bloques de detección de patrones, uno para “O” y otro para “X”, donde en la Fig. 5 se muestra el bloque para las fichas marcadas con “X” y en la Fig. 6 se muestra el resultado de aplicar dicho bloque.

con-ciencias |

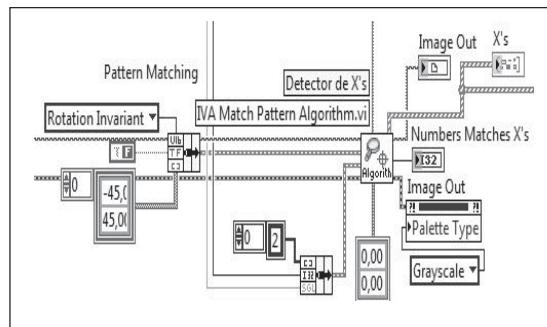


Fig. 5. Bloque de detección para equis.

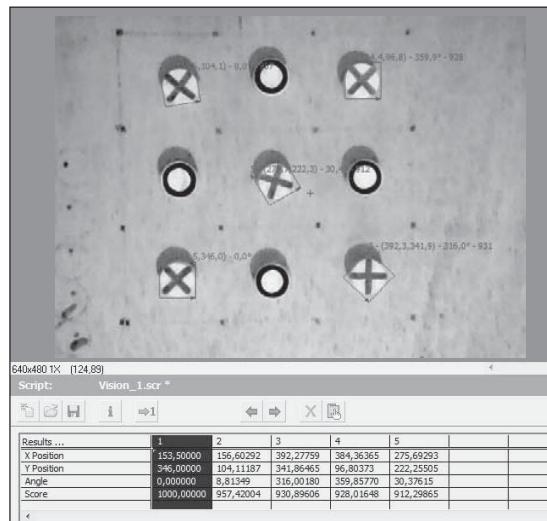


Fig. 6. Resultado de la aplicación del bloque de detección de equis en el tablero de juego.

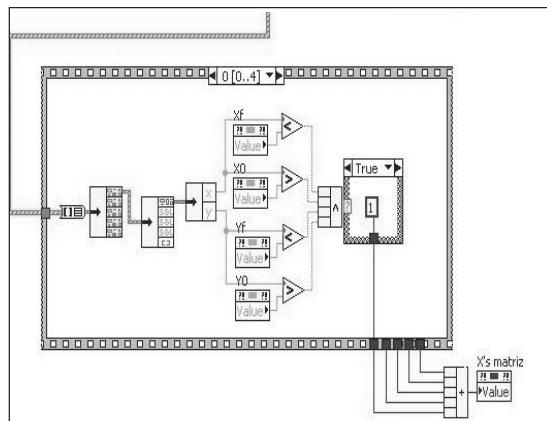


Fig. 7. Bloque de discernimiento de fichas en juego para las piezas marcadas con equis.

Posteriormente, se implementó un bloque para discernir las fichas en juego, usado para contabilizar las fichas marcadas con “X” o “O” que se encuentran dentro del tablero de juego. Este procedimiento se lleva a cabo con el fin de habilitar algunos procesos que están sujetos a una nueva jugada del ser humano. En la Fig. 7 se muestra el bloque de discernimiento de fichas en juego para las piezas marcadas con “X”. Cabe destacar que los dos bloques de discernimiento de fichas son similares, tanto en su estructura como en su funcionamiento.

Por último, se realiza un bloque capaz de asociar el campo físico de juego con la matriz de posiciones, el cual se encarga de definir las dimensiones físicas del tablero y de cada casilla, con base en las posiciones relativas de la esquina superior izquierda y de la esquina inferior derecha del tablero de juego tomadas por la cámara. En la Fig. 8 se puede visualizar el diseño en LabVIEW del bloque descrito.

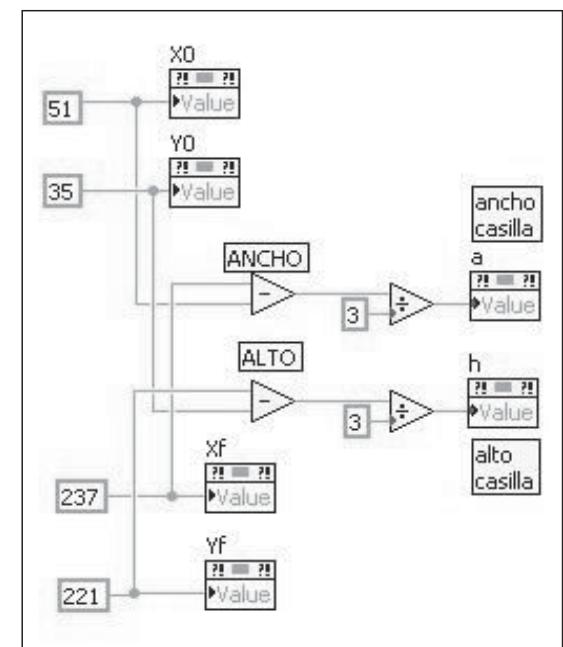


Fig. 8. Bloque de asociación entre el campo físico de juego con la matriz de posiciones.

Tabla 2. Sub VI's usados en cada jugada.

	Juega 1º Humano	Juega 1º Máquina
Jugada 0	Jugada humano	Juega al centro
Jugada 1	Si jugada 0 humano = centro => Juega a esquina. Sino => Juega al centro	Jugada humano
Jugada 2	Jugada humano	Sub VI jugada 2
Jugada 3	Sub VI Jugadas sin borde, sino => Sub VI jugadas con borde	Jugada humano
Jugada 4	Jugada humano	Sub VI bloquea o gana, sino => sub VI PC 1º 4º jugada, sino => sub VI random
Jugada 5	Sub VI bloquea o gana, sino => sub VI random	Jugada humano
Jugada 6	Jugada humano	Sub VI bloquea o gana, sino => sub VI random
Jugada 7	Sub VI bloquea o gana, sino => sub VI random	Jugada humano
Jugada 8	Jugada humano	Sub VI bloquea o gana, sino => sub VI random

4.3 Algoritmo de juego

En el juego de “Tres en raya” hay que tener claras algunas implicaciones que pueden influir en el desarrollo y resultado del juego. Una de ellas, radica en el turno de los jugadores, ya que jugar de primero tiene ciertas ventajas que se pueden aprovechar para ganar [8]. Siguiendo esta idea, el algoritmo de juego se divide en dos partes, de manera que dependiendo de la elección del usuario (sobre quién juega primero), se reduzca el algoritmo a ejecutar a la mitad. Para las jugadas importantes dentro del juego se realizaron sub VI's para cada una, y se clasificaron en dos categorías: *Jugadas especiales* y *Jugadas ordinarias*, dependiendo de su especificidad y del momento en el que se ejecutan. En la tabla 2, se mencionan los sub VI's usados por el sistema en cada jugada.

4.3.1 Jugadas especiales

Las jugadas especiales se refieren a las jugadas clave que le darían la posibilidad de triunfo en el

caso de que el brazo jugara primero, o de empate, cuando jugara de segundo. Éstas generalmente se ejecutan durante los primeros cinco turnos de juego y están definidas con base en las estrategias de juego para el tres en raya descritas en el apartado anterior. Algunas jugadas especiales requirieron poca inteligencia, por lo cual no requirieron bloques de ejecución relevantes, sin embargo, otras fueron bastante más complejas, por lo que se crearon “subVI's” individuales para cada una.

4.3.2 Jugadas ordinarias

Las jugadas ordinarias son las usadas durante todo el juego con el fin de bloquear, en el caso de que el contrincante tenga ya dos fichas en línea, o con la intención de ganar, en el caso de que dos de las fichas propias se encuentren en línea. Para las jugadas finales se tiene otra clase de jugada ordinaria, que en caso de que haya un empate seguro pero quedando casillas desocupadas, asigne una ficha a cualquiera de las casillas vacías (aleatoria-

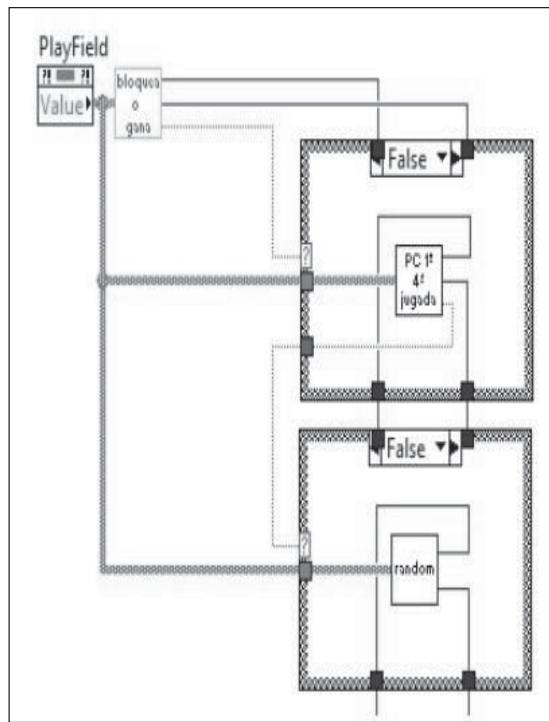


Fig. 9. Ejemplo de sub VI's de jugadas conectados en cascada

mente). Para el caso del proyecto, el ser humano siempre jugará con las fichas “X”.

En cada turno de juego, las posibles jugadas se dispusieron en cascada, dependiendo de la pertinencia de éstas para una disposición de tablero determinada. Un ejemplo se muestra en la Fig. 9, donde se prueba primero si hay posibilidad inmediata de ganar o de bloquear con el sub VI *bloquea o gana* (alta prioridad). Luego, en caso de que la disposición del tablero fuera otra, se ejecuta otro VI, en este caso de jugada especial para comprobarlo, llamado *PC 1° 4° jugada* (prioridad media). Finalmente, si la posición de las fichas no corresponde con ninguno de los dos casos anteriores se ejecuta una función *random* (baja prioridad), que simplemente ubica la ficha en cualquier posición restante en el tablero, cuando ya no se corre riesgo de derrota ni hay posibilidad de triunfo por parte del sistema artificial.

4.4 Control del brazo robótico

Lograr la ejecución de los respectivos movimientos del brazo robótico requiere de una integración con los demás módulos del sistema; no obstante, también fue necesario la implementación de la cinemática inversa [9] propia del brazo Fig. 10, en donde éste se modeló como un manipulador sobre una base rotante que tiene su campo de acción en un espacio 2-dimensional. Todo esto conlleva al desarrollo del módulo de control de movimiento del brazo robótico.

En primer lugar, fue necesario crear un instrumento virtual llamado *control_brazo* que permitiera probar los movimientos de cada servo dentro de los rangos admisibles y la adecuada comunicación de tramas entre el brazo robótico y el ordenador, empleando el protocolo RS-232. Posteriormente, se implementó el instrumento virtual *guardar_ruta* que guarda un listado específico de tramas en un archivo de texto para crear así una rutina de movimiento. Este bloque incorpora la cinemática inversa de forma tal que se puede controlar el giro de un servo o la posición espacial del elemento efector. Luego, se estableció el instrumento virtual *leer_ruta* que tiene como fin leer las rutinas guardadas previamente para conseguir que el brazo robótico realice dichas rutinas. Finalmente, se desea integrar el módulo de control de movimiento del brazo robótico con el algoritmo de juego, lo

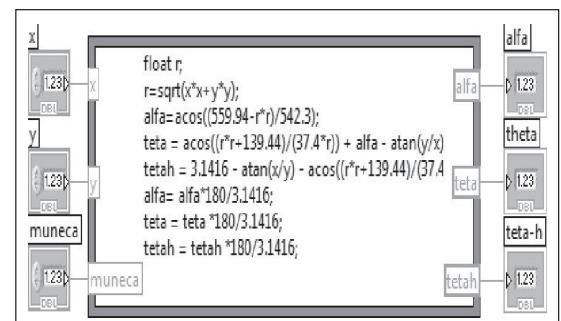


Fig. 10. Fragmento de código para la cinemática inversa.

cual se realiza por medio del instrumento virtual *recoger y poner*, que permite que el brazo recoja una de las cinco fichas ubicadas fuera del campo de juego (en un orden específico) y las coloque de forma correcta dentro del campo de juego, dependiendo de las decisiones tomadas por el algoritmo de juego. Se puede considerar como una forma más compleja del instrumento *leer ruta*, ya que no solo lee una secuencia de movimientos sino que gestiona todas las rutinas de movimiento del brazo implicadas en el juego de tres en raya.

5. RESULTADOS

En esta sección se pretende demostrar si el método de *erosión con elemento estructurante de 3x3 y de una iteración* usado para el procesamiento de la imagen fue pertinente. Para tal fin, se realizaron varias pruebas con distintos métodos de procesamiento de imagen con el software *NI Vision Assistant 8,5®*. Las pruebas consisten en determinar cuál de los métodos ofrece un mayor puntaje para la *media de los aciertos* del patrón, cuál tiene la menor *desviación estándar entre aciertos* y cuál posee la mayor *diferencia entre la media de aciertos y la media de los patrones*

obtenidos erróneamente. Todo esto con el fin de lograr la mayor exactitud respecto al patrón base en la detección de los aciertos y a su vez, evitar capturas erróneas de imágenes similares al patrón, pero que se salen del contexto del juego. El procedimiento que se siguió fue colocar en el campo de juego varias fichas de la figura “X” ya que para estas piezas la coincidencia del patrón se ve afectada por el ángulo de giro, situación que no presentan las fichas “O”. Posteriormente, se capturó la imagen del tablero de juego, se tomó como patrón una de las fichas, y finalmente se aplicaron algunos métodos de procesamiento, tomando valores para cada uno. Los resultados obtenidos de las pruebas se recopilaron en la tabla 3, donde los valores de “Score” se presentan en una escala que va de 0 a 1000, donde 1000 indica una coincidencia perfecta de la imagen obtenida respecto al patrón medido.

De los resultados alcanzados se puede deducir que el método de erosión con una sola iteración (erosión 1) tiene la mayor media de “Score” para aciertos, con un valor de 96,5% de exactitud para la coincidencia respecto al patrón. Además, la desviación estándar tiene un valor muy bajo, es decir,

Tabla 3. Comparación entre valores estadísticos comentados para los métodos probados. Los números que acompañan a los métodos “Erosión” y “Dilatar” representan el número de iteraciones efectuado.

Método de Procesamiento	Media Score de Aciertos	Desviación Estándar	Media Score de Erróneos	Diferencia entre Medias
Sin método	950,94	38,58	648,12	302,83
Erosión 1	965,01	21,66	211,48	753,53
Erosión 2	912,11	63,66	499,09	413,02
Erosión 3	869,17	99,01	546,59	322,58
Dilatar 1	586,37	308,38	565,69	20,69
Dilatar 2	652,35	380,94	672,21	-19,86
Cerrar	923,09	62,68	481,40	441,69
Abrir	763,23	426,79	653,54	109,69
Filtro sobel	715,18	405,44	564,61	150,57
Imágenes binarias	687,64	394,86	432,93	254,71

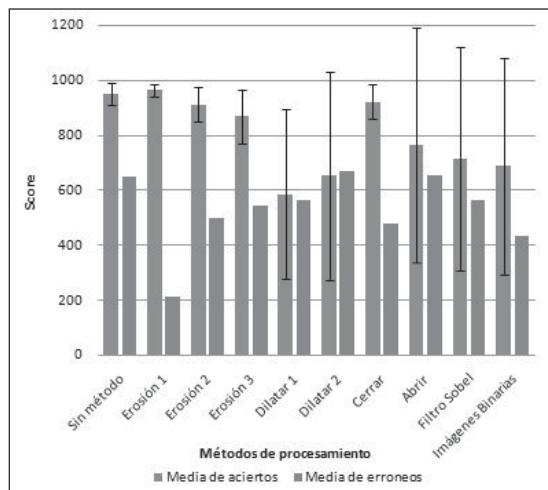


Fig. 11. Gráfica que muestra los resultados obtenidos para los diferentes métodos de procesamiento.

los valores de “score” para los aciertos mantienen una gran cercanía entre ellos. Finalmente, la diferencia entre medias de “score” de aciertos y de erróneos es la más alta para los métodos probados, por lo que la *erosión con una iteración de elemento estructurante de 3x3* evita a toda costa que el sistema asuma una imagen en el campo de juego como ficha, cuando no es tal. Una representación gráfica de estos datos se muestra en la Fig. 11.

6. CONCLUSIONES

Las estructuras de programación estáticas resultaron adecuadas para el desarrollo del algoritmo de juego, debido a las características del mismo tomadas antes del desarrollo del sistema inteligente, específicamente la simetría del tablero de juego, la especificidad de las jugadas y la jerarquización de éstas.

El desarrollo de la cinemática inversa no fue tan necesario como se esperaba, al no requerir movimientos continuos. Solo precisamos de movimientos muy puntuales a las posiciones predefinidas para las fichas involucradas en la partida de “Tres en raya”.

Para el reconocimiento de patrones es más conveniente usar imágenes en escala de grises, ya que el plano de luminancia de una imagen, posee la suficiente cantidad de información para la aplicación.

El método de erosión resultó ser el más adecuado para el reconocimiento de los objetos utilizados en la aplicación, mostrando mejores indicadores que los demás métodos probados.

El prototipo final satisfizo las expectativas esperadas en torno al desarrollo del sistema artificial inteligente, ya que las partidas se juegan de la forma adecuada, y el sistema robótico tiene tal nivel de inteligencia, que no hay posibilidad de derrota.

7. FINANCIAMIENTO

El brazo robótico Lynxmotion ® AL5D, el software citado en la tabla 4 y el ordenador, son financiados por el director del proyecto, ingeniero Henry Moreno, que permite su uso para el desarrollo del proyecto. En caso de dispositivos anexos a la práctica que no sean encontrados en los laboratorios del departamento, los estudiantes se encargan de adquirirlos.

Los laboratorios utilizados corresponden al Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Nacional de Colombia, Edificio 411.

Tabla 4. Descripción de los artículos usados durante el desarrollo del proyecto y su respectivo costo.

Artículo	Valor	
	USD	Pesos (\$)
Brazo robótico Lynxmotion AL5D	473,63	932.933
Programa de control		
RIOS / SSC-32	49,95	98.389
LabVIEW base	1.375,00	2.708.406
Ordenador de escritorio	500,00	984.875
TOTAL	2.398,58	4.724.603

REFERENCIAS

- [1] Logitech, “*Logitech® QuickCam® Connect™ (E2500 Series) User’s Guide*,” Fremont, CA, USA, 2008.
- [2] Intel Corporation, Lucent Technologies Inc, Microsoft Corporation, NEC Corporation, Philips Electronics, Compaq Computer Corporation and Hewlett-Packard Company, *Universal Serial Bus Specification*, USA, 2000.
- [3] Andreas Birk’s Homepage, *USB Protocol Specification*, Andreas Birk’s. [En línea]. Disponible en: http://www.faculty.iubremen.de/birk/lectures/PC1012003/14usb/FINAL%20VERSION/usb_protocol.html
- [4] National Instruments Corporation, *NI Vision Assistant Tutorial*, Austin, TX, USA, 2005.
- [5] National Instruments Corporation, *NI Vision Concepts Manual*, Austin, TX, USA, 2007.
- [6] J. Angulo, F. Ortiz, and F. Torres, *Segmentación de Imágenes Aéreas Mediante Morfología Matemática en Color y Geodesia Cromática*, Alicante, España, 2008. [En línea]. Disponible en: http://www.cea-ifac.es/actividades/jornadas/XXIII/documentos/ja02_060.pdf
- [7] R. Bello and W. Torres, *Procesamiento de Imágenes a Color Utilizando Morfología Matemática*, Caracas, Venezuela, 2004, [En línea]. Disponible en: [http://www.iiisci.org/journal/CV\\$/risci/pdfs/C382LR.pdf](http://www.iiisci.org/journal/CV$/risci/pdfs/C382LR.pdf)
- [8] R. Aycock, *How To Win at Tic Tac Toe*, 2002. [En línea]. Disponible en: <http://www.cs.jhu.edu/~jorgev/cs106/ttt.pdf>
- [9] F. Martínez and M. Castiblanco, “Proyección, diseño y construcción de plataforma robótica para investigación en inteligencia artificial”, *Tecnura*, vol. 14, no. 27, pp. 7-17, Dic. 2010.