

PLANEAMIENTO DE TRAYECTORIAS DE UN ROBOT MÓVIL



Autor:

DIEGO ALEXANDER TIBADUIZA BURGOS

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
MAESTRÍA EN INGENIERÍA
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
ENERO DE 2006**

PLANEAMIENTO DE TRAYECTORIAS DE UN ROBOT MÓVIL



Autor:

DIEGO ALEXANDER TIBADUIZA BURGOS

TRABAJO DE INVESTIGACIÓN

Director:

Dr. Tech ROBERTO MARTINEZ ANGEL

Codirector:

MPE. JAIME GUILLERMO BARRERO PEREZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
MAESTRÍA EN INGENIERÍA
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
ENERO DE 2006**

CONTENIDO

	Pag.
INTRODUCCIÓN	1
1. PLANTEAMIENTO DEL PROBLEMA	3
1.1 ROBOTS MÓVILES	4
1.1.1 Configuraciones de robots móviles	5
2. IMPLEMENTACIÓN DEL SISTEMA	9
2.1 PISTA	9
2.2 ROBOTS MÓVILES	9
2.3 OBSTÁCULOS	10
2.4 SISTEMA DE VISIÓN	10
2.5 COMUNICACIÓN ETHERNET	11
2.6 COMUNICACIÓN VÍA RADIOFRECUENCIA	12
2.7 ULTRASONIDO	12
3. ALGORITMO DE CONTROL	14
3.1 CONTROL DE ALTO NIVEL	14
3.1.1 Descomposición en celdas.	15
3.1.2 Mapa de caminos	16
3.1.3 Campos de potencial	16
3.1.4 Algoritmos genéticos	19
3.2 ALGORITMOS IMPLEMENTADOS	24
3.2.1 Campos de potencial	24
3.2.2 Algoritmos genéticos	36
3.3 ALGORITMO DE CONTROL DE BAJO NIVEL	41
3.3.1 Control de servomotores	44
3.3.2 Modulo de ultrasonido	45
3.3.3 Trayectoria planeada	49
3.3.4 Programación	52
4. VISIÓN DINÁMICA	59
4.1 GENERALIDADES DEL ANÁLISIS DINÁMICO DE	

IMÁGENES	
4.1.1 Definición	59
4.1.2 Etapas fundamentales de la aplicación implementada	59
4.2 GENERALIDADES	61
4.2.1 Descripción de la escena	61
4.2.2 Elementos de adquisición	62
4.2.3 Algoritmo de adquisición y configuración de la tarjeta digitalizadora.	66
4.3 PRE-PROCESAMIENTO	69
4.4 DETECCIÓN Y SEGMENTACION DE MOVIMIENTO	69
4.4.1 Método basado en Color	69
4.4.2 Algoritmo de color	73
4.4.3 Método basado en Gradientes	76
4.4.4 Método Basado En Correlación	80
4.5 LOCALIZACIÓN E IDENTIFICACION	84
4.5.1. Localización de objetos en la imagen	85
4.5.2. Algoritmo de Identificación	85
4.5.3 Cálculo de Orientación del móvil	89
5. PRUEBAS Y ANÁLISIS DE RESULTADOS	92
5.1 PRUEBAS REALIZADAS PARA EL SISTEMA DE VISIÓN	92
5.1.1 Posición	93
5.1.2 Orientación	94
5.1.3 Seguimiento	95
5.1.4 Tiempo de latencia	98
5.2 CARACTERIZACION DEL MODULO DE ULTRASONIDO SRF04	99
5.3 PRUEBAS DE DESEMPEÑO DEL ROBOT MÓVIL	101
5.3.1 Comunicación	102
5.3.2 Precisión	102
5.3.3 Repetibilidad	

5.4 PROGRAMAS ELABORADOS	107
5.4.1 Campos de Potencial	107
5.4.2 Algoritmos Genéticos	109
5.4.3 Sistemas de referencia	111
6. OBSERVACIONES Y CONCLUSIONES	113
7. RECOMENDACIONES PARA TRABAJOS FUTUROS	118
8. REFERENCIAS BIBLIOGRAFICAS	119

LISTA DE FIGURAS

	Pag.
Figura 1. Celda de manufactura Universidad Pontificia Javeriana Cali	3
Figura 2. Configuración síncrona	5
Figura 3. Configuración diferencial	6
Figura 4. Configuración diferencial	6
Figura 5. Triciclo clásico	7
Figura 6. Configuración Ackerman	7
Figura 7. Elementos de la adquisición de imágenes	11
Figura 8. Módulo de Ultrasonido	12
Figura 9. Esquema general de la implementación.	13
Figura 10. Vista del área de trabajo	13
Figura 11. Diagrama de bloques de algoritmo del sistema	14
Figura 12. Problema de un mínimo local en campos de potencial	19
Figura 13. Esquema básico del operador de cruce	23
Figura 14. Representación del área de trabajo cuando la pista es de 20x20	26
Figura 15. Potencial alrededor del punto de llegada.	27
Figura 16. Campo de potencial alrededor de un obstáculo	28
Figura 17. Vista general de la pista.	29
Figura 18. Representación de los datos ingresados al algoritmo con un obstáculo fijo	30
Figura 19. Representación de los datos ingresados al algoritmo con un obstáculo fijo	31
Figura 20. Resultados obtenidos luego de ejecutar el algoritmo.	32
Figura 21. Representación de los movimientos.	32
Figura 22. Definición de puntos vecinos dentro de una matriz	33
Figura 23. Distribución de los obstáculos, punto de inicio y punto	34

de llegada.	
Figura 24. Resultados obtenidos para la distribución definida en la figura 23.	34
Figura 25. Resultados del método de campos de potencial con 3 obstáculos	35
Figura 26. Resultados obtenidos cuando se tienen 6 obstáculos.	36
Figura 27. Ejemplo de trayectoria codificada.	38
Figura 28. Método de Cruce	39
Figura 29. Camino exitoso sin zonas de riesgo en los obstáculos, generación 20	39
Figura 30. Camino exitoso sin zonas de riesgo, generación 110.	40
Figura 31. Camino exitoso con grilla de 32 celdas, considerando zonas de riesgo (generación 205).	40
Figura 32. Diagrama de bloques del sistema de control de bajo nivel.	41
Figura 33. Módulo de Radiofrecuencia	41
Figura 34. Circuito experimental con un 555	43
Figura 35. Tren de pulsos generado por el circuito.	43
Figura 36. Señal de control para el servomotor.	44
Figura 37. Conexiones del modulo del ultrasonido SRF04	46
Figura 38. Diagrama de Tiempos del SRF04	47
Figura 39. Rango efectivo del SRF04.	48
Figura 40. Captura del pulso de eco.	49
Figura 41. División del Byte para Algoritmo Genético	50
Figura 42. División del Byte para la programación con Campos de Potencial	51
Figura 43. Diagrama de flujo del móvil controlado.	53
Figura 44. Obstáculo Móvil	54
Figura 45. Diagrama de Flujo del móvil aleatorio	55
Figura 46. Representación de las direcciones del móvil.	56

Figura 47. Diagrama de Subprograma de Giro	57
Figura 48. Diagrama de subprograma Avanzar.	58
Figura 49. Diagrama de bloques de la aplicación implementada	61
Figura 50. Distribución general de la escena	62
Figura 51. Foto tomada sin calibración	63
Figura 52. Foto obtenida después de la calibración	63
Figura 53. Foto tomada sin calibración	64
Figura 54. Foto obtenida después de la calibración	64
Figura 55. Modelo de distorsión completo para lente de 4 mm	65
Figura 56. Diagrama de flujo para algoritmo de adquisición	67
Figura 57. Adquisición de video y frames	68
Figura 58. Diagrama cromático CIE 1931 para un observador de 2°	71
Figura 59. Observador de 2°.	71
Figura 60. Espacio y color HLS	72
Figura 61. Diagrama de flujo para Algoritmo de Color	75
Figura 62. Segmentación del color amarillo con modelo HLS.	76
a) Imagen original. b) Binarización. c) Imagen con filtrado morfológico	
Figura 63. Segmentación del color naranja con modelo xyY	76
a)Imagen original. b) Binarización. c) Imagen con filtrado morfológico	
Figura 64. Diagrama para Algoritmo basado en Gradientes	78
Figura 65. Operaciones del algoritmo de Jain	79
Figura 66. Segmentación de movimiento con gradientes.	80
a) imagen $A(t_0)$. b)imagen $A(t_0+\Delta t)$. c)Gradiente espacial.	
d)Gradiente temporal. e)bordes del objeto en movimiento	
f)objeto segmentado	
Figura 67. Diagrama del algoritmo de correlación	83
Figura 68. Resultados del algoritmo de correlación	

Figura 69. Diagrama de flujo para algoritmo de identificación	86
Figura 70. Dimensiones de los móviles	87
Figura 71. Identificación de un móvil	88
Figura 72. Diagrama par el algoritmo alterno	89
Figura 73. Sistema de referencia fijado en el centro de la escena	90
Figura 74. Sistema de referencia variable	91
Figura 75. Posiciones usadas para determinar la distribución del error.	93
Figura 76. Distribución del error de posición en los algoritmos implementados	94
Figura 77. Error en orientación.	95
Figura 78. Seguimiento de una trayectoria recta a)con el modelo de color HLS b)con el modelo de color xyY c)con el algoritmo basado en gradientes d)con correlación	96
Figura 79. Seguimiento de una trayectoria circular a)con el modelo de color HLS b)con el modelo de color xyY c)con el algoritmo basado en gradientes d)con correlación	97
Figura 80. Caracterización del sensor SRF04	101
Figura 81. Dispersión de los datos en la prueba de repetibilidad.	106
Figura 82. Interfaz off-line de capos de potencial	107
Figura 83. Interfaz off-line que lee los valores de los obstáculos fijos dentro de la pista usando cámara.	108
Figura 84. Interfaz gráfica programa de planeación de trayectorias usando algoritmos genéticos.	110
Figura 85. Ejemplo de trayectoria obtenida con freyja.	110
Figura 86. Ejemplo de trayectoria al ejecutar Freyja.	111
Figura 87. Sistemas de referencia	112

LISTA DE TABLAS

Tabla 1. Codificación de cromosomas.	37
Tabla 2. Función y configuración de puertos.	42
Tabla 3. Asignación de puertos para los servomotores.	45
Tabla 4. Asignación de puertos para el módulo de ultrasonido.	49
Tabla 5. Acción del móvil para AG.	50
Tabla 6. Acción del móvil para CP.	51
Tabla 7. Tiempos de latencia	99
Tabla 8. Velocidad máxima de procesamiento.	99
Tabla 9. Resultados del laboratorio en la calibración del SRF04.	100
Tabla 10. Prueba de precisión en el robot móvil	103
Tabla 11. Resultados de la prueba de repetibilidad.	105

LISTA DE ANEXOS

Anexo A. Construcción del robot móvil	Pág 123
Anexo B. Principales características de la tarjeta digitalizadora dfg/compress piranha y de la cámara DFK 50H13	133
Anexo C. Instalación e interfase grafica de "fenix"	145

RESUMEN

TITULO: PLANEAMIENTO DE TRAYECTORIAS DE UN ROBOT MÓVIL*

AUTOR: TIBADUIZA BURGOS, DIEGO ALEXANDER**

PALABRAS CLAVES: planeamiento de caminos, campos de potencial, algoritmos genéticos, radiofrecuencia.

DESCRIPCIÓN

En este trabajo de investigación se presenta el desarrollo de dos técnicas de planeamiento de trayectorias. Una de las técnicas es campos de potencial y la otra corresponde a los algoritmos genéticos.

Cada una de estas técnicas se probó sobre un sistema físico que consta de una pista de madera de 1,50 m * 2,40 m, un sistema de visión artificial, dos robots móviles tipo diferencial y unos obstáculos fijos en forma de tetraedro.

El algoritmo de campos de potencial define unos pesos específicos al área de trabajo, para de esta manera diferenciar a los obstáculos y generar un camino libre de colisiones.

El algoritmo genético emula tres operaciones básicas como son reproducción, cruce y mutación para crear trayectorias cada vez mejores en función de un peso específico.

Todo el sistema es sensado con una cámara cuyas imágenes son procesadas por un computador, el cual genera la posición y orientación de cada objeto sobre el área de trabajo. Estos datos son leídos por los algoritmos de control, los cuales generan la trayectoria y envían por puerto serial a un módulo de RF para cada móvil.

El ambiente dinámico se simula con uno de los robots móviles, el cual posee un modulo de ultrasonido que sensa el ambiente y se mueve por el sin ninguna trayectoria planeada.

* Work of Master Investigation.

**School of Electronic, Electric Engineering and of Telecommunications. Roberto Martinez Angel.

ABSTRACT

TITLE: PATH PLANNING IN A MOBILE ROBOT*

AUTHOR: TIBADUIZA BURGOS, DIEGO ALEXANDER**

KEYWORDS: path planning, potential fields, genetic algorithms, radiofrecuency.

DESCRIPTION:

In this work of investigation the development of two techniques of planning of trajectories is presented. One of the techniques is potential fields and the other corresponds to the genetic algorithms.

Each one of these techniques was proven on a physical system that consists of a wood track of 1.50 m * 2.40 m, a system of artificial vision, two robots mobile type differential and fixed obstacles with tetrahedric form.

The algorithm of potential fields defines specific weights to the work area, for this way differentiating to the obstacles and generating a way free of collisions.

The genetic algorithm emulates three basic operations as they are reproduction, crosses and mutation to create better trajectories every time based on a specific weight.

All the system is sense with a camera whose images are processed in a computer, which generates the position and direction of each object on the work area. These data are read by the control algorithms, which generate the trajectory and send by serial port to a module of RF for each mobile.

The dynamic area is simulated with one of robots mobile, which has I modulate of ultrasound that sense the area and moves by without no planned trajectory.

* Work of Master Investigation.

**School of Electronic, Electric Engineering and of Telecommunications. Roberto Martinez Angel.

INTRODUCCIÓN

La robótica es un campo de amplias perspectivas para el desarrollo industrial que en un tiempo relativamente corto podría permitir a países en desarrollo apropiarse de técnicas que utilizando las vías tradicionales podrían requerir un tiempo considerable.

Cada vez más los robots deben interactuar en ambientes cuyos espacios de trabajo no son estáticos, lo cual le exige al robot herramientas que le permitan desempeñarse adecuadamente a pesar de los cambios que se presenten, es ahí donde el planeamiento de trayectorias se presenta como una solución.

En este trabajo de investigación de Maestría se busca implementar estrategias para planeamiento de trayectorias de un robot móvil en un ambiente cambiante en el cual se mueve otro robot en presencia de varios obstáculos. El objetivo es llevar al móvil desde el punto origen hasta el punto destino dentro del área de trabajo, evitando todo tipo de colisiones. El lazo de realimentación lo constituye una cámara que permite determinar en todo instante la posición de los dos robots y de los obstáculos estacionarios. El algoritmo puede ser aplicado en un terreno como una pista con características similares a las de la ROBOCUP ó en cualquier tipo de aplicación en las que se tenga las coordenadas de los obstáculos y puntos de interés.

El presente trabajo está enmarcado dentro de las líneas de investigación y desarrollo del grupo CEMOS¹ de la Escuela de ingenierías Eléctrica, Electrónica y Telecomunicaciones y nace de la necesidad de fortalecer la línea de robótica en la E3T y de mejorar el laboratorio de robótica de la asignatura que se imparte en la Universidad Industrial de Santander.

¹ Control, Electrónica, Modelamiento y Simulación.

Este proyecto de investigación incluye la elaboración de un algoritmo de visión dinámica encargado de sensar en un plano 2D, la posición de dos obstáculos fijos, un obstáculo móvil y la posición del móvil que ejecuta la trayectoria. Para el planeamiento y ejecución de la trayectoria, se implementó dos programas que tienen por objeto, encontrar y ejecutar una trayectoria libre de colisiones para un robot móvil. Estos programas se basan en la técnica de algoritmos genéticos y de campos de potencial.

El presente documento se ha dividido en cuatro capítulos: en el primero se hace la descripción del planteamiento del problema y una recopilación de conceptos básicos acerca de la robótica.

En el segundo capítulo, se hace una descripción de la implementación del sistema junto con los componentes generales. El algoritmo de control se detalla en el tercer capítulo, en este se describe tanto el control de alto nivel encargado de generar las trayectorias como el control de bajo nivel, encargado de ejecutar sobre el móvil las acciones necesarias para que pueda ser ejecutada la trayectoria.

En el capítulo cuatro se describe el algoritmo de visión dinámica implementado y en el capítulo cinco se presentan las diferentes pruebas y resultados, por último se encuentran las conclusiones y observaciones a las que se llegaron con unas posibles recomendaciones para futuras mejoras.

Los anexos hacen referencia a la construcción del robot, características de la tarjeta de digitalización usada y a una breve explicación de la interfaz de visión.

1. PLANTEAMIENTO DEL PROBLEMA

En un sistema automatizado de producción se tienen diferentes elementos tales como: máquinas, herramientas, depósito de materia prima y anaqueles donde se almacena el stock, (ver figura 1). Se necesita poseer estibadoras inteligentes que sean capaces de llevar los productos terminados desde las máquinas a la cadena de ensamble y luego a los anaqueles correspondientes.

Figura 1. Celda de manufactura Universidad Pontificia Javeriana Cali



En este contexto el interés de la robótica se centra en los manipuladores, robots que manejan las celdas de manufactura y robots móviles para realizar el transporte de material de una celda a otra o a la cadena de montaje.

De acuerdo a esto se puede tener:

Manipuladores simples

Robots estacionarios

Robots móviles

El movimiento de las piezas implica desplazamiento no sólo de la pieza sino de los eslabones del robot dentro del espacio de trabajo, el movimiento debe hacerse de manera segura para evitar daños tanto del robot como de la pieza que se transporta.

Es ahí cuando es necesario contar con estrategias que permitan mover un robot de un punto a otro mediante una trayectoria planeada para evitar colisiones en el área de trabajo.

La trayectoria debe entonces, planearse cuidadosamente teniendo en cuenta que los ambientes dentro del sistema automatizado puede estar siempre demarcados por áreas de trabajo constantes como también por áreas de trabajo que pueden variar.

El planeamiento de trayectorias se usa para encontrar trayectorias posibles en un ambiente de trabajo. Existen diferentes métodos para encontrar trayectorias y estos pueden ser aplicados a robots móviles y estacionarios. El objetivo del presente trabajo es realizar el planeamiento de trayectorias de un robot móvil; a continuación se presentan algunos conceptos básicos de la robótica móvil.

1.1 ROBOTS MÓVILES

Son robots cuyo desplazamiento no está restringido a la máxima longitud de sus articulaciones.

Los vehículos con ruedas son la solución más simple y eficiente para conseguir la movilidad en terrenos suficientemente duros y libres de obstáculos, permitiendo conseguir velocidades relativamente altas.

1.1.1 Configuraciones de robots móviles

Las configuraciones usadas para el movimiento de robots móviles suelen ser diversas, algunas más complejas que otras, pero por lo general para el caso práctico se suele trabajar con cuatro tipos de configuraciones: la síncrona, la diferencial, el triciclo clásico y la configuración de Ackerman.

La configuración síncrona es aquella en la cual hay transmisiones que permiten orientar las tres ruedas simultáneamente con una velocidad angular ω y hacer que el vehículo se desplace con una velocidad angular v tal como se aprecia en la figura 2:

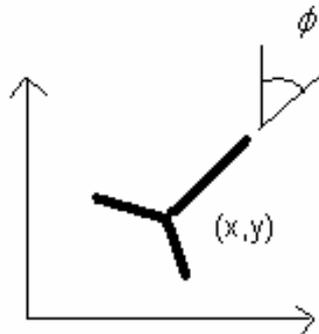
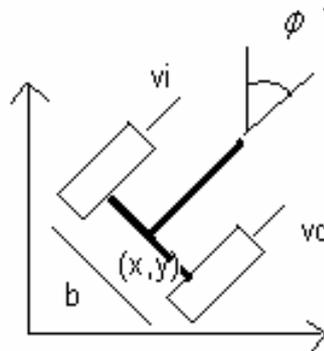


Figura 2. Configuración síncrona

Fuente: ANÍBAL OLLERO BATURONE. ROBÓTICA, MANIPULADORES Y ROBOTS MÓVILES. MARCOMBO S.A. 2001.

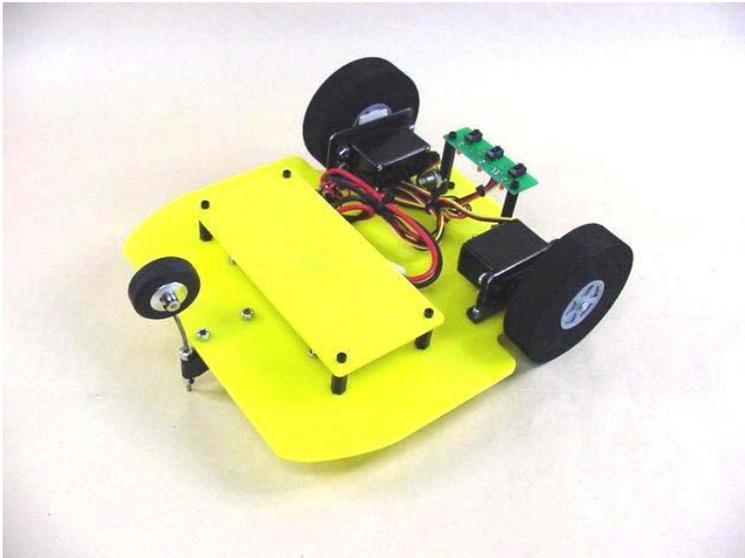
La locomoción con guiado diferencial es aquella en la que las variables de control son las velocidades de las ruedas laterales v_l y v_d , tal como se puede observar en las figuras 3 y 4.

Figura 3: Configuración diferencial



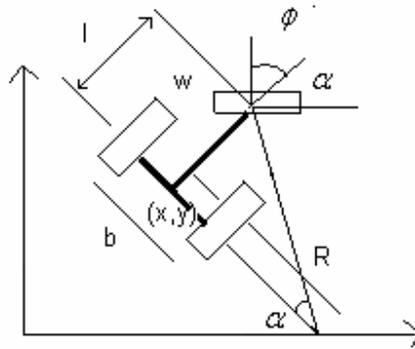
Fuente: ANÍBAL OLLERO BATURONE. ROBÓTICA, MANIPULADORES Y ROBOTS MÓVILES. MARCOMBO S.A. 2001.

Figura 4. Configuración diferencial



El triciclo convencional es aquel en que la rueda delantera se utiliza tanto para la orientación como para la tracción. Las variables de control suelen tomarse como el ángulo α de dirección de la rueda delantera y la velocidad de giro de la misma rueda w_t . Aquí se supone que el eje guía (x,y) está en el eje trasero. El esquema de esta configuración se puede apreciar en la figura 5.

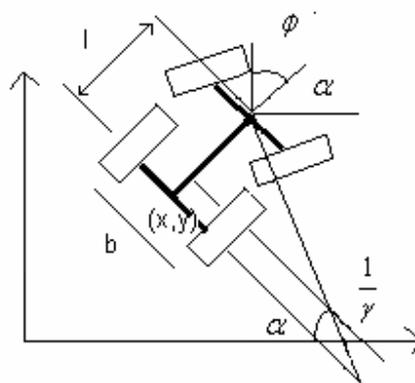
Figura 5. Triciclo clásico



Fuente: ANÍBAL OLLERO BATURONE. ROBÓTICA, MANIPULADORES Y ROBOTS MÓVILES. MARCOMBO S.A. 2001.

El último tipo de configuración a considerar es la de Ackerman tal como se aprecia en la figura 6. Se supone que el centro guiado del vehículo está sobre la mitad del eje de las ruedas de tracción (ruedas traseras). En este modelo la velocidad real de cada rueda es diferente.

Figura 6. Configuración Ackerman



Fuente: ANÍBAL OLLERO BATURONE. ROBÓTICA, MANIPULADORES Y ROBOTS MÓVILES. MARCOMBO S.A. 2001.

Para el presente trabajo de investigación se ha elegido la configuración diferencial porque es una configuración cuyo movimiento es sencillo de controlar, basta con manejar el movimiento de las dos ruedas impulsoras mediante dos motores para realizar desplazamientos lineales y angulares. Con esta configuración hay un consumo de potencia menor que en otras configuraciones, lo que garantiza un mayor tiempo de funcionamiento con una fuente de alimentación portátil.

2. IMPLEMENTACIÓN DEL SISTEMA

Se implementó una plataforma de pruebas para planeamiento de trayectorias de robots móviles consistente en:

Una pista

Robots Móviles

Obstáculos

Sistema de Visión

Comunicación Ethernet

Comunicación vía Radiofrecuencia

Sensado con ultrasonido

A continuación se hará una descripción de cada uno de los elementos que componen la plataforma de pruebas:

2.1 PISTA

La pista elaborada consiste en una plataforma de madera de dimensiones 2.44 m *1.52 m, la cual fue pintada de color negro opaco para resaltar los elementos que se encuentran en la escena y evitar que el algoritmo de visión capture datos errados por las sombras que se puedan presentar.

A esta pista se le construyó una barrera por todo el borde con el fin de indicarle al robot al usar ultrasonido que al frente existe un orillo.

2.2 ROBOTS MÓVILES

Se construyeron dos robots móviles para la elaboración de este trabajo de investigación. Los robots son de guiado diferencial, poseen dos ruedas matrices, en las cuales se utiliza servomotores para el accionamiento de las y una rueda loca.

Estos robots son alimentados cada uno por un banco de 6 baterías recargables de 1.2 V.

Uno de los robots es usado para ejecutar el planeamiento de trayectorias y el otro se usa como obstáculo dentro de la pista. Tanto el robot que ejecuta la trayectoria como el obstáculo móvil son controlados cada uno por un Microcontrolador MOTOROLA MC68HC908GP32.

En el caso del obstáculo móvil el microcontrolador se encarga de trabajar con un programa para el control del carro mediante ultrasonido, es decir el robot se mueve de manera aleatoria por la pista dependiendo de lo que se encuentre sobre ella.

El microcontrolador del robot móvil que ejecutará la trayectoria es el encargado de recibir e interpretar las señales de radiofrecuencia enviadas desde el host de control para ejecutarlas sobre los motores. Además se encarga de recibir las señales del módulo de ultrasonido para evitar colisionar en caso de que ocurra alguna anomalía en el proceso.

2.3 OBSTÁCULOS

Se construyeron dos obstáculos fijos y un obstáculo móvil. El obstáculo móvil es el que se describió en el numeral anterior, los obstáculos fijos son cubos de acrílico cuyas medidas son: 5 cm de alto, 5 cm de largo y 5 cm de ancho; se escogió acrílico para evitar que en caso de existir una colisión no haya daños en el móvil ni en los obstáculos. El color seleccionado fue azul para la realización de las pruebas, aunque este puede cambiar simplemente seleccionado el color en el algoritmo de visión artificial.

2.4 SISTEMA DE VISIÓN

El sistema de visión artificial esta compuesto por:

1 cámara análoga

1 tarjeta digitalizadora

Para la obtención de la señal que se va a digitalizar se hizo uso de la cámara análoga DFK50H13/N existente en el laboratorio de robótica que posee un lente de 8mm y salida de video compuesto. A esta cámara fue necesario adaptarle un lente de 4mm para lograr obtener datos de un área mayor.

La digitalización de las señales provenientes de la Cámara, se realiza usando una tarjeta de compresión de video PCI de 32 bits DFG/Compress con una entrada / salida de video compuesto.

La tarjeta digitalizadora se encuentra instalada en un PC con las siguientes características: Intel Pentium 4 de 1.5 GHz y 256 MB de RAM, en este computador se encuentra el algoritmo de visión artificial, el cual está basado en detección de color para detectar la posición de los objetos móviles y estáticos sobre la pista. Los elementos de la adquisición se pueden apreciar en la figura 7.

Figura 7. Elementos de la adquisición de imágenes



2.5 COMUNICACIÓN ETHERNET

Es la encargada de transmitir los datos de posición desde el computador con el algoritmo de visión al computador con los algoritmos de control. Estos datos son las entradas de los programas que generan la trayectoria.

Ethernet es una Norma o estándar (IEEE 802.3) que determina la forma en que los puestos de la red envían y reciben datos sobre un medio físico compartido que se

comporta como un bus lógico, independientemente de su configuración física. Ethernet es la capa física más popular de la tecnología LAN usada actualmente.

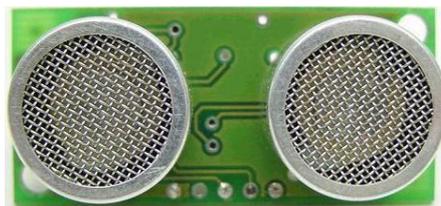
2.6 COMUNICACIÓN VÍA RADIOFRECUENCIA

Esta comunicación se realiza entre el host de control y el móvil que ejecuta la trayectoria a través de unos módulos de radiofrecuencia que transmiten a una frecuencia de 433 Mhz, usando modulación AM. Estas señales son enviadas a través de puerto serial desde el host de control al módulo de transmisión y son transmitidas y recibidas de forma serial.

2.7 ULTRASONIDO

Se usa como método de sensado local para detectar si existe algún obstáculo en la cercanía de alguno de los dos móviles. Este sistema sirve de respaldo para informarle al microcontrolador en caso de existir alguna anomalía en el sistema de detección por cámara o errores en la comunicación. En la figura 8 se puede apreciar el módulo usado.

Figura 8. Módulo de Ultrasonido



Una vista general de los diferentes sistemas implementados se puede apreciar en la figura 9. La figura 10 representa la escena en la que se realiza la planeación.

Figura 9. Esquema general de la implementación.

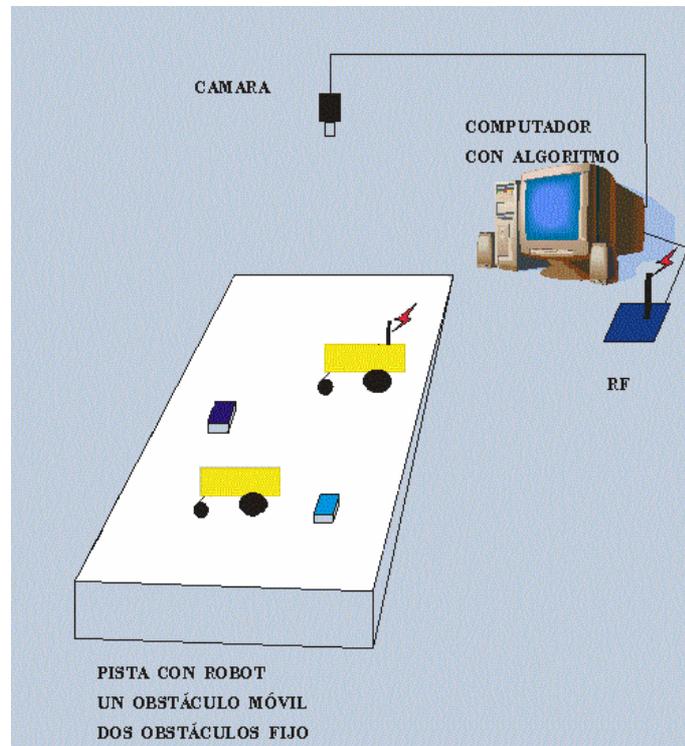
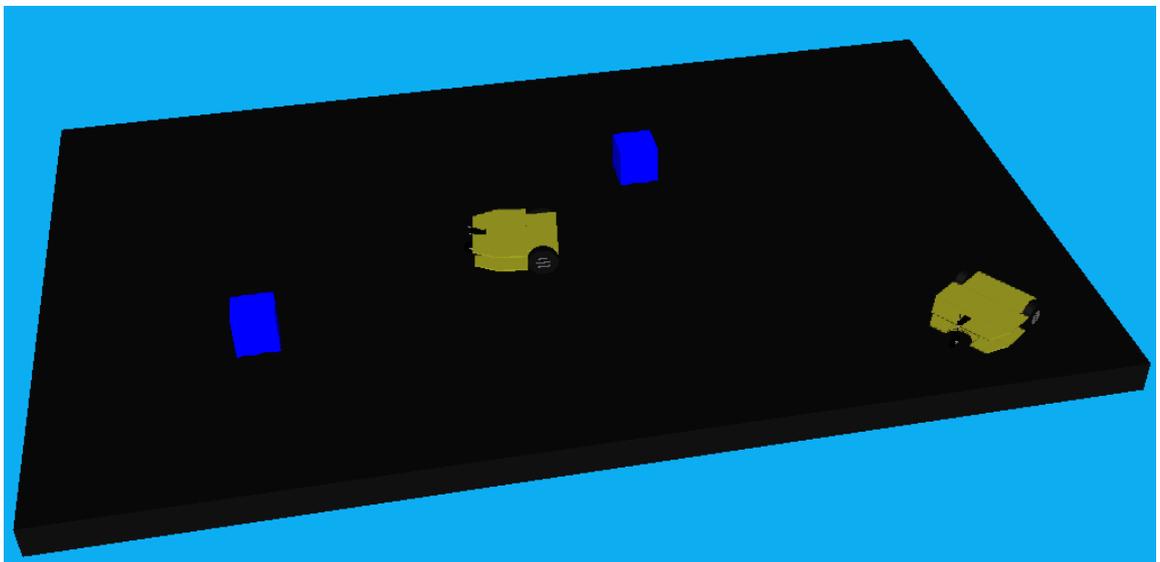


Figura 10. Vista del área de trabajo



3. ALGORITMO DE CONTROL

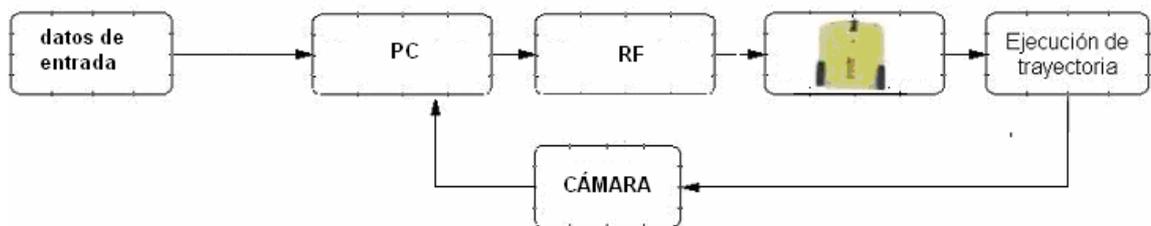
El sistema de control implementado en el presente trabajo de investigación se puede dividir en control de bajo nivel y control de alto nivel.

El control de bajo nivel se realizó en un microcontrolador MOTOROLA MC68HC908GP32, el cual es el encargado de recibir las señales de control vía radiofrecuencia e interpretarlas para ser ejecutadas sobre los servomotores del móvil. Dentro de este control se encuentra el módulo de ultrasonido el cual realimenta al microcontrolador la información de su entorno mas próximo y sirve como soporte para cualquier falla en el sistema.

El control de alto nivel se desarrollo en un computador cuyas características son: Pentium 4 de 2.4 Ghz y RAM de 512 MB con el cual se corre alguna de las dos técnicas implementadas como son campos de potencial y algoritmos genéticos.

Tanto el control de bajo nivel como el de alto nivel están enmarcados en el diagrama de bloques de la figura 11.

Figura 11. Diagrama de bloques de algoritmo del sistema



3.1 CONTROL DE ALTO NIVEL

El planeamiento de trayectorias es una de las tareas más importantes en el control inteligente de un robot móvil autónomo. El movimiento del robot es a menudo descompuesto en planeamiento de caminos y planificación de trayectorias, aunque ellos no son independientes el uno del otro.

El planeamiento de caminos consiste en generar un camino libre de colisiones en un ambiente con obstáculos y optimizar este con respecto a algún criterio.

Un algoritmo para planeamiento de caminos se dice que esta off-line (fuera de línea) si el ambiente es estático y conocido y el algoritmo genera por anticipado un camino. Se dice que esta on-line (en línea) si este es capaz de producir un nuevo camino en respuestas a cambios en el ambiente.

La planeación de trayectorias consiste en fijar el movimiento de un robot móvil a lo largo de un camino planeado.

Existen numerosos métodos para la planeación de movimiento de un robot móvil. Pocos algoritmos se han desarrollado para planeamiento de movimiento on-line de un robot móvil en un terreno desconocido.

De forma general se puede hablar de tres métodos tradicionales:

- métodos de descomposición en celdas
- métodos de mapa de caminos (road-map)
- métodos de campo potencial.

Cualquiera de estos métodos actúa bien tanto en el *espacio de trabajo* del robot, como en el espacio de configuraciones. Se llama espacio de trabajo al lugar donde se encuentran el robot y los obstáculos y espacio de configuración al espacio que consiste en todas las posibles configuraciones del robot.

3.1.1 Descomposición en celdas.

El método de descomposición en celdas construye un camino en el espacio de configuración dividiendo el espacio libre, esto es, la parte del espacio de configuración donde el robot no colisiona con ningún obstáculo en simples celdas; esto puede hacerse de forma exacta o aproximada. Después, las celdas son

insertadas en un grafo en el que celdas adyacentes están conectadas. Entre estas celdas, el movimiento es normalmente fácil de calcular. El movimiento que conecta la configuración inicial y la final puede entonces calcularse determinando las celdas que contienen las configuraciones inicial y final, encontrando un camino entre estas celdas en el grafo, y finalmente calculando el movimiento entre cada arista del camino en el grafo.

3.1.2 Mapa de caminos

El método de mapa de caminos intenta construir un grafo de caminos, por los que es seguro viajar, entre dos obstáculos. Esto puede hacerse, bien para el espacio de trabajo o para el de configuración. Una vez que se tiene el grafo disponible, el movimiento se calcula moviendo el robot desde la configuración inicial hasta alguna posición en un camino cercano, siguiendo después las diferentes rutas hasta acercarse a la configuración final dejando la última ruta y moviendo el robot hasta la configuración final.

Tanto descomposición de celdas y mapa de caminos son llamados métodos globales ya que primero construyen una estructura de datos que es usada luego para encontrar los caminos entre dos configuraciones distintas del robot. Una ventaja de estos métodos es que la estructura de datos sólo debe ser calculada una sola vez, y puede luego ser usada nuevamente para muchas configuraciones inicial-final distintas. Sin embargo, estas estructuras de datos tienden a ser muy grandes, y los cálculos geométricos requeridos son a menudo dificultosos y requieren mucho cálculo computacional.

3.1.3 Campos de potencial

El método de campo de potencial es conocido como un método local, ya que sólo calcula el movimiento para una configuración inicial-final dada. Comienzan en la inicial e intentan mover el robot en pequeños pasos hacia el objetivo.

Este método esencialmente trata el ambiente de trabajo como un campo de fuerzas, dando un valor diferente a cada objeto sobre este. El objetivo o punto de llegada se considera como un imán de polaridad contraria a la del móvil, es decir que tiene fuerza atractora mientras que los obstáculos se comportan como imanes de igual polaridad emitiendo fuerzas repulsivas. Este método considera la velocidad del móvil en la cercanía del obstáculo.

Ha sido usado ampliamente para robots móviles así como para manipuladores con excelentes resultados, aunque su acogida es mayor en los robots móviles donde se han trabajado tanto en planeamiento on-line como off-line.

Básicamente la planificación se puede trabajar como sigue:

- 1) Considerar que cada obstáculo genera una fuerza repulsora normal a la superficie sobre la que se realiza el planeamiento y que la posición objetivo es una fuerza de atracción.
- 2) Calcular el vector de fuerzas resultante
- 3) Calcular una nueva posición para el robot como resultado de aplicar una fuerza aceleradora.
- 4) Regresar al paso 1.

- El campo de potencial diferencial se construye sumando el campo de la meta, U_g , y el campo de los obstáculos, U_o :

$$U(q) = U_g(q) + \sum U_o(q)$$

- A partir de este campo se construye un campo de fuerzas artificial resultante:

$$F = -\nabla U(q) = \begin{pmatrix} dU/dx \\ dU/dy \end{pmatrix}$$

- Una vez construido el campo de fuerzas, el robot se mueve en la dirección de la fuerza local. Con esto se tiene un esquema robusto que implícitamente tiene un plan de cualquier punto del espacio a la meta

Para obtener las fuerzas hay que modelar las funciones de potencial de la meta y obstáculos calculando el potencial para cada punto del espacio libre:

- Meta – “atractor parabólico”

$$U_g(q) = k_1 \text{dist}(q, \text{meta})^2$$

- Obstáculo – “barrera potencial exponencial”

$$U_o(q) = k_2 \text{dist}(q, \text{obstaculo})^{-1}$$

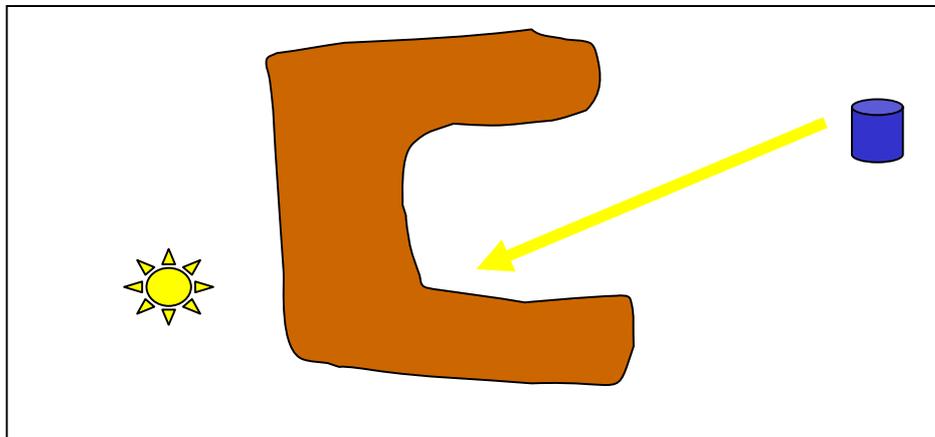
- Ventajas:

- Se pueden generar trayectorias en tiempo real a partir del campo de fuerzas
- Las trayectorias generadas son suaves
- Facilita acoplar los aspectos de planeación y de control

- Desventajas:

- La principal desventaja de este método es que en su forma simplificada con problemas no triviales conduce a que el robot caiga en mínimos hacia un mínimo local en donde las diferentes fuerzas se cancelan entre si tal como se aprecia en la figura 12. Para evitar este problema se han probado varias técnicas de las cuales no se hablará en este plan pero que se pueden consultar en la bibliografía citada.

Figura 12. Problema de un mínimo local en campos de potencial



Fuente: presentación: Robótica Inteligente. L. Enrique Sucar, Marco López

En este trabajo de investigación se trabajó el planeamiento de movimiento adaptativo, el cual puede modificar el camino existente siempre que ocurra un cambio en el ambiente.(e.j. un robot móvil detecta un obstáculo desconocido).

3.1.4 Algoritmos genéticos

Los Algoritmos Genéticos (A.G. ó AG) son métodos de búsqueda basados en mecanismos de selección y genética natural. Estos fueron introducidos por John Holland, sus colegas y estudiantes en la Universidad de Michigan en 1970.

La evolución genética tiene lugar en los cromosomas, en donde está codificada la información del ser vivo. La información almacenada en el cromosoma varía de unas generaciones a otras. En el proceso de formación de un nuevo individuo, se combina la información cromosómica de los progenitores, aunque la forma exacta en que se realiza es aún desconocida. Aunque muchos aspectos están todavía por discernir, existen unos principios generales ampliamente aceptados por la comunidad científica. Algunos de estos son:

1. La evolución opera en los cromosomas en lugar de hacerlo en los individuos a los que representan.

2. La selección natural es el proceso por el que los cromosomas con "buenas estructuras" se reproducen más a menudo que los demás.
3. En el proceso de reproducción tiene lugar la evolución mediante la combinación de los cromosomas de los progenitores.
4. La evolución biológica no tiene memoria en el sentido de que en la formación de los cromosomas únicamente se considera la información del período anterior

FUNDAMENTOS DE GENETICA

Antes de definir un Algoritmo Genético (AG), es necesario poseer algunos conceptos de genética que ayudarán a comprender mejor su analogía y funcionamiento. Las definiciones dadas en este numeral son tomadas del libro *La Salud* del Dr. *Claude B Blouin*:

- **GEN:** Unidad elemental del cromosoma, construida por ácido desoxirribonucleico, que lleva un mensaje hereditario muy preciso, que lo transmite, reproduciéndose y que, bajo la influencia de diversos factores, es capaz de sufrir una mutación que modifica el mensaje hereditario.

- **CROMOSOMAS:** Cada una de las pequeñas formaciones alargadas dispuestas por pares, constituidas esencialmente por DNA.

Los cromosomas contienen la información genética del organismo. Cada tipo de organismo tiene un número de cromosomas determinado; en la especie humana, por ejemplo, hay 23 pares de cromosomas organizados en 8 grupos según el tamaño y la forma. La mitad de cada cromosomas proceden del padre y la otra mitad de la madre.

- **ADN:** El ADN es quien lleva la información necesaria para dirigir la síntesis de proteínas y la replicación. Se llama síntesis de proteínas a la producción de las proteínas que necesita la célula o el virus para realizar sus funciones y desarrollarse. La replicación es el conjunto de reacciones por medio de las cuales el ADN se copia a sí mismo cada vez que una célula o un virus se reproduce y

transmite a la descendencia la información que contiene. En casi todos los organismos celulares el ADN está organizado en forma de cromosomas, situados en el núcleo de la célula.

- **GENOTIPO:** Conjunto de los genes llevados por la totalidad de los cromosomas de un individuo, entre ellos los genes recesivos no se expresan.
- **FENOTIPO:** Conjunto de los rasgos aparentes de un individuo, correspondiente a la fracción observable y exteriorizada del genotipo.
- **ALELO:** Se dice de cada uno de los genes que ocupan el mismo locus en cada uno de los cromosomas de un mismo par y que ejercen, su efecto, idéntico o diferente, sobre un mismo carácter determinado.
- **LOCUS:** Lugar preciso y constante que corresponde a la sede de un gen concreto en cada uno de los cromosomas de un par concreto.
- **MUTACIONES:** Aunque la replicación del ADN es muy precisa, no es perfecta. Muy rara vez se producen errores, y el ADN nuevo contiene uno o más nucleótidos cambiados. Un error de este tipo, que recibe el nombre de mutación, puede tener lugar en cualquier zona del ADN.

La mayoría de las mutaciones genéticas son perjudiciales para el organismo que las porta.

ANALOGIAS CON LOS A.G.

Aunque no se ha establecido directamente un mapeo directo entre los términos de la genética natural y los programados por los A.G. se trabajan básicamente los siguientes:

- **GENES:** Cada BIT(0 y 1 en una cadena binaria) de la cadena.
- **CROMOSOMAS:** Es el conjunto de genes, es decir es cada individuo.
- **GENOTIPO:** Es el conjunto de cromosomas que conforman la población.
- **FENOTIPO:** Es el valor entero del genotipo(al evaluar en la función de evaluación).

ALGORITMO GENETICO SIMPLE

Para llevar a la práctica un esquema de Algoritmo Genético es necesario tener en cuenta los siguientes elementos:

- Una representación cromosómica
- Una población inicial
- Una medida de evaluación
- Un criterio de selección / eliminación de cromosomas
- Una o varias operaciones de recombinación
- Una o varias operaciones de mutación

La mecánica de un A.G. es sorprendentemente simple; nada más complejo que copiar cadenas y partes de estas.

Un A.G. simple que produce buenos resultados esta compuesto de:

- A. Reproducción
- B. Cruce
- C. Mutación

A. REPRODUCCION

Es el proceso por medio del cual cadenas individuales son copiadas de acuerdo al valor de su función objetivo, f , (los biólogos llaman a esta función, propiedad ("**fitness** "). Intuitivamente se puede pensar en la función f como una medida de beneficio, utilidad o bondad que se quiere maximizar. Se copia cadenas acorde a los valores promedio, con lo cual las cadenas con un alto valor tienen la mayor probabilidad de contribuir a la creación de la siguiente generación. Este operador es una versión artificial de la selección natural de Darwin y es quien en definitiva indica quien vive o muere.

El operador de reproducción puede ser implementado por diversos métodos, pero quizá el más usado es el de un individuo basado en una ruleta, donde cada individuo posee un lugar de acuerdo a su función propiedad dentro de esta.

B. CRUCE

Los Operadores de Cruzamiento mas utilizados son:

- De un punto: Se elige aleatoriamente un punto de ruptura en los padres y se intercambian sus bits.
- De dos puntos: Se eligen dos puntos de ruptura al azar para intercambiar.
- Uniforme: En cada bit se elige al azar un padre para que contribuya con su bit al del hijo, mientras que el segundo hijo recibe el bit del otro padre.
- PMX, SEX: Son operadores más sofisticados fruto de mezclar y aleatorizar los anteriores.

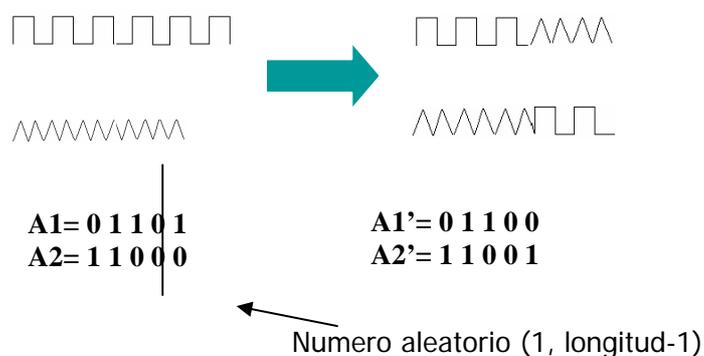
La operación más usada en las aplicaciones es la de cruce usando un solo punto.

A continuación se describen los dos pasos a seguir:

1. Los padres se escogen de forma aleatoria y se acomodan en el **matting pool** (lugar de encuentros para el apareamiento).
2. Cada par de cadenas se cruzan escogiéndose el lugar de cruce con un número aleatorio (1, L-1).

En la figura 13 se puede apreciar como actúa el operador de cruce de una forma gráfica para tener una idea más general de esta operación.

Figura 13. Esquema básico del operador de cruce



C. MUTACIÓN

Es una alteración aleatoria de un valor de posición de una cadena.

Se ha encontrado que la frecuencia de mutación es de una mutación por cada mil bits.

La rata de mutación en poblaciones naturales es igualmente pequeña, por lo que se concluye que es apropiado considerarla como un mecanismo secundario de adaptación de un algoritmo genético.

3.2 ALGORITMOS IMPLEMENTADOS

Para el presente trabajo se uso campos de potencial y algoritmos genéticos como técnica de planeamiento de trayectorias. Se seleccionaron estos dos, porque la idea era comparar un método de inteligencia artificial con un método tradicional. A continuación se describirá lo desarrollado para cada uno de los algoritmos.

3.2.1 Campos de potencial

Este método esencialmente trata el ambiente de trabajo como un campo de fuerzas, dando un valor diferente a cada objeto sobre el área de trabajo. El objetivo o punto de llegada se considera como un imán de polaridad contraria a la del móvil, es decir que tiene fuerza de atracción mientras que los obstáculos se comportan como imanes de igual polaridad generando fuerzas repulsivas. La implementación de este método se realizó usando una matriz para la discretización del terreno, de tal manera que cada punto sobre la matriz es un área de la pista.

Básicamente la planificación se puede trabajar de la siguiente manera:

- Inicializar variables
- Leer y ubicar puntos importantes del terreno sobre la matriz.

- Definir potencial para el punto de llegada
- Definir potencial para cada obstáculo
- Mover el móvil desde el punto de inicio al punto de llegada siguiendo el potencial menor o igual y guardar ese movimiento hasta encontrar el punto de llegada.

Para implementar este método fue necesario tener en cuenta las siguientes consideraciones:

- Existe una discretización del terreno.
- Cada punto de la pista se considera afectado por un valor numérico que indica la relación de este con el punto de llegada. De otra manera cada celda considerada posee un potencial que indica si la trayectoria puede o no pasar por allí.
- De acuerdo con la distribución numérica asignada, la trayectoria se hace siguiendo un potencial descendente
- El algoritmo puede ser on-line si se actualiza periódicamente los valores detectados por una cámara o puede ser off-line si se posee la información de la ubicación de los obstáculos solo al comienzo del programa, con lo cual sólo se encuentra un camino sin poder revisar si algún obstáculo ha entrado a la trayectoria planeada.
- El algoritmo es ajustable al área de trabajo. Con esto se trata de generar algo de robustez lo cual permite usar el mismo programa para diferentes tamaños de pista. En este caso el área de trabajo quedaría definida por las características del sistema de visión tales como altura de la cámara y lentes usados.

En el caso de planeamiento on-line la velocidad de actualización del algoritmo depende del tiempo que demore la actualización de los datos de la pista, es decir

el tiempo de respuesta del algoritmo de visión. También dependerá del lenguaje usado y la velocidad de procesamiento del computador usado

INICIALIZACIÓN DE LAS VARIABLES

Es la primera etapa del algoritmo, que consiste en obtener la lectura de los datos tales como: posiciones de inicio, llegada, posiciones de los obstáculos y tamaño de la pista. Cuando el planeamiento se hace on-line estos datos se obtienen del algoritmo de visión y se actualizan periódicamente dentro del programa, para determinar si el camino se cruza o no con un obstáculo.

Una vez conocido el tamaño del área de trabajo se crea una matriz para simular la pista tal como se muestra en la figura 14 y esta se inicializa con ceros. Cada elemento de la matriz corresponde a un area determinada, con lo cual si el terreno de trabajo es por ejemplo de (200*200) cm² cada elemento de la matriz corresponde con una area de (10*10) cm² en la pista. El área de cada celda puede ajustarse sin importar si el área de trabajo no es cuadrada.

Figura 14. Representación del área de trabajo cuando la pista es de 20x20

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

En el caso de la pista trabajada, el área es de $(150 \times 240) \text{ cm}^2$, por lo cual la pista se dividió en áreas de $(5 \times 5) \text{ cm}^2$.

CREACIÓN DEL POTENCIAL ALREDEDOR DEL PUNTO DE LLEGADA

Para poder generar un potencial que haga converger todos los caminos al punto de llegada, se le asigna a este punto un valor numérico igual a "0" y a partir de este se generan valores numéricos ascendentes hasta cubrir la totalidad de la pista, tal como se puede observar en la figura 15. El único punto que queda sin potencial es el de inicio, el cual se representa con otro valor superior al potencial mayor en la pista.

Figura 15. Potencial alrededor del punto de llegada.

3	3	3	3	3	3	3
3	2	2	2	2	2	3
3	2	1	1	1	2	3
3	2	1	0	1	2	3
3	2	1	1	1	2	3
3	2	2	2	2	2	3
3	3	3	3	3	3	3

Realizando esta configuración en la pista, se implementa la fuerza de atracción con lo cual se asegura que desde cualquier punto siguiendo una secuencia descendente de números se converge al punto de llegada.

CREACIÓN DEL POTENCIAL ALREDEDOR DE UN OBSTÁCULO

Para incluir los obstáculos es necesario tener en cuenta las siguientes características:

- El obstáculo posee un área efectiva propia de sus dimensiones, a través de la cual no se puede realizar una trayectoria porque habría colisión. Este obstáculo se representa en el algoritmo con el número “200”, aunque el valor puede ser cualquiera mayor que los valores del campo de potencial ya definidos en la pista.
- Cuando el obstáculo tiene una forma irregular diferente al valor de discretización del terreno, se considera como zona de riesgo los alrededores de éste, lo cual impide que una trayectoria sea definida en sus cercanías.

Otro tipo de zona de riesgo muy frecuente se presenta cuando dos obstáculos están muy cerca uno del otro. Existe la posibilidad de que el algoritmo genere una trayectoria que pase por entre los obstáculos, sin tener en cuenta que las dimensiones del móvil puedan no permitir realizar esta trayectoria. Para evitar esta situación, se genera alrededor del obstáculo un potencial descendente tal como se muestra en la figura 16.

Figura 16. Campo de potencial alrededor de un obstáculo

49	49	49	49	49
49	50	50	50	49
49	50	200	50	49
49	50	50	50	49
49	49	49	49	49

El número de cuadros alrededor del obstáculo con potenciales grandes depende de cuanta zona de riesgo se quiera considerar.

Con valores numéricos alrededor del obstáculo se simula la fuerza repulsora que los caracteriza y puesto que la trayectoria transcurre de manera descendente, al encontrar valores grandes de potencial se desvía a zonas más seguras.

CREACIÓN DEL POTENCIAL EN LOS ORILLOS DE LA PISTA

Esta es una forma de asegurarse que el móvil jamás salga del área de trabajo y corresponde más a un método de seguridad que a una necesidad, ya que si el robot posee sensores de proximidad, estos pueden evitar las colisiones con las paredes del orillo de la pista.

Los bordes del área de trabajo están representados por un potencial grande, en este caso su valor corresponde con el número 50 y al igual que para un obstáculo, se genera una zona de riesgo para evitar que el móvil se acerque demasiado.

En la figura 17 se muestra el potencial alrededor de la pista, el potencial del punto de llegada y el de los obstáculos.

Figura 17. Vista general de la pista.

50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
50	49	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	13	49	50
50	49	13	100	12	12	12	12	12	12	12	12	12	12	12	12	12	12	49	50
50	49	13	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11	49	50
50	49	13	12	11	47	47	47	47	47	47	47	47	47	10	10	10	10	49	50
50	49	13	12	11	47	48	48	48	48	48	48	48	48	47	9	9	9	49	50
50	49	13	12	11	47	48	49	49	49	49	49	48	47	8	8	8	8	49	50
50	49	13	12	11	47	48	49	50	50	50	49	48	47	7	7	7	7	49	50
50	49	13	12	11	47	48	49	50	200	50	49	48	47	6	6	6	6	49	50
50	49	13	12	11	47	48	49	50	50	50	49	48	47	5	5	5	5	49	50
50	49	13	12	11	47	48	49	49	49	49	48	47	4	4	4	4	4	49	50
50	49	13	12	11	47	48	48	48	48	48	48	47	3	3	3	3	3	49	50
50	49	13	12	11	47	47	47	47	47	47	47	47	47	2	2	2	2	49	50
50	49	13	12	11	10	9	8	7	6	5	4	3	2	1	1	1	2	49	50
50	49	13	12	11	10	9	8	7	6	5	4	3	2	1	0	1	2	49	50
50	49	13	12	11	10	9	8	7	6	5	4	3	2	1	1	1	2	49	50
50	49	13	12	11	10	9	8	7	6	5	4	3	2	2	2	2	2	49	50
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50

RESULTADOS DEL ALGORITMO

A continuación se muestra algunos ejemplos de las trayectorias que se pueden obtener con el algoritmo de campos de potencial desarrollado. Estos algoritmos

son sólo pruebas al azar para demostrar la efectividad del algoritmo. Las pruebas mas exhaustivas se encuentran en el capítulo de pruebas y análisis de resultados, donde además se hace una comparación con el otro método implementado.

EJEMPLO 1:

Parámetros iniciales:

Posición inicial: $(x,y) = (5,5)$.

Posición Final: $(x,y) = (15,15)$

Número de obstáculos = 1

Posición del obstáculo: $(x,y) = (9,9)$

Al introducir estos datos en el algoritmo se obtiene la representación que se muestra en las figuras 18 y 19, los resultados obtenidos después de ejecutado la planeamiento se pueden observar en la figura 20.

Figura 18. Representación de los datos ingresados al algoritmo con un obstáculo fijo

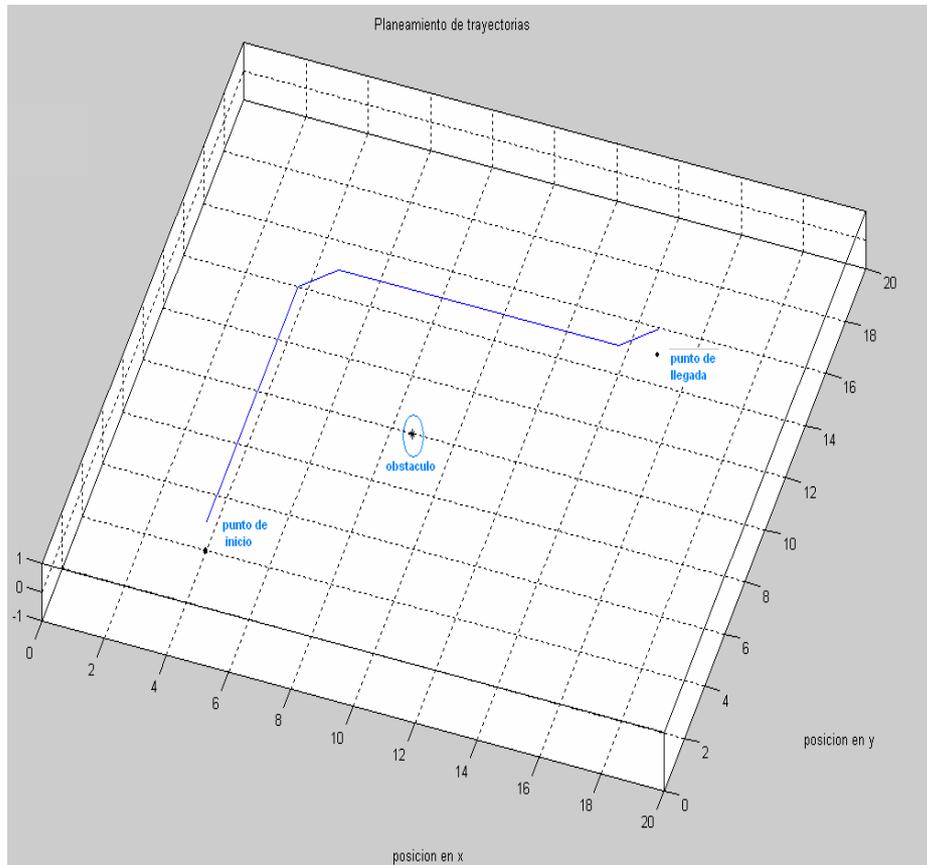
50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50	
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	
50	49	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	49	50	
50	49	12	11	11	11	11	11	11	11	11	11	11	11	11	11	11	49	50	
50	49	12	11	100	47	47	47	47	47	47	47	47	10	10	10	10	49	50	
50	49	12	11	47	48	48	48	48	48	48	48	47	9	9	9	9	49	50	
50	49	12	11	47	48	49	49	49	49	49	48	47	8	8	8	8	49	50	
50	49	12	11	47	48	49	50	50	50	49	48	47	7	7	7	7	49	50	
50	49	12	11	47	48	49	50	200	50	49	48	47	6	6	6	6	49	50	
50	49	12	11	47	48	49	50	50	50	49	48	47	5	5	5	5	49	50	
50	49	12	11	47	48	49	49	49	49	49	48	47	4	4	4	4	49	50	
50	49	12	11	47	48	48	48	48	48	48	48	47	3	3	3	3	49	50	
50	49	12	11	47	47	47	47	47	47	47	47	47	2	2	2	2	3	49	50
50	49	12	11	10	9	8	7	6	5	4	3	2	1	1	1	2	3	49	50
50	49	12	11	10	9	8	7	6	5	4	3	2	1	0	1	2	3	49	50
50	49	12	11	10	9	8	7	6	5	4	3	2	1	1	1	2	3	49	50
50	49	12	11	10	9	8	7	6	5	4	3	2	2	2	2	2	3	49	50
50	49	12	11	10	9	8	7	6	5	4	3	3	3	3	3	3	3	49	50
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	
50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50	

El punto de inicio se representa con el número "100", el punto de llegada esta definido como "0" y los obstáculos están representados con el número "200". Todos los movimientos correspondientes a la trayectoria encontrada son ejecutados dentro de la matriz y exportados para definir al móvil hacia donde moverse. La estructura definida para los movimientos es la que se observa en la figura 9.

Figura 19. Representación de los datos ingresados al algoritmo con un obstáculo fijo

50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50	
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	
50	49	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	49	50	
50	49	12	100	11	11	11	11	11	11	11	11	11	11	11	11	11	11	49	50	
50	49	12	100	100	47	47	47	47	47	47	47	47	10	10	10	10	10	49	50	
50	49	12	100	47	48	48	48	48	48	48	48	47	9	9	9	9	9	49	50	
50	49	12	100	47	48	49	49	49	49	49	49	48	47	8	8	8	8	8	49	50
50	49	12	100	47	48	49	50	50	50	49	48	47	7	7	7	7	7	49	50	
50	49	12	100	47	48	49	50	200	50	49	48	47	6	6	6	6	6	49	50	
50	49	12	100	47	48	49	50	50	50	49	48	47	5	5	5	5	5	49	50	
50	49	12	100	47	48	49	49	49	49	49	48	47	4	4	4	4	4	49	50	
50	49	12	100	47	48	48	48	48	48	48	48	47	3	3	3	3	3	49	50	
50	49	12	100	47	47	47	47	47	47	47	47	47	2	2	2	2	3	49	50	
50	49	12	11	100	100	100	100	100	100	100	100	100	100	1	1	2	3	49	50	
50	49	12	11	10	9	8	7	6	5	4	3	2	1	100	1	2	3	49	50	
50	49	12	11	10	9	8	7	6	5	4	3	2	1	1	1	2	3	49	50	
50	49	12	11	10	9	8	7	6	5	4	3	2	2	2	2	2	3	49	50	
50	49	12	11	10	9	8	7	6	5	4	3	3	3	3	3	3	3	49	50	
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	
50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50	

Figura 20. Resultados obtenidos luego de ejecutar el algoritmo.



Como se puede observar entre las graficas 18 y 19 lo único que se ha modificado en la matriz es el camino que en este caso se ha representado con el número “100” en las posiciones resultado del planeamiento.

Figura 21. Representación de los movimientos.

Diagonal inf. Izquierda	Abajo	Diagonal inf derecha
Atrás	Punto Actual	Adelante
Diagonal sup. izquierda	Arriba	Diagonal Sup izquierda

La estructura de la figura 21 esta definida de la forma como aumentan y disminuyen los índices de las matrices tal como puede apreciarse en la figura 22.

Figura 22. Definición de puntos vecinos dentro de una matriz

$A(x-1,y-1)$	$A(x-1,y)$	$A(x-1,y+1)$
$A(x,y-1)$	$A(x,y)$	$A(x,y+1)$
$A(x+1,y-1)$	$A(x+1,j)$	$A(x+1,y+1)$

Aquí el movimiento en filas corresponde a los movimientos en x , y los movimientos en columnas los movimientos en y .

El algoritmo funciona con cualquier número de obstáculos y en cualquier dirección. El número de obstáculos depende del tamaño del terreno, ya que un número elevado de obstáculos reduciría la posibilidad de tener espacios para generar la trayectoria y considerar zonas de riesgo.

EJEMPLO 2:

Parámetros iniciales:

Posición inicial: $(x,y) = (17,17)$.

Posición Final: $(x,y) = (4,4)$

Número de obstáculos = 3

Posición del obstáculo1: $(x,y) = (16,4)$

Posición del obstáculo2: $(x,y) = (4,16)$

Posición del obstáculo3: $(x,y) = (15,8)$

Al introducir estos datos en el algoritmo se obtiene la representación que se muestra en la figura 23, los resultados obtenidos después del algoritmo se pueden observar en las figuras 24 y 25.

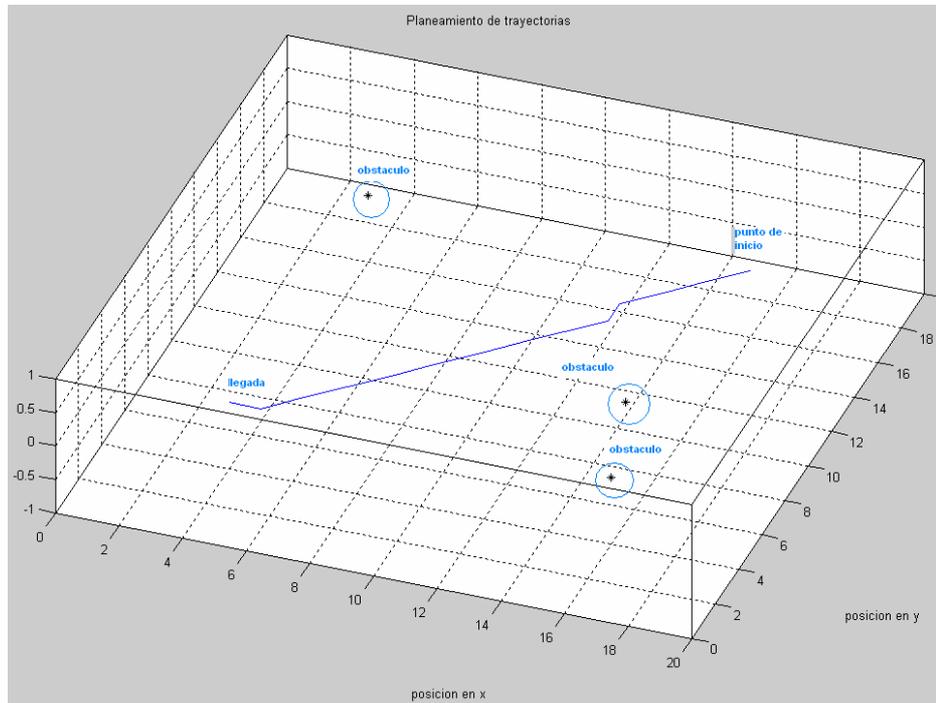
Figura 23. Distribución de los obstáculos, punto de inicio y punto de llegada.

50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
50	49	1	1	1	2	3	4	5	6	7	47	48	49	50	50	50	49	50
50	49	1	0	1	2	3	4	5	6	7	47	48	49	50	200	50	49	50
50	49	1	1	1	2	3	4	5	6	7	47	48	49	50	50	50	49	50
50	49	2	2	2	2	3	4	5	6	7	47	48	49	49	49	49	49	50
50	49	3	3	3	3	3	4	5	6	7	47	48	48	48	48	48	49	50
50	49	4	4	4	4	4	4	5	6	7	47	47	47	47	47	47	49	50
50	49	5	5	5	5	5	5	5	6	7	8	9	10	11	12	13	14	50
50	49	6	6	6	6	6	6	6	6	7	8	9	10	11	12	13	14	50
50	49	7	47	47	47	47	47	47	47	47	47	9	10	11	12	13	14	50
50	49	47	47	48	48	48	48	48	48	48	47	9	10	11	12	13	14	50
50	49	48	47	48	49	49	49	49	49	48	47	9	10	11	12	13	14	50
50	49	49	49	49	49	50	50	50	49	48	47	10	10	11	12	13	14	50
50	49	50	50	50	49	50	200	50	49	48	47	11	11	11	11	11	11	50
50	49	50	200	50	49	50	50	50	49	48	47	12	12	11	12	12	12	50
50	49	50	50	50	49	49	49	49	49	48	47	13	13	11	12	100	13	50
50	49	49	49	49	49	48	48	48	48	48	47	14	14	11	12	13	14	50
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50

Figura 24. Resultados obtenidos para la distribución definida en la figura 62.

50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
50	49	1	1	1	2	3	4	5	6	7	47	48	49	50	50	50	49	50
50	49	1	100	100	2	3	4	5	6	7	47	48	49	50	200	50	49	50
50	49	1	1	1	100	3	4	5	6	7	47	48	49	50	50	50	49	50
50	49	2	2	2	2	100	4	5	6	7	47	48	49	49	49	49	49	50
50	49	3	3	3	3	3	100	5	6	7	47	48	48	48	48	48	49	50
50	49	4	4	4	4	4	4	100	6	7	47	47	47	47	47	47	49	50
50	49	5	5	5	5	5	5	5	100	7	8	9	10	11	12	13	14	50
50	49	6	6	6	6	6	6	6	6	100	8	9	10	11	12	13	14	50
50	49	7	47	47	47	47	47	47	47	47	100	9	10	11	12	13	14	50
50	49	47	47	48	48	48	48	48	48	48	47	100	10	11	12	13	14	50
50	49	48	47	48	49	49	49	49	49	48	47	100	10	11	12	13	14	50
50	49	49	49	49	49	50	50	50	49	48	47	10	100	11	12	13	14	50
50	49	50	50	50	49	50	200	50	49	48	47	11	11	100	11	11	11	50
50	49	50	200	50	49	50	50	50	49	48	47	12	12	11	100	12	12	50
50	49	50	50	50	49	49	49	49	49	48	47	13	13	11	12	100	13	50
50	49	49	49	49	49	48	48	48	48	48	47	14	14	11	12	13	14	50
50	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49	49
50	49	50	50	50	50	50	50	50	50	50	50	50	50	50	50	50	49	50

Figura 25. Resultados del método de campos de potencial con 3 obstáculos



EJEMPLO 3:

En este caso se realizará la simulación cuando se tienen seis obstáculos.

Parámetros iniciales:

Posición inicial: $(x,y) = (3,3)$.

Posición Final: $(x,y) = (17,17)$

Número de obstáculos = 4

Posición del obstáculo1: $(x,y) = (9,9)$

Posición del obstáculo2: $(x,y) = (3,17)$

Posición del obstáculo3: $(x,y) = (17,3)$

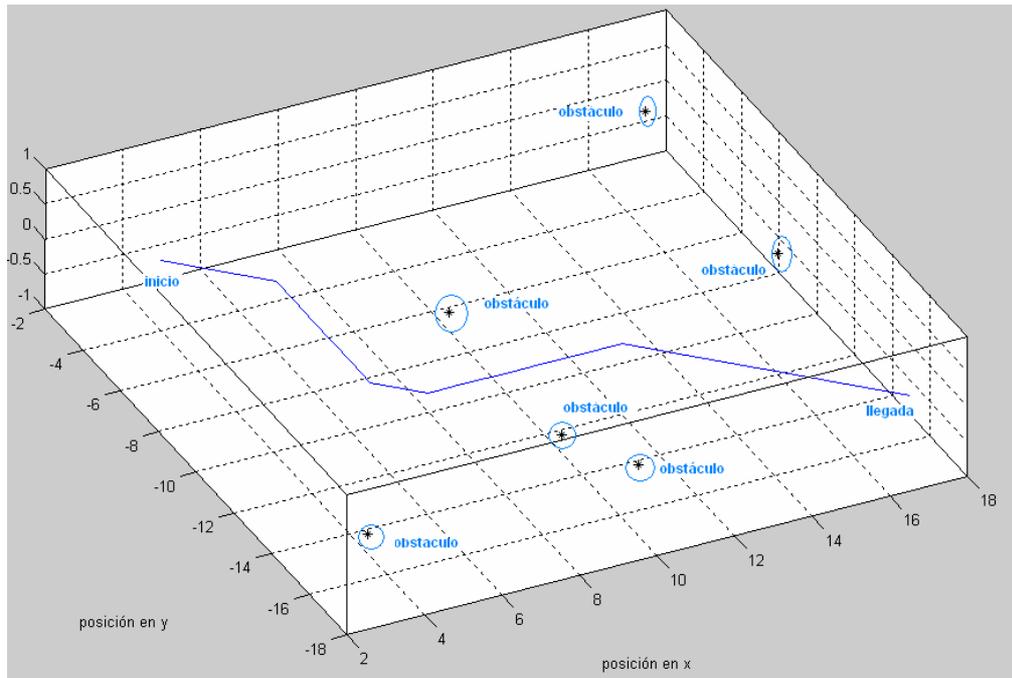
Posición del obstáculo4: $(x,y) = (9,1)$

Posición del obstáculo5: $(x,y) = (9,15)$

Posición del obstáculo6: $(x,y) = (10,17)$

En la figura 26 se puede apreciar el resultado del algoritmo cuando se tienen 6 obstáculos.

Figura 26. Resultados obtenidos cuando se tienen 6 obstáculos.



3.2.2 Algoritmos genéticos

La implementación de un algoritmo genético simple implica tener en cuenta tres operaciones características como reproducción, cruce y mutación.

Para realizar estas etapas el algoritmo debe ser codificado en cadenas que se llaman cromosomas, cada una conformada por elementos característicos denominados genes.

CODIFICACIÓN

Si se piensa en una trayectoria se debe pensar en como definirla en términos de sus componentes fundamentales (genes) para esto se trabaja en la codificación binaria que permite ordenar como cromosomas, cualidades como su dirección en

el plano cartesiano y la distancia que le toma ir de un punto a otro.

Los cromosomas escogidos representan: monotonía de la trayectoria en x y y : cromosoma α ; dirección: cromosoma β ; y distancia: cromosoma δ .

La estrategia de codificación de los genes para los cromosomas es la que se observa en la Tabla 1:

TABLA 1.CODIFICACIÓN DE CROMOSOMAS.

Cromo soma	Codificación	Característica que representa
α	0	Monótono en X (MX)
	1	Monótono en Y (MY)
β	00	Vertical para MX y horizontal para MY
	01	Diagonal superior para MX Diagonal izquierda para MY
	10	Horizontal para MX Vertical para MY
	11	Diagonal Inferior para MX Diagonal derecha para MY
δ	Según ecuación (1)	

El número de genes del cromosoma está dado por la expresión:

$$1 + \log_2 N = \text{GenesCromosoma} \delta \quad (11)$$

Donde N es el número de celdas que representen el territorio a cubrir por el robot móvil. Cada pareja dirección-distancia, representan pasos de distancia variable (parejas $\beta\delta$)

Cuando el cromosoma de dirección sea codificado en "00" el cromosoma δ puede ser positivo o negativo, es decir el primer gen de este cromosoma es 0 para positivo o 1 para negativo.

Si es positivo y el individuo es MX representará una vertical positiva hacia arriba seguida de una diagonal superior derecha, si es negativo representara una vertical hacia abajo seguida de una diagonal inferior derecha. Si es positivo y el individuo es MY representara una horizontal positiva hacia la derecha seguida de una diagonal inferior derecha, si es negativo representara una horizontal negativa hacia la izquierda seguida de una diagonal inferior izquierda. La figura 27 y 28 muestra la distribución de cada cromosoma.

Figura 27. Ejemplo de trayectoria codificada.

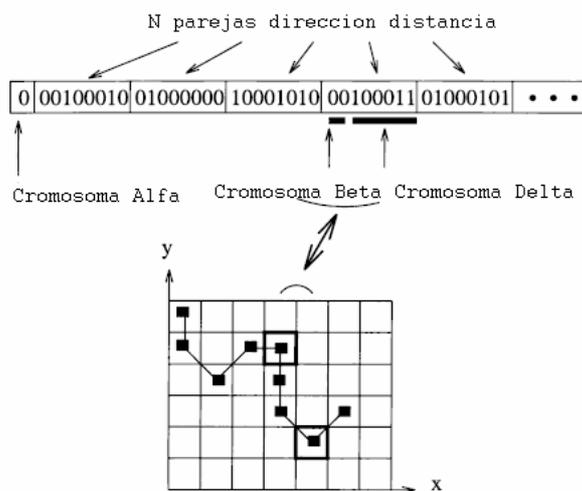
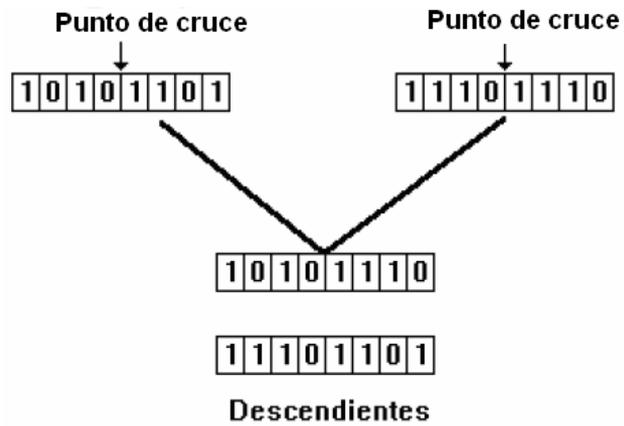


Figura 28. Método de Cruce



RESULTADOS OBTENIDOS

En las figuras 29, 30 y 31 se presenta una serie de trayectorias generadas por el Algoritmo Genético ante diferentes obstáculos en el medio.

Figura 29. Camino exitoso sin zonas de riesgo en los obstáculos, generación 20

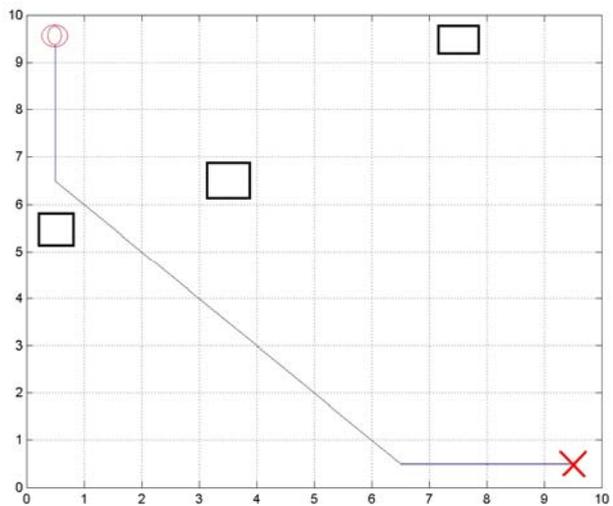


Figura 30. Camino exitoso sin zonas de riesgo, generación 110.

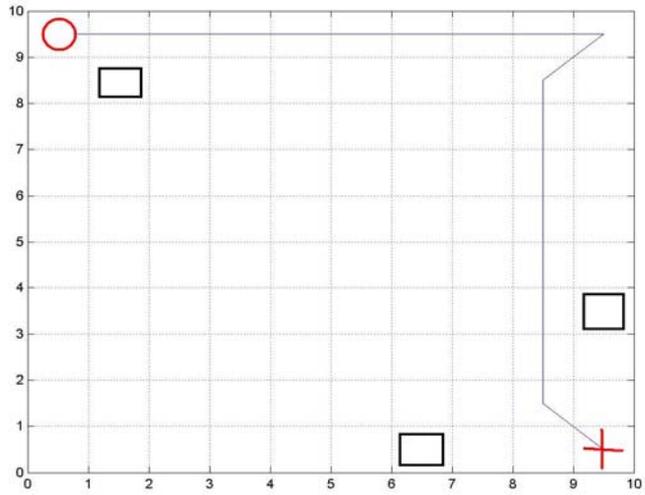
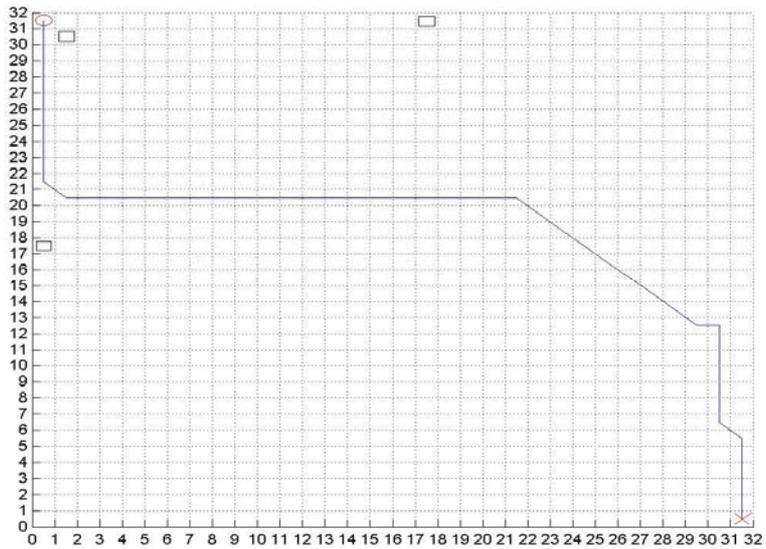


Figura 31. Camino exitoso con grilla de 32 celdas, considerando zonas de riesgo (generación 205).

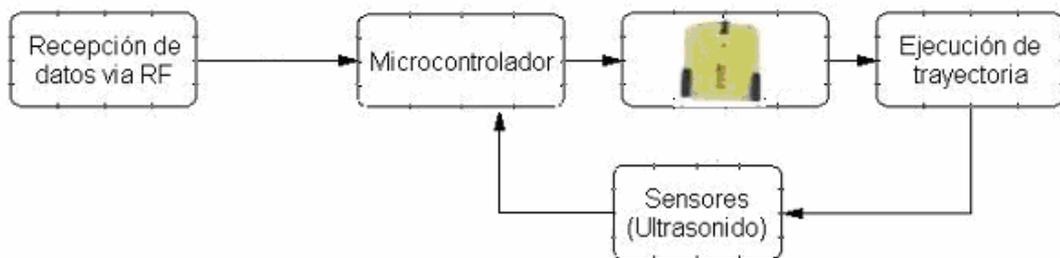


3.3 ALGORITMO DE CONTROL DE BAJO NIVEL

Una vez encontrada la trayectoria es necesario ejecutarla. Esto como se mencionó anteriormente se realizó en un microcontrolador Motorola, cuya selección se basó en la relación de señales que se trabajaron.

Un diagrama de bloques general del sistema se puede apreciar en la figura 32.

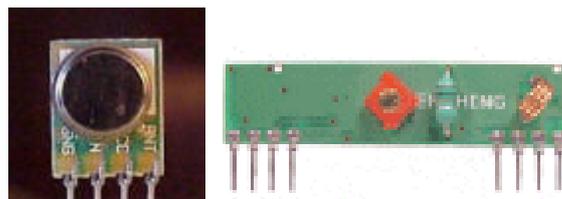
Figura 32. Diagrama de bloques del sistema de control de bajo nivel.



Este sistema solo recibe señales en un sentido vía radiofrecuencia, es decir solamente el microcontrolador recibe señales enviadas desde el host de control, y no le regresa información a este.

La transmisión de los datos se realiza usando un TSW433 y un RSW433 los cuales trabajan a una frecuencia de 433 Mhz y usan modulación AM para la transmisión de los datos. Estos módulos transmiten datos en forma serial a una velocidad de 1200 baudios. La figura 33 muestra los módulos usados.

Figura 33. Módulo de Radiofrecuencia



Tomado de suconel@epm.net.co

La configuración del microcontrolador y la asignación de pines se realizó basados en la tabla 2.

Tabla 2. Función y configuración de puertos.

Puerto	Configuración	Asignación
PTD4/T1CH0	Salida	Generación de interrupciones para las señales PWM
PTD5/T1CH1	Entrada	Captura de la señal ECO del Ultrasonido
PTB0-PTB7	Salida	Generación señal control Servomotores (max 8 servos)
PTE1/RxD	Entrada	Recepción de datos de la Trayectoria
PTA5-PTA6	Salida	Señal de excitación del Ultrasonido
PTA3	Entrada	Selección de algoritmo de Planeación
PTA4	Salida	Selección de canal del Multiplexor 2 a 1

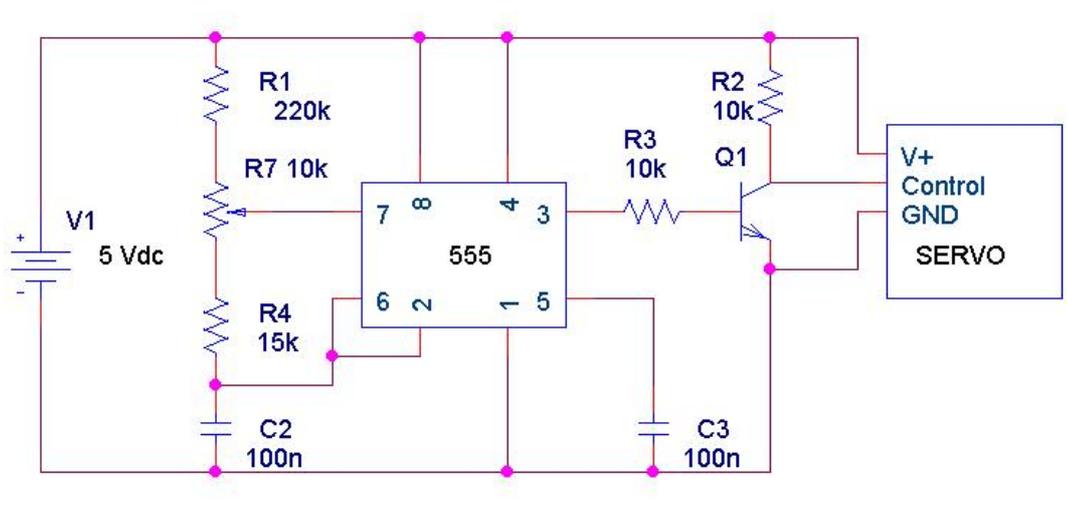
En el microcontrolador se realizó la programación para el algoritmo genético y campos de potencial. Los puertos usados son los mismos independientemente de la programación, lo que cambia es la codificación de las salidas dentro del programa como se explicará mas adelante.

El resultado del procesamiento de la información en el microcontrolador es enviado a los motores para ejecutar la acción propia de un robot tipo diferencial.

Para cada servomotor que se va a controlar, se realizó una prueba preliminar para encontrar exactamente el período y la duración de los pulsos que generan el mejor funcionamiento. Esto se hace con el fin de evitar inconvenientes a la hora de programar. Para realizar la prueba se utilizó el circuito mostrado en la figura 34. El

circuito consta de un monoestable (555) cuya frecuencia está determinada por el valor de los capacitores C2 y C3. La señal generada se invierte mediante un transistor conectado en configuración "colector común" que funciona en modo de saturación (esto significa APAGADO ó ENCENDIDO).

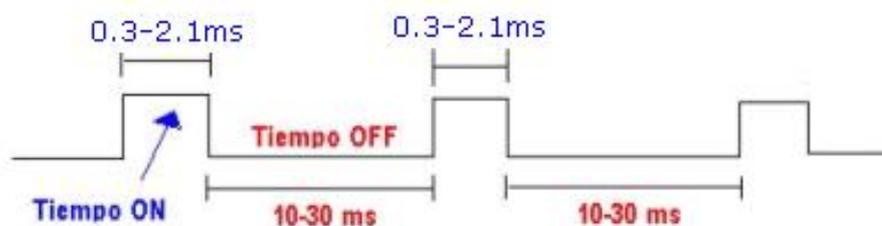
Figura 34. Circuito experimental con un 555



Al girar el potenciómetro (R7) se modifica el ancho del pulso de la señal de control lo cual hace girar el servomotor. De esta forma se busca el ancho de pulso necesario para ubicar el servo en cada uno de sus extremos.

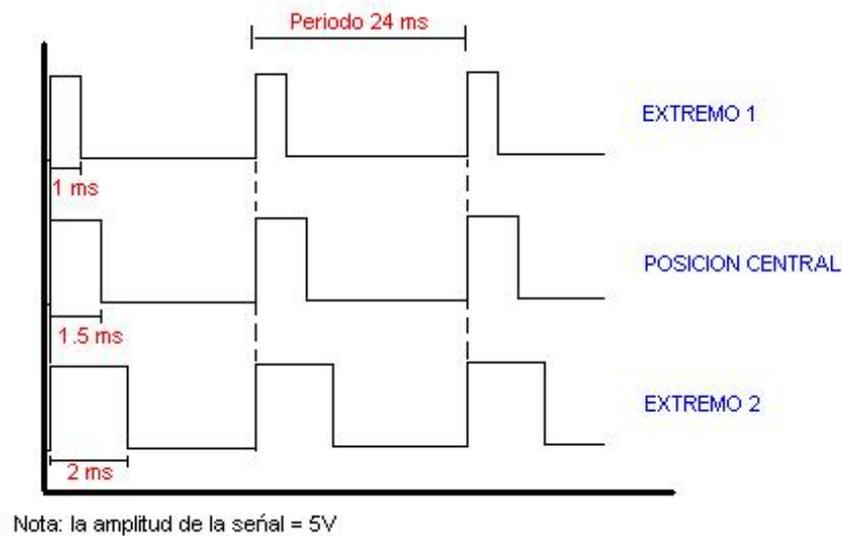
En la figura 35 se observa la señal de control que entrega el circuito experimental.

Figura 35. Tren de pulsos generado por el circuito.



Al realizar esta prueba al servomotor HITEC HS-422 se encontró que los límites de operación corresponden a un mínimo de 1ms y un máximo de 2ms de duración del pulso. Estos valores posicionan el servomotor en sus respectivos extremos. Un valor de 1,5ms de ancho de pulso ubica el servomotor en su posición central como se observa en la figura 36.

Figura 36. Señal de control para el servomotor.



Cuando se sobrepasan los límites de movimiento del servo, éste emite un zumbido indicando que se debe cambiar la longitud del pulso.

3.3.1 Control de servomotores

Las señales para el control de los servomotores son generadas por un puerto del microcontrolador. Para la generación de la señal PWM, se uso el *timer* y un comparador (interrupciones asociadas); de este modo el microcontrolador queda libre para realizar otras tareas, y la generación de la señal es automática y más efectiva. El mecanismo consiste en programar el *timer* con el ancho del pulso (el período de la señal) y al comparador con el valor de duración del pulso a nivel alto.

Cuando se produce una interrupción de *overflow* del *timer*, la subrutina de interrupción debe poner la señal PWM a nivel alto y cuando se produzca la interrupción del comparador, ésta debe poner la señal PWM a nivel bajo.

Los servomotores se modificaron a rotación continua como se explicó en el capítulo anterior. Se fijó una posición neutral correspondiente a un ancho de pulso de 1.5ms. Al aplicarle una señal con este ancho de pulso el servomotor permanece inmóvil. Para hacer girar el servomotor se le aplica una señal con ancho de pulso mayor o menor a 1.5ms, dependiendo del sentido de giro que se desee. Debido a que no hay realimentación porque se desacoplo el potenciómetro, el error no se minimiza y el servo continúa rotando.

En la tabla 3 se observan los puertos del microcontrolador que generan las señales de control para los servomotores.

Tabla 3. Asignación de puertos para los servomotores.

Puerto	Función
PTB2	Señal PWM para La llanta derecha
PTB3	Señal PWM para La llanta izquierda

3.3.2 Modulo de ultrasonido

El término ultrasonidos designa en principio a los sonidos de alta frecuencia, aunque en realidad corresponde a los sonidos que se encuentran justo por encima de la banda que es capaz de captar el oído humano. El intervalo de frecuencia que corresponde a los ultrasonidos va desde 20 Khz. hasta 500 KHz. Los sonidos a estas frecuencias son más direccionales que los sonidos de frecuencias bajas.

La determinación de la distancia se logra al enviar trenes de pulsos con periodos muy cortos y medir el tiempo transcurrido desde la emisión hasta la recepción del

eco. Este tiempo es proporcional a la distancia recorrida por las ondas de presión ultrasónicas.

Este sensor **permite** detectar objetos y calcular la distancia a la que se encuentra en un rango de 3 a 300 cm. Para su funcionamiento se debe excitar el emisor con una señal adecuada en amplitud y frecuencia de tal modo que este produzca una señal ultrasónica que se envía al medio externo por medio de un micrófono.

Este modulo contiene toda la electrónica necesaria para realizar el proceso de emisión, recepción y medición. Requiere cuatro conexiones para su funcionamiento, dos de ellas son la alimentación del circuito, (0 Volt y +5 Volt dc) y las dos restantes son las conexiones de las señales. El esquema se observa en la figura 37.

Figura 37. Conexiones del modulo del ultrasonido SRF04

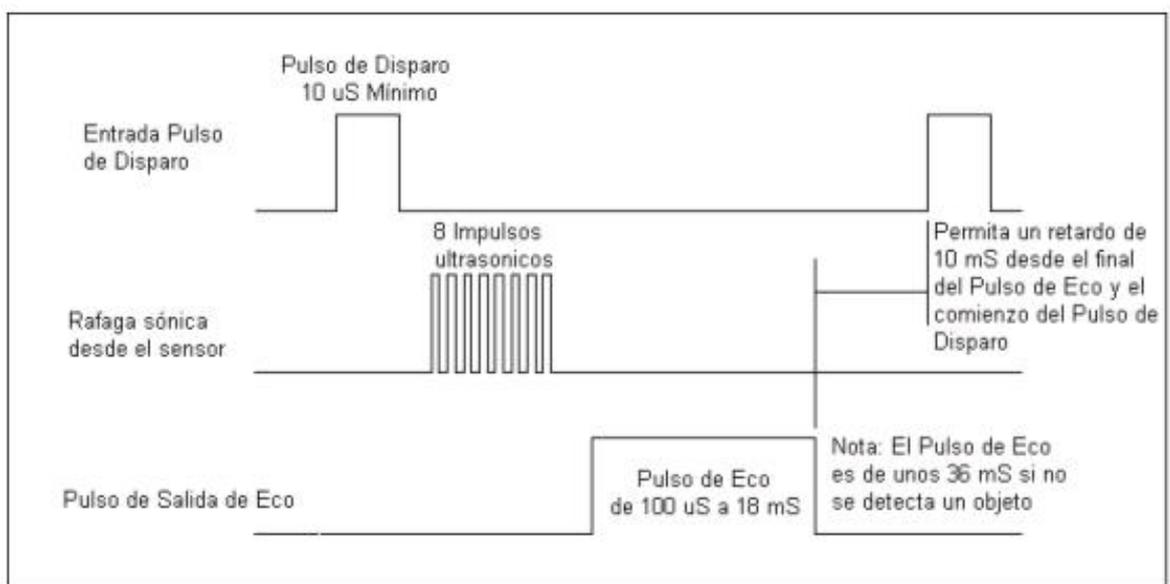


El sensor puede detectar objetos ubicados a una distancia entre tres y trescientos centímetros, por medio de un pulso de tensión cuyo ancho varia entre 100us para 3cm y 18ms para 300cm. Para distancias mayores el sensor envía un pulso de 36ms. Este pulso se obtiene como respuesta (eco) a un pulso de disparo de 10us,

enviada por el microcontrolador. Al terminar la recepción del pulso de eco se debe esperar 10ms antes de enviar el siguiente pulso de disparo.

Las señales de entrada de disparo y el pulso de salida de eco para este modulo se pueden observar en la figura 38.

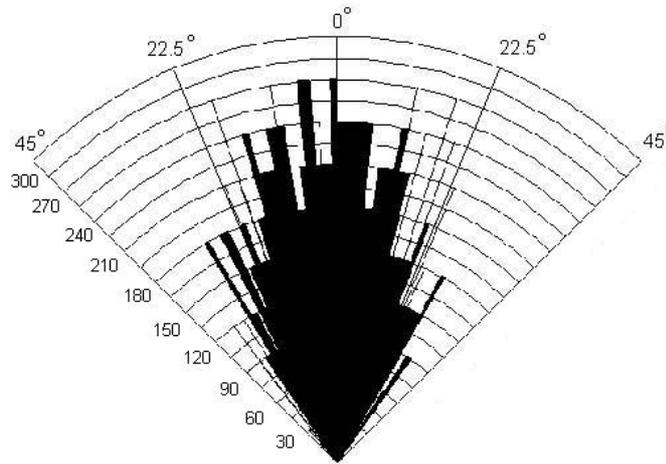
Figura 38. Diagrama de Tiempos del SRF04



El pulso de 100us como mínimo en la salida de eco se debe a que el circuito es desconectado durante este tiempo para evitar el ruido del pulso inicial.

Este modulo de ultrasonido tiene un Rango efectivo de aproximadamente 30° como se puede observar en la figura 39.

Figura 39. Rango efectivo del SRF04.



Tomada de SupeRobotica.com

CONTROL DEL ULTRASONIDO

Para el control del módulo de ultrasonido se debe generar una señal de excitación y capturar el pulso de eco que genera el modulo. La señal de excitación es generada por un pin del puerto A mediante la interrupción de “overflow” generada por el *timer*.

La captura del pulso de eco se realiza configurando un canal del *timer del microcontrolador* (PTD5/T1CH1) como entrada “input capture”. Primero se detecta el flanco ascendente configurando la función input capture en modo “Rising edge” y se almacena este valor como T1; luego es configurando en modo “Falling edge” para detectar el flanco descendente, almacenando este valor como T2. La captura de los flancos se observa en la figura 40. La diferencia entre T2 y T1 es el ancho del pulso capturado.

Figura 40. Captura del pulso de eco.



La configuración de los puertos que controlan el modulo de ultrasonido se observa en la tabla 4.

Tabla 4. Asignación de puertos para el modulo de ultrasonido.

Puerto	Función
PTD5/T1CH1	Captura del pulso De ECO
PTA5	Señal de excitación

3.3.3 Trayectoria planeada

El planeamiento de caminos es realizado en el PC utilizando algoritmos genéticos o campos de potencial y luego se envía al microcontrolador vía radio frecuencia en una cadena de bytes. Para cada algoritmo de planeación se realizo un código independiente. La selección del algoritmo se realiza mediante hardware, con un interruptor colocando el pin PTA3 a 1 (5 volts) para seleccionar el Algoritmo Genético y a 0 (0 volts) para Campos de potencial. Cada uno de ellos se explica a continuación.

ALGORITMO GENÉTICO (AG)

En esta estrategia cada byte recibido por el microcontrolador representa un segmento de la trayectoria y contiene la información descrita en la figura 41.

Figura 41. División del Byte para Algoritmo Genético

1 BYTE			
Monotonía (M)	Dirección	Signo	Distancia
1 Bit	2 Bits	1 Bit	4 Bits

El primer bit M indica que un camino es monótono en X cuando M es igual a cero y monótono en Y cuando M es igual a uno.

La descripción del byte se muestra en la tabla 5.

Tabla 5. Acción del móvil para AG.

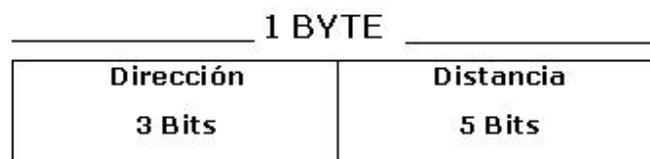
Byte	Descripción
0000XXXX	Vertical superior + diagonal superior derecha
0001XXXX	Vertical inferior + diagonal inferior derecha
001XXXXX	Diagonal superior derecha
010XXXXX	Horizontal derecha
011XXXXX	Diagonal inferior derecha
1000XXXX	Horizontal derecha + Diagonal inferior derecha
1001XXXX	Horizontal izquierda + Diagonal inferior izquierda
101XXXXX	Diagonal inferior izquierda
110XXXXX	Vertical inferior
111XXXXX	Diagonal inferior derecha

Los 4 bits de distancia representan el número de pasos que el móvil debe ejecutar en dicha dirección, donde la longitud del paso está predefinida según el número de divisiones utilizadas en la plataforma de trabajo (celdas), para realizar el planeamiento del camino.

CAMPOS DE POTENCIAL (CP)

El camino planeado con este método está descrito en una cadena de bytes, donde cada uno representa un segmento del camino planeado con la información de dirección, los 3 bits más significativos y distancia con los bits restantes, como se muestra en la figura 42.

Figura 42. División del Byte para la programación con Campos de Potencial



La descripción del byte se observa en la tabla 6.

Tabla 6. Acción del móvil para CP.

Byte	Descripción
000XXXXX	Horizontal derecha
001XXXXX	Diagonal inferior derecha
010XXXXX	Vertical inferior
011XXXXX	Diagonal inferior izquierda
100XXXXX	Horizontal izquierda
101XXXXX	Diagonal superior izquierda
110XXXXX	Vertical superior
111XXXXX	Diagonal superior derecha

Los 5 bits de distancia representan el número de pasos que el móvil debe ejecutar en la dirección especificada, donde la longitud del paso está predefinida según el número de divisiones utilizadas en la plataforma de trabajo (celdas), para realizar el planeamiento del camino.

3.3.4 Programación

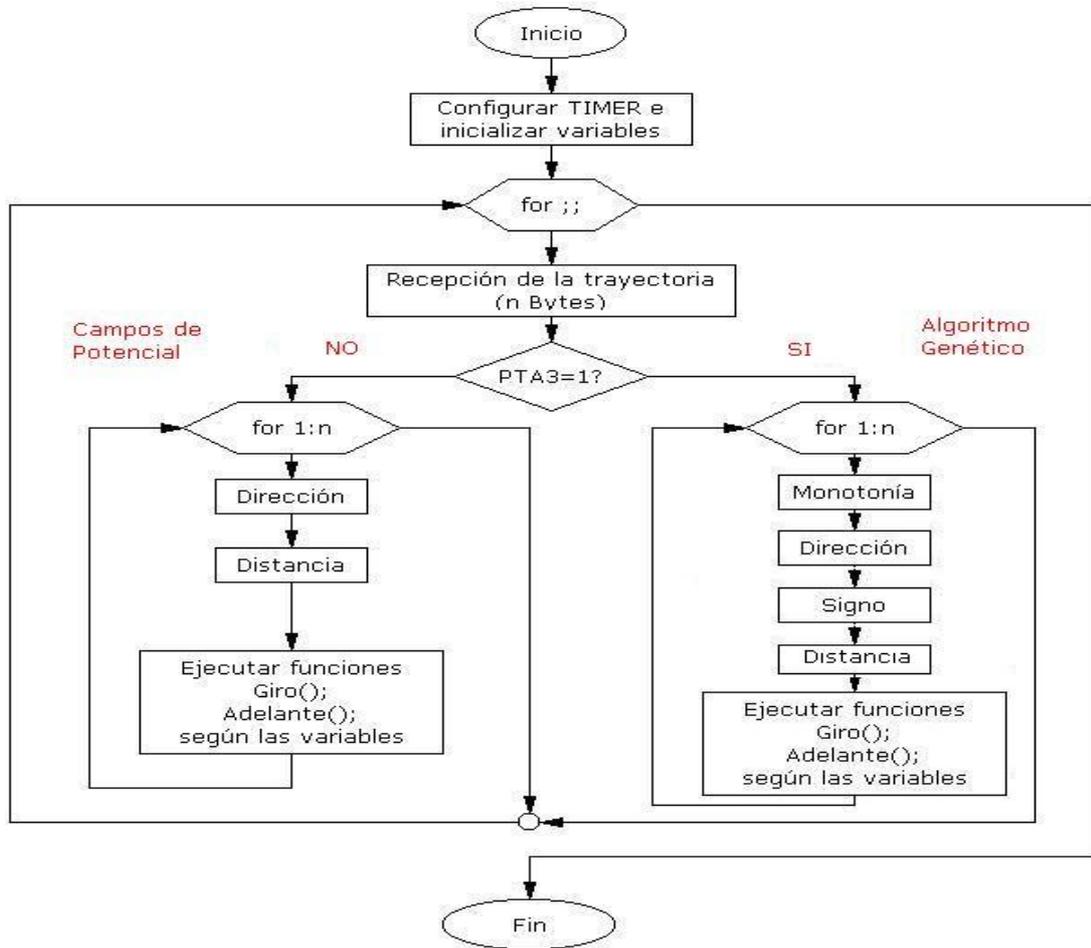
Para cada móvil se realizó un programa de control. El “*móvil controlado*” tiene como objetivo recibir las instrucciones enviadas desde el PC y planear la trayectoria descrita en cada instrucción. El objetivo del “*obstáculo móvil*” es desplazarse por toda la pista evitando obstáculos y de esta forma generar el ambiente dinámico. A continuación se da una explicación del programa de control para cada móvil y los subprogramas utilizados.

PROGRAMACIÓN DEL MÓVIL CONTROLADO

El programa tiene como objetivo recibir las instrucciones desde el PC y planear la trayectoria descrita en cada instrucción. El móvil tiene un módulo de ultrasonido ubicado en la parte frontal. Su función es la de evitar colisiones al ejecutar la trayectoria.

El diagrama de flujo para el móvil controlado se muestra en la figura 43.

Figura 43. Diagrama de flujo del móvil controlado.

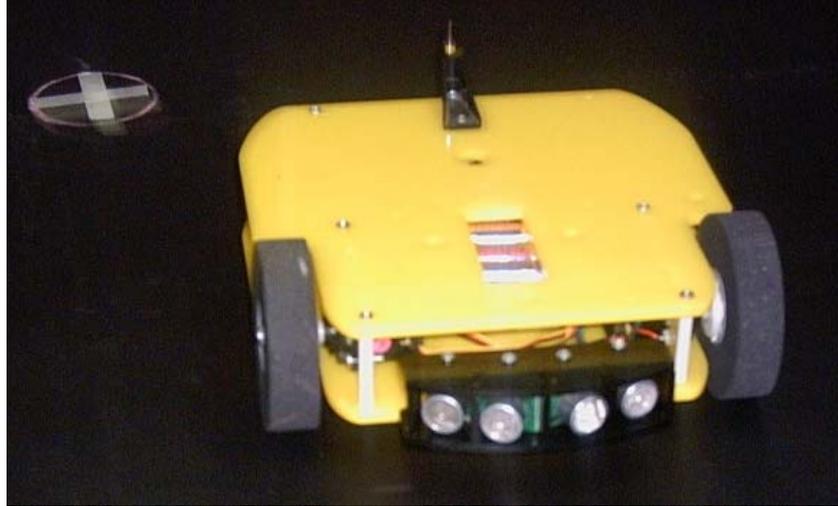


En el diagrama de flujo, **n** representa el número de Bytes de la trayectoria recibida y se observa la extracción de la información de cada uno de ellos según el algoritmo seleccionado.

PROGRAMACIÓN DEL OBSTÁCULO MOVIL.

El objetivo del obstáculo móvil es desplazarse por toda el área de la pista. El algoritmo esta diseñado para que el móvil este siempre en movimiento. En la figura 44 se puede observar el obstáculo móvil.

Figura 44. Obstáculo Móvil

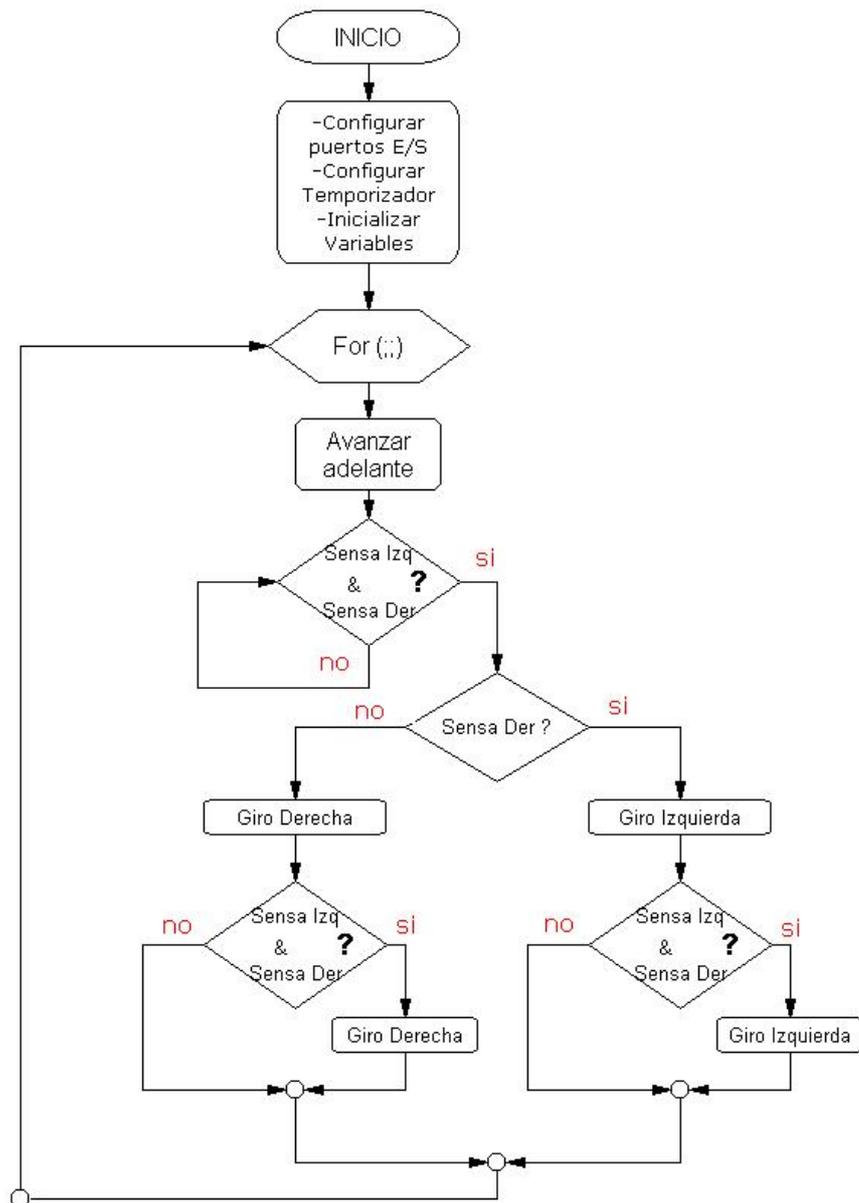


Sin la presencia de obstáculos el móvil se desplaza siempre hacia delante, analizando su entorno mediante los módulos de ultrasonido. El análisis del entorno se realiza de forma alterna entre los dos módulos de ultrasonido, esto se debe a que se tiene disponible un solo canal del timer del microcontrolador para la captura del eco.

Para realizar la captura de forma alterna se utilizó un multiplexor 2 a 1, el cual es controlado por uno de los puertos de Entrada/salida (PTA4) que posee el microcontrolador. Al colocar PTA4 en uno, se habilita la captura del modulo derecho y en cero la del modulo izquierdo. Si un obstáculo es sentido por el modulo derecho el móvil gira 45 grados hacia la izquierda intentando esquivarlo. Si es sentido por el modulo izquierdo el móvil gira 45 grados hacia la derecha. Si una vez terminado el giro se sigue detectando el obstáculo, el móvil continúa girando en el mismo sentido.

El diagrama de flujo del algoritmo se observa en la figura 45.

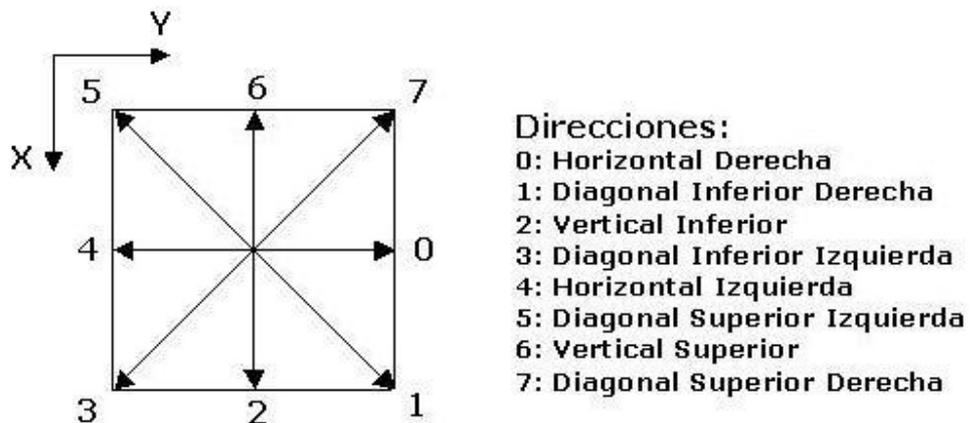
Figura 45. Diagrama de Flujo del móvil aleatorio



SUBPROGRAMA DE GIRO.

Este subprograma, definido como una función posee de entrada un número entero que denota la dirección hacia la cual se debe ubicar el móvil. Las direcciones que puede tomar el móvil se observan en la figura 46. El plano XY en la figura, se tomo de acuerdo con la ubicación de los elementos en una matriz, donde X representa las filas y Y las columnas.

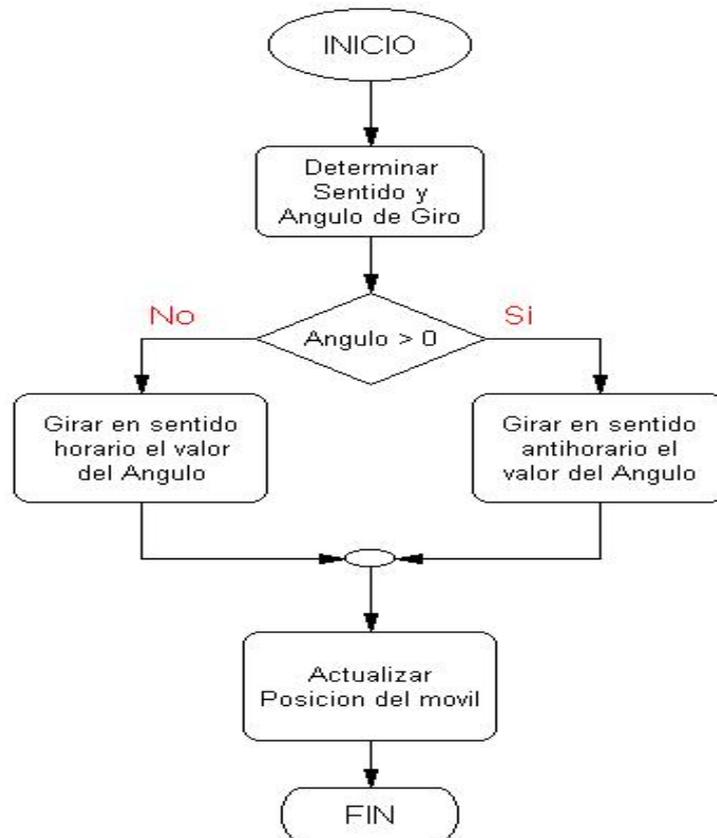
Figura 46. Representación de las direcciones del móvil.



Luego es determinado el sentido de giro si es horario o antihorario para llegar a la dirección deseada garantizando un giro menor o igual a 180 grados. Para conseguir esto se debe conocer la dirección en la cual está el móvil y a la cual se desea ubicar. Como cada dirección esta denotada por un número, con la diferencia de estas posiciones (Dirección Deseada – Dirección actual) se determina el número de posiciones que debe recorrer y con el signo de la diferencia se determina el sentido del giro.

En la figura 47 se observa el diagrama de flujo para el subprograma de giro.

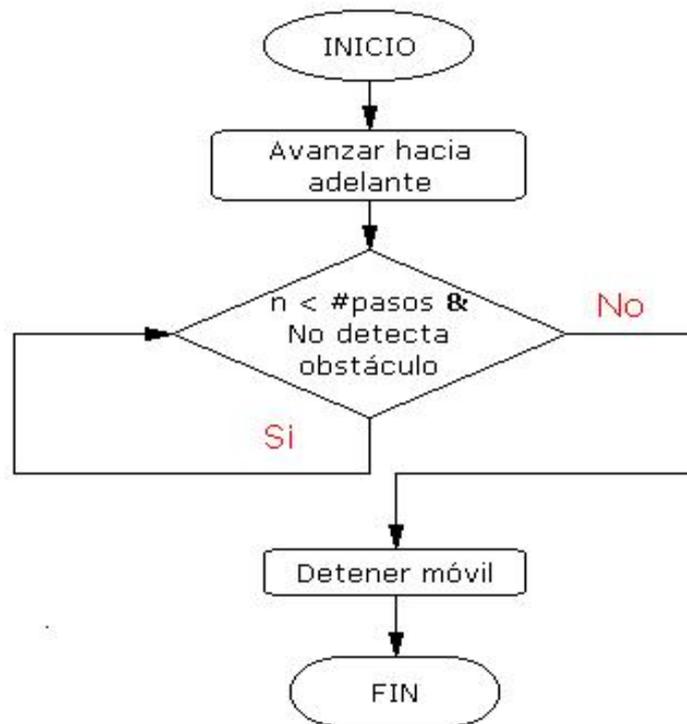
Figura 47. Diagrama de Subprograma de Giro



SUBPROGRAMA DE AVANZAR.

Este subprograma tiene como entrada un número entero el cual representa el número de pasos que avanzará el móvil. La longitud del paso esta predefinida según las divisiones de la pista que tiene en cuenta el algoritmo de planeación. El diagrama seguido para esta subrutina se puede verificar en la figura 48.

Figura 48. Diagrama de subprograma Avanzar.



El móvil controlado se desplaza hacia delante hasta alcanzar el número de pasos mientras que no se detecte ningún obstáculo, en caso contrario el móvil se detiene y sale del subprograma.

4. VISIÓN DINÁMICA

4.1 GENERALIDADES DEL ANÁLISIS DINÁMICO DE IMÁGENES

4.1.1 Definición

El análisis del movimiento o análisis dinámico de imágenes es una aplicación de la visión computacional que permite realizar el seguimiento y localización de los objetos en movimiento presentes en una secuencia de imágenes.

Al ser parte de la visión computacional, el análisis dinámico de imágenes se realiza siguiendo las mismas etapas de la visión computacional, en las que aplicando procesamiento digital de imágenes se extrae la información necesaria para cumplir con la tarea de localización y seguimiento.

4.1.2 Etapas de un Sistema de Visión Computacional

A. Adquisición de Imágenes.

En esta etapa se obtienen las imágenes visuales. Dos elementos son necesarios para adquirir imágenes digitales de video, el primero es un dispositivo físico que sea sensible a la banda visible del espectro electromagnético y que produzca una señal eléctrica proporcional al nivel de energía percibido, y el segundo es un digitalizador capaz de convertir la señal eléctrica del dispositivo en una forma digital, aunque en la actualidad existen cámaras que entregan la señal digital directamente al computador haciendo así innecesario el digitalizador.

B. Pre-procesamiento.

El objetivo principal del pre-procesamiento es una mejora de los datos de la imagen suprimiendo los efectos de ruido e incrementando las características deseadas para un exitoso procesamiento.

C. Segmentación.

En esta etapa los objetos de interés en la imagen son separados del fondo para su posterior estudio.

D. Representación y Descripción.

La representación es el proceso mediante el cual se obtienen características convenientes para representar un objeto segmentado en la escena.

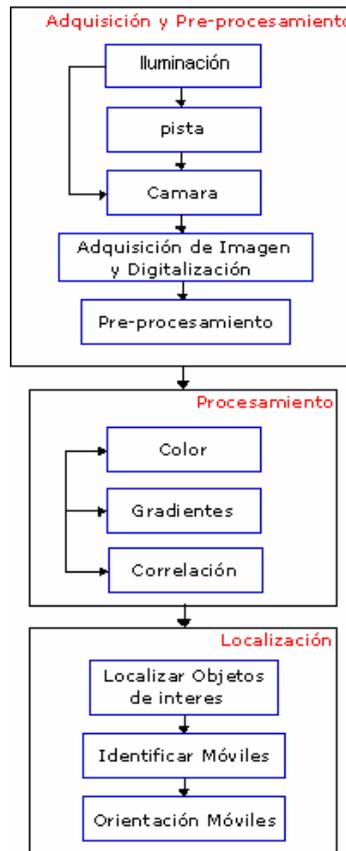
E. Reconocimiento e Interpretación.

Con base en la información suministrada por una serie de descriptores utilizados, el reconocimiento asigna una etiqueta a cada elemento. La interpretación se encarga de asignar un significado a un conjunto de objetos reconocidos.

4.1.3 Etapas fundamentales de la aplicación implementada

Las etapas implementadas en este trabajo, para la localización dinámica de objetos en movimiento se observan en la figura 49. Las etapas de reconocimiento e interpretación y de representación y descripción se encuentran fusionadas en una sola llamada localización. Las demás etapas conservan una estructura semejante a la descrita anteriormente, para un sistema de visión computacional.

Figura 49. Diagrama de bloques de la aplicación implementada



4.2 GENERALIDADES

4.2.1 Descripción de la escena

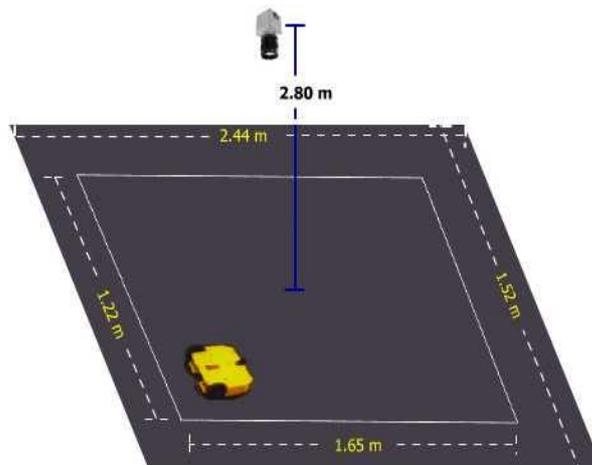
El escenario en general consta de una pista de madera, con un área de 3,7088 m². El color de la pista es negro mate con el objetivo de mejorar los resultados en el proceso de segmentación de los elementos en movimiento y posibles problemas de reflexión de la luz por el uso de pinturas brillantes.

La cámara se ubicó a una altura de 2,80 m de altura de la pista, procurando que su eje óptico quedara perpendicular al piso, de esta forma se obtiene un área efectiva de captura de 2,013 m² cuando se usa un lente de 8mm. Debido a que el área cubierta con el lente de 8mm es muy pequeña y la dimensión de cada uno de

los carros es considerable, las restricciones de movimiento no permiten realizar un buen planeamiento, por lo cual se instaló en la cámara un lente de 4 mm y se amplió el área de captura efectiva tomando toda la pista, es decir 3.708 m². En cuanto a la iluminación se limitó a las fuentes con las que se cuenta en el laboratorio. Estas fuentes son dos lámparas de techo fluorescentes de 40 W.

Un esquema de distribución de los principales elementos y algunas dimensiones de importancia en el escenario se observan en la figura 50.

Figura 50. Distribución general de la escena



4.2.2 Elementos de adquisición

Los dos elementos fundamentales en el sistema de adquisición son: la cámara y la tarjeta digitalizadora.

Cámara.

Para la obtención de la señal que se va a digitalizar se hizo uso de una cámara análoga DFK50H13/N, que posee un lente de 8mm y salida de video compuesto. Fue necesario realizar la calibración para evitar errores en los resultados.

Luego se optó por realizar nuevamente la calibración de la cámara usando un lente de 4mm debido a que con este no se logró tomar el área de trabajo que se necesitaba.

Para la calibración realizada se utilizaron 24 imágenes y se usó la toolbox de calibración de Matlab llamada calib. Con ella se logró obtener los resultados que se aprecian en las figuras 51 a 54. Tal como se podrá observar en la figura 55 donde se muestra el modelo de distorsión radial y tangencial existe un alto grado de aberración, teniéndose en las esquinas hasta 60 píxeles de desviación radial.

Figura 51. Foto tomada sin calibración

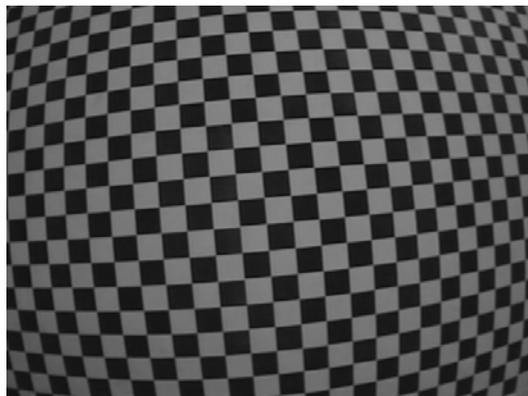


Figura 52. Foto obtenida después de la calibración

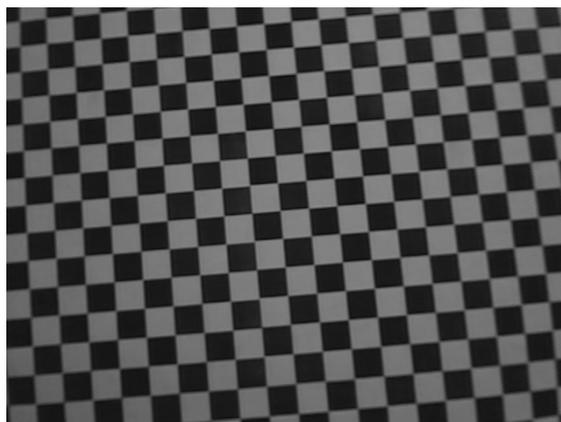


Figura 53. Foto tomada sin calibración

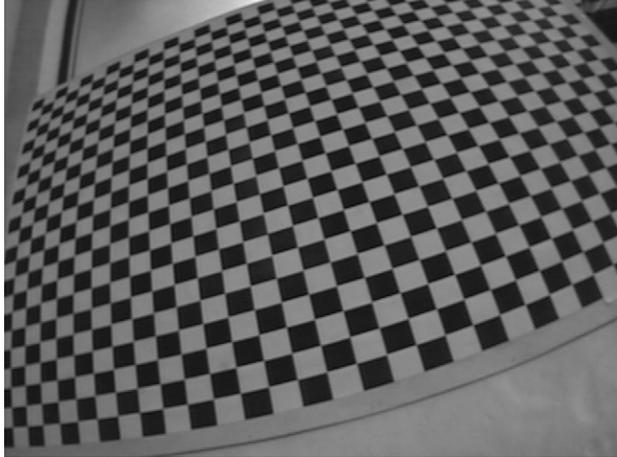
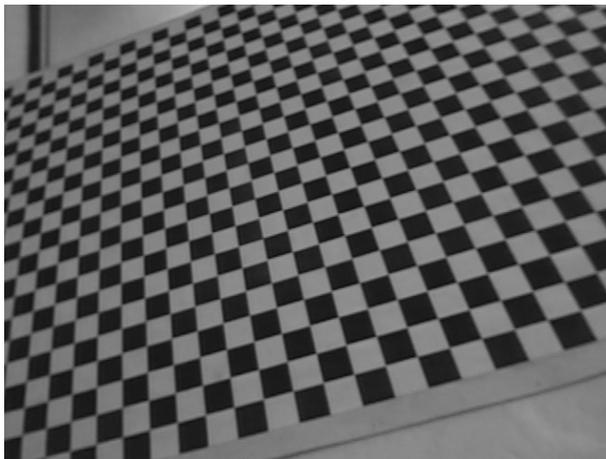


Figura 54. Foto obtenida después de la calibración



Focal Length: $fc = [600.47492 \ 604.09831] \pm [9.86222 \ 10.04268]$
Principal point: $cc = [282.20427 \ 223.68173] \pm [6.90267 \ 7.08700]$
Skew: $\alpha_c = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes =
 90.00000 ± 0.00000 degrees
Distortion: $kc = [-0.54004 \ 0.34794 \ 0.00396 \ 0.00033 \ 0.00000] \pm [$
 $0.01790 \ 0.03443 \ 0.00279 \ 0.00235 \ 0.00000]$
Pixel error: $err = [1.38326 \ 0.78623]$

Tarjeta digitalizadora

Para realizar la digitalización de las señales provenientes de la Cámara, se utilizó una tarjeta de compresión de video PCI de 32 bits DFG/Compress con una entrada / salida de video compuesto y un conector para obtener la señal de 8 cámaras diferentes. En la aplicación solamente se uso una cámara.

4.2.3 Algoritmo de adquisición y configuración de la tarjeta digitalizadora

El primer paso para realizar una correcta adquisición consiste en inicializar el hardware, ya que no se puede utilizar alguna otra función sin llevar a cabo este paso. En caso que la tarjeta de adquisición no se haya inicializado correctamente se desplegará un mensaje de error y finalizará el proceso. Si la tarjeta ya ha sido inicializada, estos pasos no se ejecutarán. Un esquema general se puede apreciar en la figura 56.

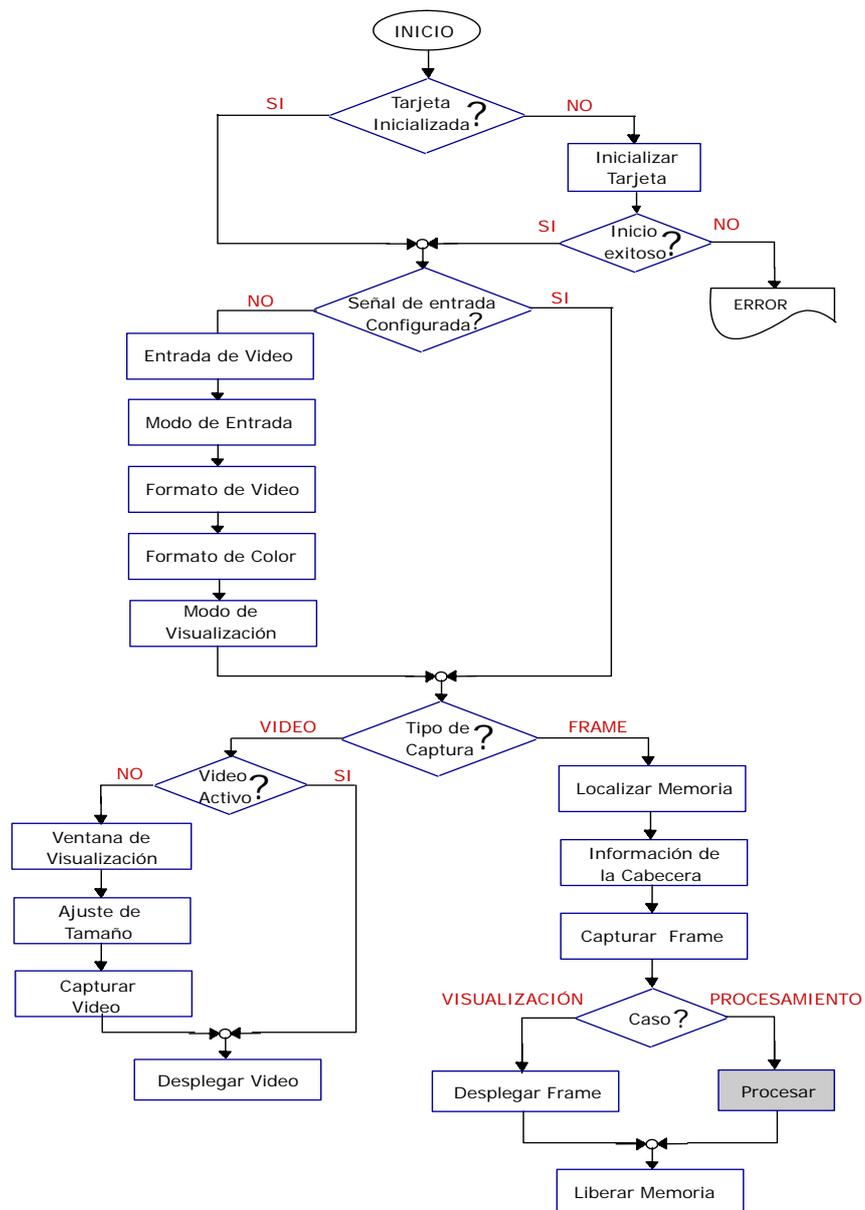
Una vez inicializado el hardware, se fijan algunos parámetros de la señal de video de entrada como son:

- a. canal de entrada: la tarjeta soporta cuatro canales de entrada de video.
- b. modo de la señal: video compuesto o S-VHS (Y/C).
- c. formato de video: PAL o NTSC.

d. formato de color: RGB, YUV. El proceso de adquisición de este proyecto captura imágenes RGB en verdadero color de 24 bits.

e. modo de visualización

Figura 56. Diagrama de flujo para algoritmo de adquisición

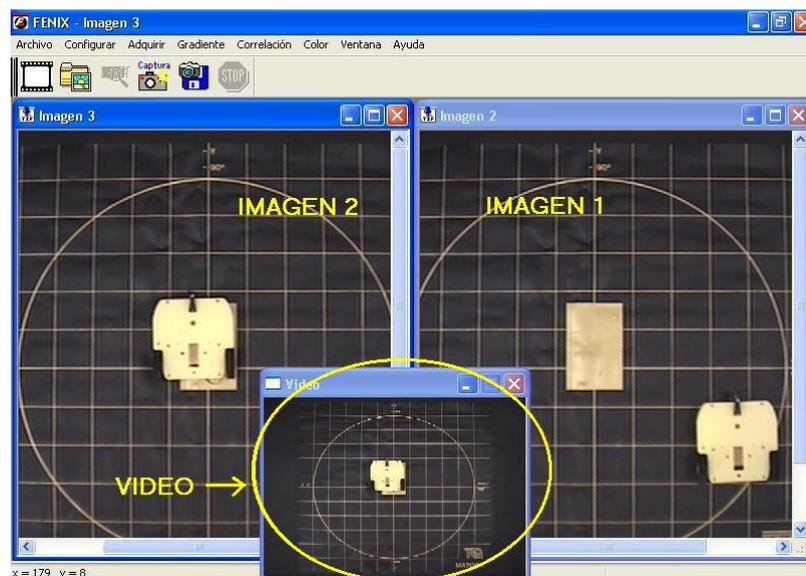


Para la captura del video, se implementó una interfase en la cual presenta la señal de video en una ventana de diálogo de tamaño ajustable. A partir de esta señal, se puede capturar cuadros que permanecen estáticos en un documento separado. Una imagen capturada de esta manera se conoce como un Frame.

Para capturar un frame, primero se selecciona un área en memoria donde almacenar los datos de la imagen a adquirir, también se ajusta la cabecera que corresponde a propiedades de la imagen como las dimensiones y el número de bits por píxel. Luego, si lo que se desea son imágenes independientes del proceso de detección y segmentación de objetos en movimiento se efectúa la captura y posterior despliegue, de lo contrario los frames adquiridos no serán desplegados, lo cual permite optimizar recursos computacionales y disminuir tiempo en el proceso.

Finalmente se libera la memoria ocupada, de tal manera que al capturar los frames siguientes no aumente el tamaño ocupado en memoria. De esta manera, el manejo de memoria tendrá el mismo grado de eficiencia en cuanto al numero de bytes usados por la aplicación, sin depender del tiempo que dure el proceso.

Figura 57. Adquisición de video y frames



En la figura 57 se aprecia el resultado de implementar el algoritmo de adquisición descrito anteriormente en visual C++ 6.0, el cual permite capturar video y desplegarlo en un único dialogo y además capturar múltiples imágenes en diferentes vistas.

4.3 PRE-PROCESAMIENTO

Una de los objetivos fundamentales del pre-procesamiento en la aplicación desarrollada en el proyecto es eliminar el ruido presente en la imágenes adquiridas, que aunque no es muy notorio si puede afectar en forma drástica los resultados obtenidos en el proceso de detección de móviles. Para solucionar este problema se opto por utilizar filtros de mediana [pajares], los cuales eliminan en gran parte el ruido presente en la imagen sin alterar de forma notoria la calidad visual de la misma.

4.4 DETECCIÓN Y SEGMENTACION DE MOVIMIENTO

Los algoritmos implementados para el efecto, son de amplio uso en aplicaciones de análisis dinámico de imágenes como es el caso del algoritmo de Jain utilizada en sistemas de vigilancia vial el cual se basa en la detección de bordes de los vehículos en movimiento en una secuencia de imágenes o algoritmos de reconocimiento de color utilizado en robótica de tipo recreativa como es el caso de la Copa de fútbol robótico.

Los diferentes algoritmos usados para la aplicación son los siguientes:

4.4.1 Método basado en Color

Los algoritmos de detección y segmentación de movimiento basados en el reconocimiento del color, distinguen el color de los objetos de interés presentes en la escena para así realizar su segmentación.

En este trabajo se implementaron dos modelos de color [12] el xyY y HLS para realizar la detección. La principal razón para elegir estos modelos radica en que la información cromática y la de brillo es separada en componentes distintas, lo cual hace a estos modelos robustos ante los cambios de iluminación que se presenten en el escenario. Además, al estar realizando detección solamente en las componentes que contienen la información cromática, el tiempo en la segmentación disminuye, lo cual disminuye también el tiempo total de procesamiento.

Modelos xyY y HLS

- **Modelo De Color xyY** : en 1931 la *Commission Internationale de l'Éclairage (CIE)*² define tres colores imaginarios base x, y e Y, como una solución al problema que tiene el modelo RGB³ de no poder representar muchos de los colores percibidos por el ojo humano.

El modelo xyY, esta basado en las curvas de respuesta espectral de los tres receptores de color del ojo (observador patrón) las cuales difieren ligeramente de una persona a otra [8]. El CIE ha definido un “observador estándar” basándose en la respuesta espectral media de la población.

Los colores primarios para el observador estándar se encuentran representados en un modelo espacial llamado *diagrama cromático CIE 1931 para un observador de 2°*, este diagrama se observa en la figura 58, además en la figura 59 se observa en qué consiste el concepto de un observador de 2°.

² Traducción Francesa de “Comisión Internacional sobre iluminantes”, principal organización internacional encargada de los problemas sobre medición del color.

³ Siglas en ingles de Red(rojo), Green(verde) y Blue (azul) colores base en el modelo RGB

Figura 58. Diagrama cromático CIE 1931 para un observador de 2°

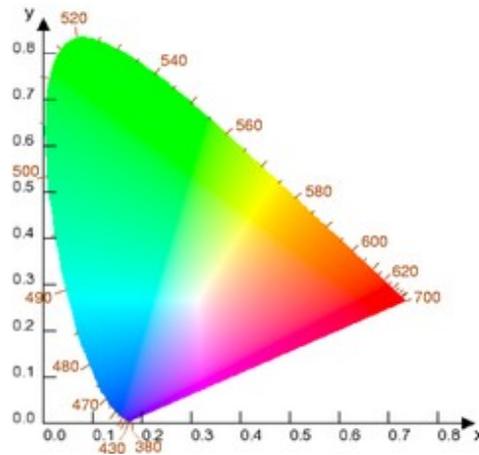
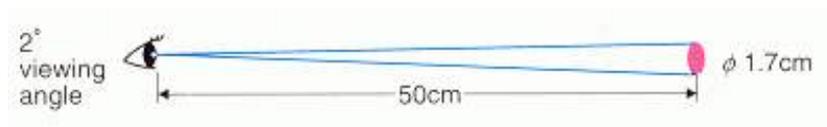


Figura 59. Observador de 2°.

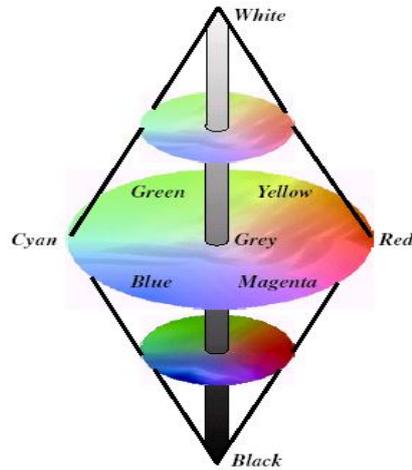


disponible en < <http://home.wanadoo.nl/paulschils/10.00.htm> > [noviembre de 2004]

En diagrama cromático del CIE (figura 21) se puede observar que la información de color se encuentra solamente en dos de sus componentes (x & y), puesto que en la componente Y se representa la claridad de los colores. Esto le da una ventaja sobre el modelo RGB en el cual la información de color esta contenida en sus tres componentes.

- **Modelo De Color HLS:** Este modelo utiliza los tres componentes básicos del color: matiz (Hue), saturación (Saturation) y brillo (Luminance). Una representación gráfica incluyendo estas tres propiedades genera el denominado *espacio y color*, el cual se ve en la figura 60.

Figura 60. Espacio y color HLS



RUSS John. The Image Processing Handbook. Boca Raton, FL. CRC PRESS, IEEE PRESS. 2002.

En el perímetro del disco están situados los colores en su forma mas pura, es decir con la ausencia de luz blanca. Al variar la distancia al centro del disco, los colores comienzan a mezclarse con la luz blanca, presentándose así los colores pasteles, esto se debe a que la distancia al centro (radio) señala los cambios en la componente de saturación (saturation) de los colores.

El barrido angular que se hace con una línea tomada desde el centro por toda el área del disco corresponde a cambios en el color. Esta coordenada angular corresponde al matiz (hue) del color.

Por ultimo el brillo (luminance), corresponde a la altura del doble cono y tiene un valor de 0 a 1, esto significa que a una altura cero se encuentra el color negro, a una de 0.5 se encuentra un gris de intensidad media, y a una altura de valor 1 se localiza el color blanco puro, lo cual indica que en la línea perpendicular que pasa por el centro de la circunferencia se encuentran los niveles de grises.

4.4.2 Algoritmo de color

A continuación se describe el algoritmo de segmentación por color basado en los modelos HLS y xyY, implementado en el presente proyecto.

Inicialmente se tiene una imagen RGB de la etapa de adquisición y luego se realiza una conversión de RGB al modelo de color seleccionado con anterioridad.

La transformación de RGB a HLS se describe con las siguientes ecuaciones:

$$H = \begin{cases} \theta & \text{si } B \leq G \\ 360 - \theta & \text{si } B > G \end{cases} \quad (1)$$

donde:

$$\theta = \cos^{-1} \left(\frac{\frac{1}{2}[(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right) \quad (2)$$

$$S = 1 - \frac{3}{R+G+B} [\min(R, G, B)] \quad (3)$$

La transformación de RGB a xyY se describe como:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124543 & 0.35758 & 0.180423 \\ 0.2126710 & 0.71516 & 0.072169 \\ 0.0193340 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

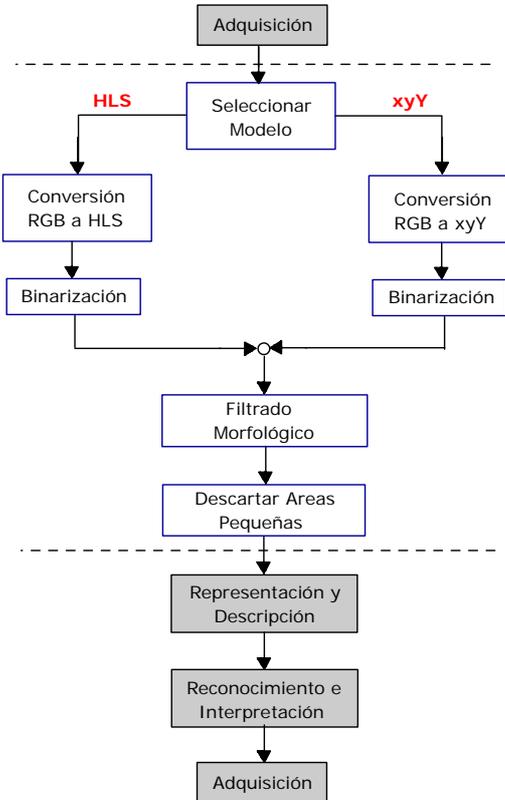
$$x = \frac{X}{X+Y+Z} \quad y = \frac{Y}{X+Y+Z} \quad (5)$$

donde la ecuación 4 corresponde a la transformación de RGB a los valores triestímulo XYZ, para el iluminante D65 [maynero] y un observador de 2°, y la ecuación 5 es la normalización de las componentes x e y.

Una vez realizada la transformación al modelo de color adecuado, se binariza la imagen obtenida de acuerdo con el color que se quiera identificar. A esta imagen binarizada se le aplica un filtrado morfológico basado en erosiones y relleno de huecos [woods] con el fin de obtener una buena segmentación de los objetos de interés en la escena. Los umbrales de binarización dependen en su valor de cada modelo de color y hacen parte de la base del conocimiento, es decir de los parámetros con los cuales se realice la segmentación. Finalmente se eliminan falsos candidatos si el área del objeto es menor a un área umbral predeterminada en el algoritmo.

En la figura 61 se muestra el diagrama para el algoritmo de color descrito anteriormente.

Figura 61. Diagrama de flujo para Algoritmo de Color



En las figuras 62 y 63 se observa el resultado de aplicar el algoritmo de color a algunas imágenes tomadas en el laboratorio. En el inciso b) de cada figura la imagen ya binarizada presenta ruido debido a condiciones de iluminación en el escenario. En el inciso c) se detalla el resultado final del proceso de detección y segmentación del color elegido después de haber aplicado el filtro morfológico a las imágenes del inciso b).

Cabe resaltar que estas imágenes no fueron adquiridas en el escenario planteado en este proyecto, sino en un escenario no controlado⁴, y puede verse que la iluminación no deja de ser un factor preponderante en el momento de realizar la

detección del color. No obstante se observa que con la utilización de los modelos de color implementados en este proyecto este problema se soluciona de forma completa.

Figura 62. Segmentación del color amarillo con modelo HLS.

a) Imagen original. b) Binarización. c) Imagen con filtrado morfológico

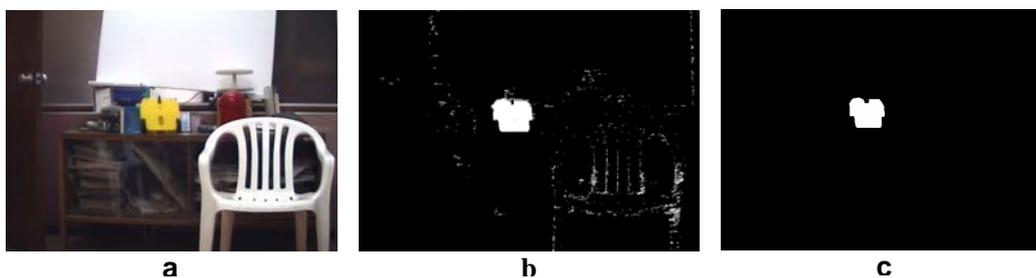
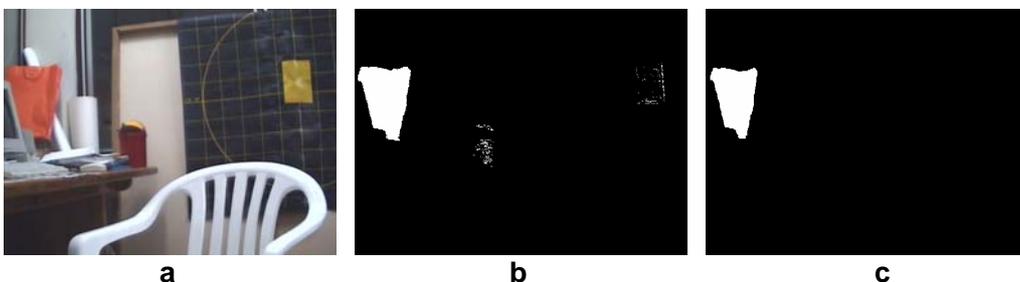


Figura 63. Segmentación del color naranja con modelo xyY

a) Imagen original. b) Binarización. c) Imagen con filtrado morfológico



4.4.3 Método basado en Gradientes

Estos métodos proveen una solución al problema de estimación de movimiento, a partir de la observación de los cambios en el brillo de una secuencia de imágenes. Aprovechando este fundamento, se puede detectar los bordes de un elemento en movimiento en cualquier escena dinámica.

En la estimación del movimiento por medio de métodos basados en gradientes uno de los principales problemas es la presencia de elementos no móviles que pueden estar en la escena y que interfieren con la correcta detección de los bordes de los elementos en movimiento. Por tal motivo en el proyecto aquí expuesto se implementó un algoritmo que se fundamenta en el algoritmo de Jain, el cual soluciona en gran medida este problema.

Algoritmo de Jain

Este algoritmo utiliza dos gradientes, el espacial y el temporal, los cuales se combinan por medio de un operador lógico AND, como se muestra en la siguiente ecuación:

$$E_T(x, y, t) = \frac{dF(x, y, t)}{dS} \bullet \frac{dF(x, y, t)}{dt} \quad (6)$$

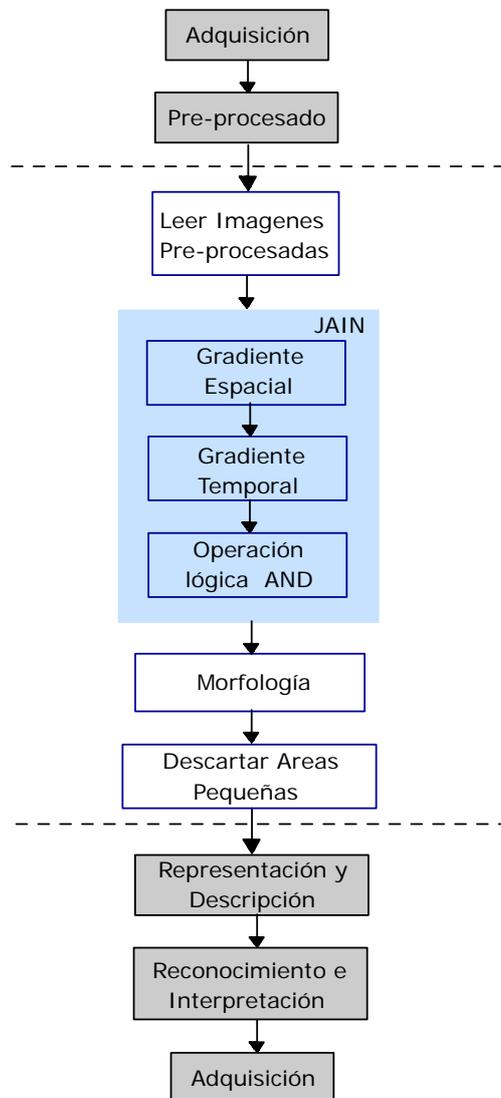
En donde $dF(x, y, t)/dS$ y $dF(x, y, t)/dt$ son respectivamente los gradientes espacial y temporal de la función de intensidad $F(x, y, t)$ ⁵. Para obtener el gradiente espacial se puede utilizar cualquier detector de bordes usado en procesamiento digital de imágenes, mientras que para determinar el gradiente temporal una opción consiste en realizar la diferencia entre los niveles de grises de dos imágenes diferentes de la misma escena. Cabe resaltar que una de las imágenes utilizadas en la diferencia de niveles de grises para calcular el gradiente temporal debe ser la misma que se utiliza para calcular el gradiente espacial.

Algoritmo para detección del movimiento.

⁵ Corresponde a los niveles de intensidad en cada píxel presente en una imagen.

Este algoritmo se basa fundamentalmente en el algoritmo de Jain y en algunas operaciones morfológicas como se puede apreciar en la figura 64.

Figura 64. Diagrama para Algoritmo basado en Gradientes



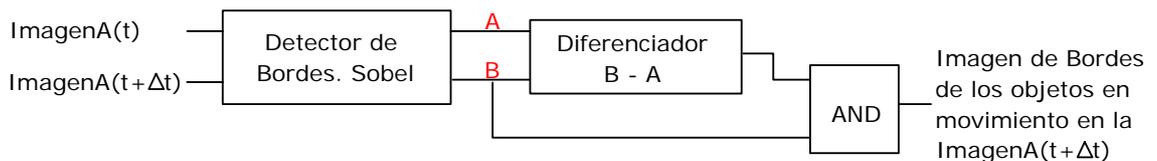
A continuación se describe el algoritmo de detección y segmentación de movimiento basado en gradientes implementado en el proyecto aquí expuesto:

En primera instancia, se leen dos imágenes en escala de grises que provienen de la etapa de pre-procesado, las cuales han sido adquiridas en instantes diferentes de una secuencia de imágenes.

$$\underbrace{\text{imagen } A(t) \dots \dots \text{ imagen } A(t + \Delta t) \dots \dots \dots}_{\text{SECUENCIA DE IMAGENES}} \quad (7)$$

Luego se aplica el método de Jain, el cual consiste inicialmente en hacer una detección de bordes a cada imagen por medio del detector de bordes de sobel [russ], después se realiza la diferencia entre la segunda y la primera de estas imágenes de bordes y a este resultado se aplica la operación lógica AND con la segunda imagen de bordes, obteniendo como resultado solamente los bordes del objeto en movimiento en la segunda imagen. El proceso se ilustra en la figura 65.

Figura 65. Operaciones del algoritmo de Jain

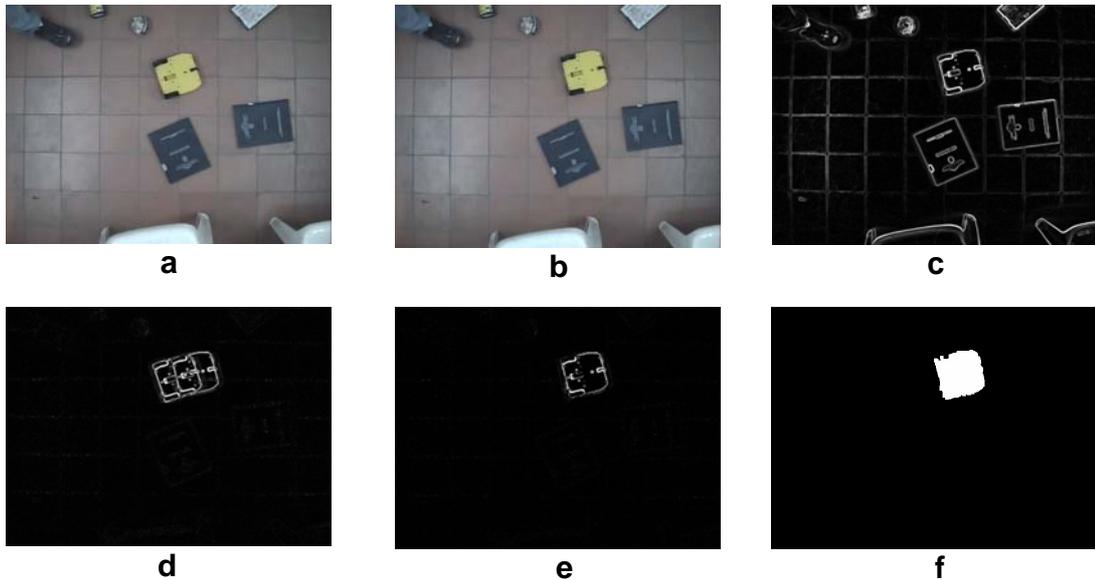


Finalmente por medio de operaciones morfológicas se realiza el relleno de la región correspondiente al objeto en movimiento y además se descartan las regiones con un área menor a un valor umbral fijado previamente.

Uno de los inconvenientes del algoritmo descrito anteriormente es que solo segmenta objetos en movimiento en una escena, por lo tanto, cuando un objeto que esta en movimiento se queda estático o sale de la escena, el algoritmo no obtendrá resultado alguno. Por esta razón se optó por realizar un algoritmo alternativo con el cual recuperar la información de localización de los objetos que ahora se encuentran fijos o que no se encuentran en la escena. Este algoritmo alternativo se expondrá en detalle en el siguiente capítulo.

Figura 66. Segmentación de movimiento con gradientes.

a) imagen $A(t_0)$. b) imagen $A(t_0 + \Delta t)$. c) Gradiente espacial. d) Gradiente temporal. e) bordes del objeto en movimiento f) objeto segmentado



En la figura 66 se observa la segmentación del objeto en movimiento (carro amarillo) realizada con el algoritmo basado en gradientes y en una escena no controlada. En los incisos c, d y e de la figura 66 se muestra el resultado de las operaciones del algoritmo de Jain. Finalmente en la figura 66-f se observa el resultado de aplicar operaciones morfológicas a la imagen 66-e obteniendo así la región interna del objeto para su posterior localización.

4.4.4 Método Basado En Correlación.

La correlación es la medida de la relación de igualdad que existe entre dos conjuntos de datos, los valores que toma, indican el grado de semejanza entre ellos.

Principios fundamentales

La correlación aplicada al análisis de imágenes y mas específicamente a la localización de objetos en movimiento dentro de una escena, consiste en

comparar secciones de una imagen para ver el nivel de semejanza con otra imagen, llamada modelo patrón [12].

La función de correlación normalmente utilizada en procesamiento digital de imágenes tiene la siguiente expresión matemática:

$$r = \sum_{i=1}^{i=N} I_i M_i \quad (8)$$

Donde M es el modelo patrón e I es la imagen a la cual se le va a encontrar la semejanza con el modelo patrón.

Esta función es máxima cuando la parte analizada de la imagen es muy semejante al modelo patrón. Infortunadamente, este método es impreciso si los niveles de brillo en la imagen son muy altos. En este caso es muy posible que la correlación alcance su máximo valor, cuando las imágenes no tienen ninguna correspondencia entre ellas; por tal motivo en el presente proyecto se utilizara la correlación normalizada, como una solución práctica a este inconveniente.

Correlación Normalizada. Es una función más compleja, en la cual el resultado no se ve afectado por cambios lineales en la imagen o valores de los píxeles en el modelo. Además, la función alcanza su valor máximo únicamente cuando la imagen y el modelo tienen una correspondencia exacta [9]. La función de correlación normalizada tiene la siguiente expresión:

$$r = \frac{N \sum IM - \sum I \sum M}{\sqrt{(N \sum I^2 - (\sum I)^2)(N \sum M^2 - (\sum M)^2)}} \quad (8)$$

Donde N corresponde al número de píxeles analizados e I y M corresponden a la imagen y modelo patrón, respectivamente.

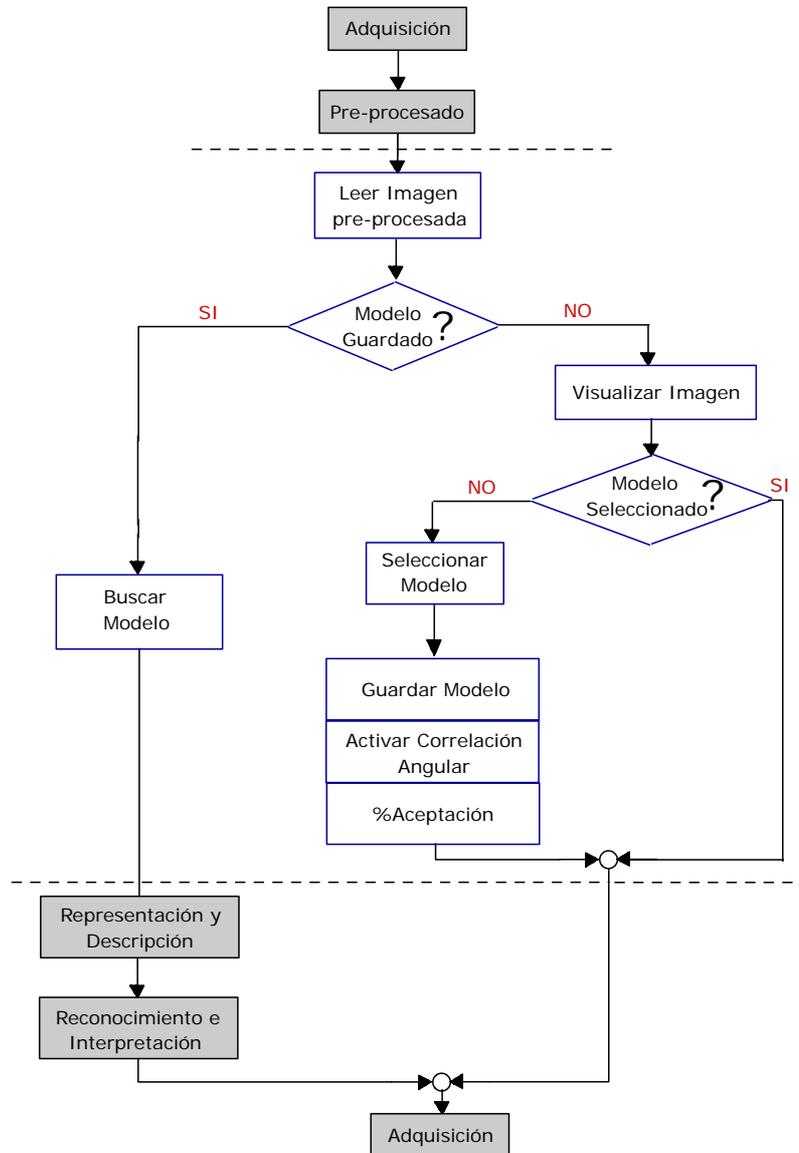
Algoritmo basado en Correlación

El objetivo principal de este algoritmo es seleccionar como modelo patrón, un objeto presente en una primera imagen en escala de grises de la escena, para posteriormente, por medio de correlación, realizar un seguimiento del móvil en las imágenes subsecuentes.

Para tal fin, se implementó un algoritmo eficaz, que para la correlación tiene en cuenta al modelo no solamente en la orientación original sino también, en posiciones rotadas, con lo cual, el modelo solo se almacene en memoria una vez. El Algoritmo utiliza un valor de aceptación para tomar o rechazar un resultado el cual esta muy ligado a las condiciones de iluminación y ha sido fijado previamente de acuerdo a las condiciones del laboratorio.

El diagrama del algoritmo para el seguimiento por medio de correlación se describe a continuación y se visualiza en la figura 67.

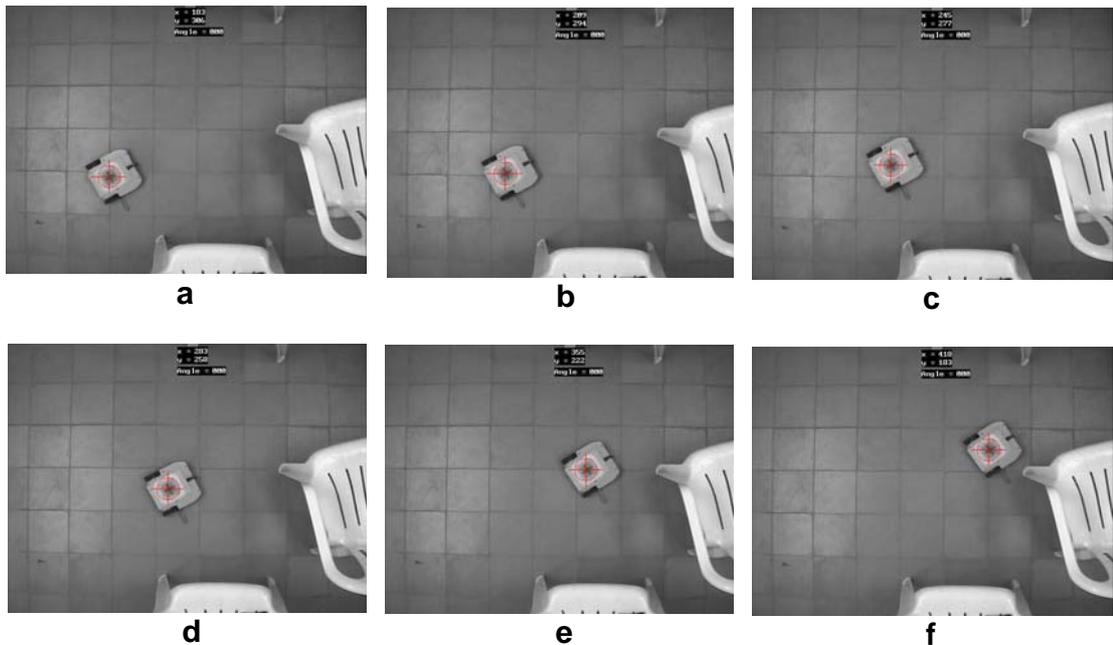
Figura 67. Diagrama del algoritmo de correlación



Inicialmente se lee una imagen en escala de grises que ha sido pre-procesada. Seguidamente se pregunta si el modelo ya está almacenado en memoria. Si la respuesta es afirmativa se ejecuta la función de correlación para buscar dicho modelo en la imagen adquirida, de lo contrario se visualiza la imagen con el propósito de que el usuario pueda seleccionar por medio del clic izquierdo del mouse el centro del modelo. Finalmente si se ha seleccionado satisfactoriamente un modelo, éste se guarda, se habilita la función de correlación angular, la cual

tiene en cuenta los movimientos rotacionales del móvil y se fija un valor umbral con el cual se decide si un resultado es aceptable o no.

Figura 68. Resultados del algoritmo de correlación



En las figuras 68-a a 68-f, se observa la secuencia de seguimiento de un móvil por medio del algoritmo de correlación, la marca roja indica el lugar donde la correlación arrojó el mayor nivel de semejanza y esta posición coincide satisfactoriamente con el modelo elegido.

4.5 LOCALIZACIÓN E IDENTIFICACION

La información en píxeles proporcionada por las imágenes toma importancia en el momento en que se la relaciona con alguna magnitud física como el área, la longitud etc, con la cual se pueda sacar conclusiones acerca de lo que ocurre en el mundo real que es captado por la cámara. La localización e identificación en la pista es la etapa que relaciona finalmente la información obtenida por el sensor que en este caso es la cámara, con la información física real en la escena.

4.5.1. Localización de objetos en la imagen

Después de realizado el proceso de detección y segmentación de movimiento es necesario realizar la localización del centroide del objeto segmentado con respecto al origen de coordenadas de la imagen. Aunque el centroide, está relacionado con una propiedad física de los objetos como es su masa, tiene una aplicación particular en el procesamiento digital de imágenes. A continuación se describe de forma breve el concepto de momentos, necesario para realizar el calculo del centroide aplicado al procesamiento digital de imágenes.

- **Momentos.** Son propiedades numéricas que dependen de las coordenadas y del valor que toman las variables en función de ellas. Para una función discreta⁶ el momento (p,q) se define de la siguiente manera :

$$M(p, q) = \sum_x \sum_y x^p y^q f(x, y) \quad (9)$$

- **Centroide.** se obtiene por medio de los momentos de orden 1 $M(1,0)$, $M(0,1)$ y de orden cero $M(0,0)$. Este último corresponde al área en píxeles del objeto segmentado. El calculo realizado con estos momentos se muestra en la siguiente ecuación.

$$X = \frac{M(1,0)}{M(0,0)} \quad Y = \frac{M(0,1)}{M(0,0)} \quad (10)$$

4.5.2. Algoritmo de Identificación

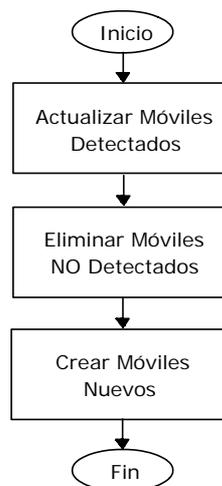
En la escena se pueden tener múltiples móviles (máximo seis), por tanto se requiere de un algoritmo que esté constantemente identificando a cada uno de los

⁶ En procesamiento digital de imágenes, corresponde a los niveles de intensidad en cada una de las posiciones dentro de la imagen.

móviles con el fin de realizar un correcto seguimiento de ellos y no llegar a confundir uno con otro.

La etapa de identificación consta de tres fases fundamentales: actualizar móviles detectados, eliminar móviles no detectados y detectar móviles nuevos. En la figura 69 se muestra el diagrama de flujo para este algoritmo.

Figura 69. Diagrama de flujo para algoritmo de identificación



A continuación se describe cada una de las fases del algoritmo de identificación.

Actualización de móviles detectados Consiste en actualizar la información de posición y orientación de cada uno de los móviles que habían sido detectados en la imagen anterior y que se han detectado en la imagen actual.

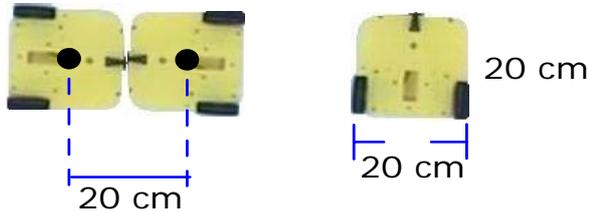
Eliminación de móviles NO detectados: consiste en eliminar la información de posición y orientación de un móvil que no ha sido detectado en la imagen actual pero que se había detectado en la imagen anterior.

Detección de móviles nuevos. En esta fase se agrega la información de posición y orientación de un móvil que se ha detectado en la imagen actual pero que no se había detectado en la imagen anterior.

La manera de conocer si un móvil que se había detectado en la imagen anterior aparece nuevamente en la imagen actual o ha desaparecido en esta última, se explica a continuación:

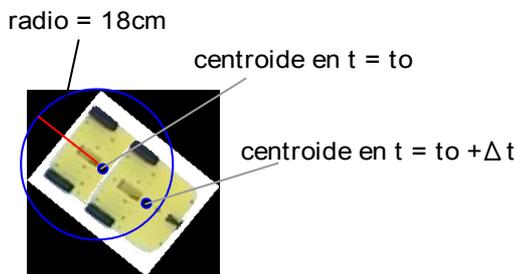
Se parte del hecho que las dimensiones de los móviles utilizados en el presente proyecto son de 20cm x 20cm y que cuando dos móviles están lo más cerca posible sin haber chocado, sus centros de gravedad se encuentran a una distancia de 20cm aproximadamente (figura 70).

Figura 70. Dimensiones de los móviles



Consecuentemente, para que pueda realizarse un correcto seguimiento de cada uno de los móviles en la escena, es necesario que el centroide de un objeto se encuentre dentro de una circunferencia de radio 18 cm con origen en el centroide del mismo objeto en la imagen anterior. (figura 71).

Figura 71. Identificación de un móvil



Al fijarse el radio de la circunferencia en 18 cm, se asegura que nunca se va a confundir un móvil que se desplaza, con otro objeto cercano. Como contraparte, esta elección limita la velocidad máxima a la cual puede desplazarse un móvil para poder ser identificado y realizar un correcto seguimiento del mismo. Esta velocidad es:

$$v_{\max} = \frac{x_{\max}}{\Delta t} \quad (9)$$
$$v_{\max} = \frac{18 \left[\frac{cm}{s} \right]}{\Delta t}$$

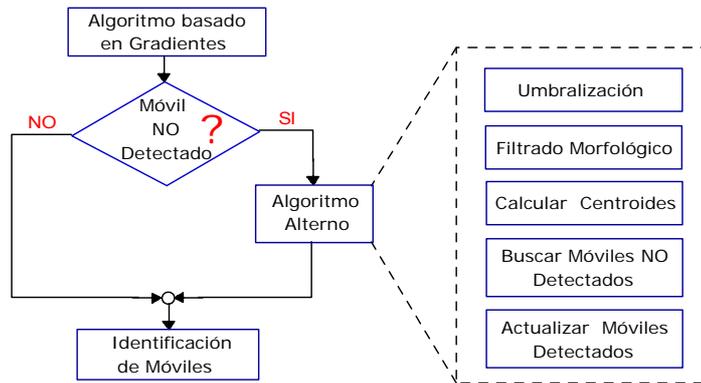
donde v_{\max} y Δt corresponden respectivamente a la velocidad máxima de desplazamiento del móvil y al tiempo que requiere la aplicación para adquirir una imagen y procesarla. El tiempo de procesamiento depende del algoritmo en ejecución.

El algoritmo descrito anteriormente, funciona correctamente siempre y cuando los móviles puedan ser localizados en todo instante, sin importar si están en movimiento o no. Cuando se ejecuta el algoritmo basado en correlación o el algoritmo basado en color, siempre se logra localizar los objetos estáticos o en movimiento, sin embargo esto no ocurre cuando se utiliza el algoritmo basado en gradientes, el cual localiza solo los objetos en movimiento.

Para subsanar este inconveniente, se optó por implementar un algoritmo alternativo, con el cual se pueda establecer la localización de un objeto móvil cuando se

quede estático o salga de la escena. El diagrama para el algoritmo alternativo se muestra en la figura 72.

Figura 72. Diagrama par el algoritmo alternativo



El algoritmo alternativo mostrado en la figura 72, solo se ejecuta cuando el número de móviles detectados en la imagen actual es inferior al detectado en la imagen anterior.

Este algoritmo inicialmente consiste en hacer una umbralización seguida de un filtrado morfológico para segmentar los móviles en la escena. Luego se realiza el cálculo del centroide de cada objeto segmentado. Finalmente se efectúa la búsqueda de los móviles no detectados por medio de comparación con los móviles detectados en la imagen anterior y se actualiza la información de estos móviles. La actualización consiste en establecer la posición de los móviles que ahora están estáticos o eliminar los móviles que ya no están en la escena.

4.5.3 Cálculo de Orientación del móvil

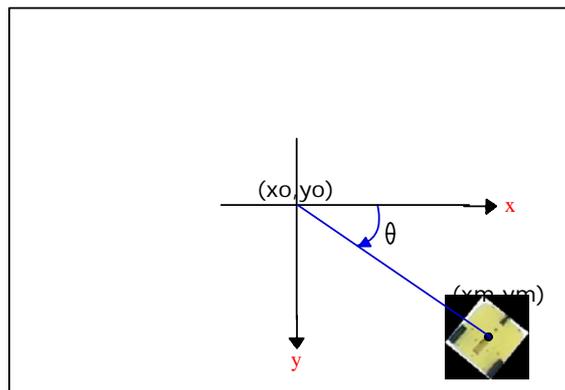
En la aplicación implementada, la orientación se puede calcular respecto a dos sistemas de referencia. El primero es un sistema de referencia fijo el cual se

puede establecer en cualquier lugar de la escena y el segundo es de referencia variable en cada instante.

Sistema de referencia fijo

En este sistema es necesario fijar el origen de coordenadas en un punto cualquiera de la escena. En la figura 73, se puede observar el sistema de referencia fijado en el centro de la escena.

Figura 73. Sistema de referencia fijado en el centro de la escena



En el sistema de referencia de la figura 73, el ángulo para un móvil aumenta en el sentido de las manecillas del reloj respecto al eje positivo x , desde 0° hasta 360° . Este ángulo se puede calcular por medio de la relación mostrada en la ecuación 10.

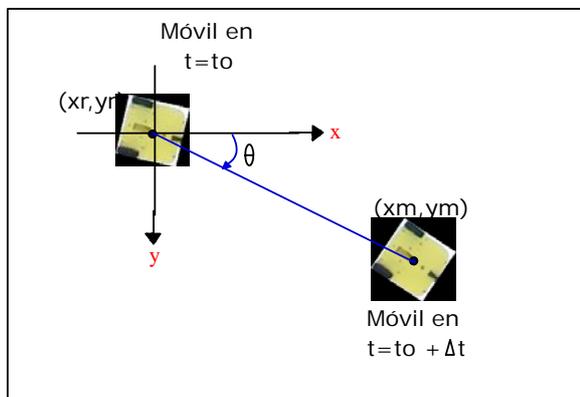
$$\theta = \tan^{-1}\left(\frac{ym - yo}{xm - xo}\right) \text{ [radianes]} \quad (10)$$
$$\theta = \frac{180}{\pi} \theta \text{ [grados]}$$

Donde el punto (x_0, y_0) es el origen de coordenadas fijo y el punto (x_m, y_m) hace referencia al centroide del móvil en cada instante.

Sistema de referencia variable

La referencia se fija en el centroide del objeto y cambia en cada instante conforme este se mueve. Este sistema se muestra en la figura 74.

Figura 74. Sistema de referencia variable



En la figura 74 se puede apreciar un móvil ubicado en instantes diferentes de tiempo. Cuando $t = t_0$ el móvil se encuentra en el punto (x_r, y_r) , un instante después, en $t = t_0 + \Delta t$, el móvil está ubicado en el punto (x_m, y_m) . La orientación del movimiento del objeto se puede calcular por medio de ecuación 11.

$$\theta = \tan^{-1} \left(\frac{y_m - y_r}{x_m - x_r} \right) [\text{radianes}] \quad (11)$$

$$\theta = \frac{180}{\pi} \theta [\text{grados}]$$

5. PRUEBAS Y ANÁLISIS DE RESULTADOS

5.1 PRUEBAS REALIZADAS PARA EL SISTEMA DE VISIÓN

En este capítulo se presentan las pruebas realizadas a los algoritmos implementados. Estas pruebas son:

- Posición.
- Orientación.
- Seguimiento de trayectorias.
- Tiempo de latencia

Para realizar estas pruebas se utilizó la aplicación (FENIX) en visual C++ 6.0, la cual genera un archivo de texto con la información de posición y orientación de los objetos móviles. Además para emular el objeto en movimiento o estático se usó en todas las pruebas un rectángulo de cartulina cuyas dimensiones son 17cm x 12cm con el fin de localizar fácilmente su centroide en la intersección de sus diagonales.

El análisis de cada uno de los resultados obtenidos en las pruebas mencionadas permitirá medir el desempeño de la aplicación en la localización tanto de objetos móviles como estáticos.

El error, el valor medio y la desviación estándar para las pruebas se calcula por medio de las siguientes relaciones:

$$error = valor\ medido - valor\ real \quad (12)$$

$$valor\ medio = \frac{1}{n} \sum_{i=0}^n x_i \quad (13)$$

$$desviación\ estándar = \sqrt{\frac{1}{n-1} \sum_{i=0}^n (x_i - valor\ medio)^2} \quad (14)$$

donde **n** es el número de muestras.

El calculo del error se efectúa con el fin de medir el nivel de exactitud, mientras la desviación estándar para establecer el nivel de precisión.

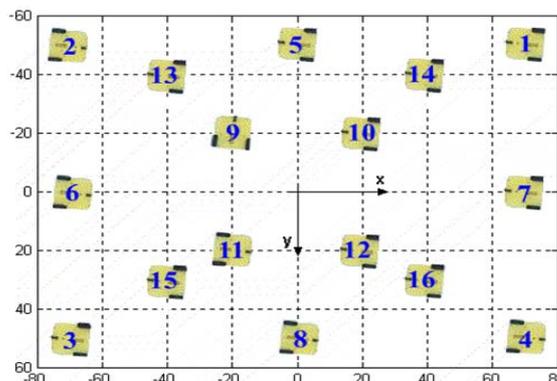
5.1.1 Posición

El objetivo principal de esta prueba es medir el nivel de exactitud y precisión que tiene la aplicación desarrollada cuando se usa para localizar (posición) los objetos de interés en la escena.

Para realizar esta prueba, se dividió la pista en una cuadrícula con divisiones de 10cm x 10cm simulando en el mundo real un sistema de coordenadas cartesiano con origen en el centro de la pista . A continuación se ubicó un objeto en diferentes posiciones en la pista, de tal manera que el centroide del objeto coincidiera con una posición (x,y) conocida en el sistema de referencia.

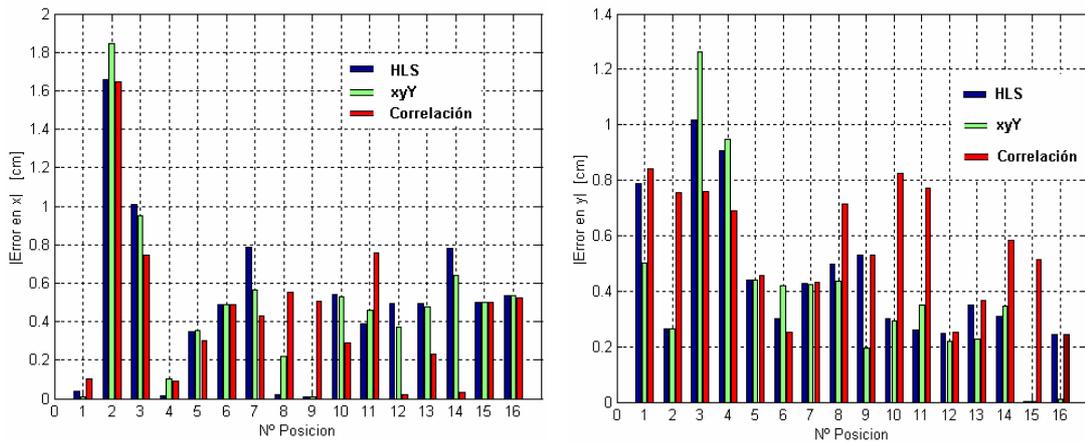
Por ultimo se localizó el objeto con la aplicación desarrollada, tomando para cada posición 50 datos. Las posiciones elegidas están distribuidas de tal manera que cubran la mayor parte del área de trabajo como se observa en la figura 75.

Figura 75. Posiciones usadas para determinar la distribución del error.



Para esta prueba, el algoritmo basado en gradientes no arroja resultados ya que el objeto que se va a localizar no se encuentra en movimiento, por lo cual este algoritmo se excluye de la misma.

Figura 76. Distribución del error de posición en los algoritmos implementados



a) Error en las coordenadas en x . b) Error en las coordenadas en y

En la distribución de la figura 76 del error se observa que el error en posición es mayor a un centímetro para las coordenadas en x para la posición 2 y para las coordenadas en y en las posiciones 3 y 4. Estos errores se deben a que los puntos mencionados anteriormente se encuentran localizados en los extremos de la imagen, lugar en el cual se presenta el mayor nivel de distorsión.

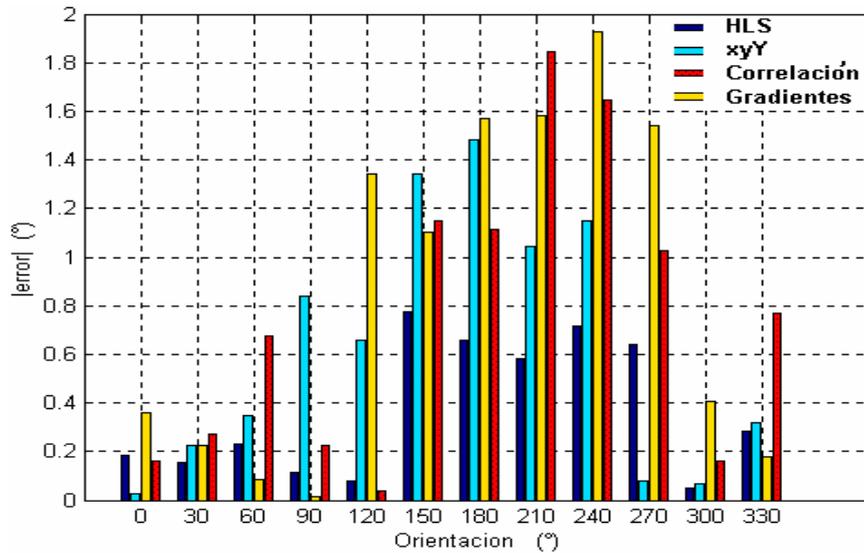
5.1.2 Orientación

Para realizar esta prueba, se simuló un sistema de coordenadas polar fijado en el centro de la pista y se realizaron divisiones de 30° en un rango de 0° a 360°.

Luego se ubicó un objeto en diferentes posiciones, de tal manera que el ángulo formado por éste respecto al origen de coordenadas, variara desde 0° hasta 330° con espaciamiento de 30°. Para cada posición se tomaron 50 datos del ángulo con los cuales se realizó su posterior análisis.

En la figura 77 el error para el algoritmo basado en el modelo de color HLS se muestra con barras verticales de color azul oscuro, para el algoritmo basado en el modelo de color xyY con barras de color azul claro, el algoritmo basado en correlación con barras amarillas y el algoritmo basado en gradientes se identifica con barras rojas.

Figura 77. Error en orientación.



En la anterior figura se puede comparar gráficamente el error en orientación para cada uno de los algoritmos.

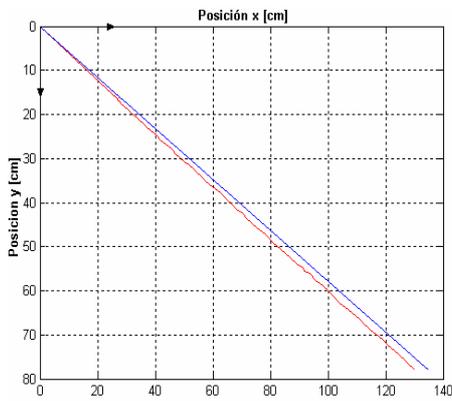
5.1.3 Seguimiento

Esta prueba se realizó con el fin de comprobar en la aplicación desarrollada el nivel de fidelidad al realizar el seguimiento de un móvil que describe una trayectoria conocida.

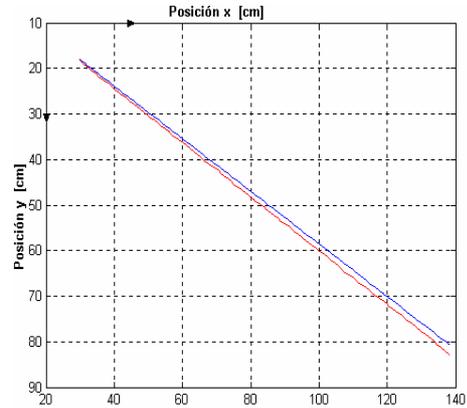
Para la ejecución de la prueba se optó por usar dos trayectorias: la primera es una línea recta con una pendiente de 30° con respecto al eje x. En la figura 78 de color azul se observa la trayectoria ideal y de color rojo el seguimiento con los algoritmos implementados

Figura 78. Seguimiento de una trayectoria recta

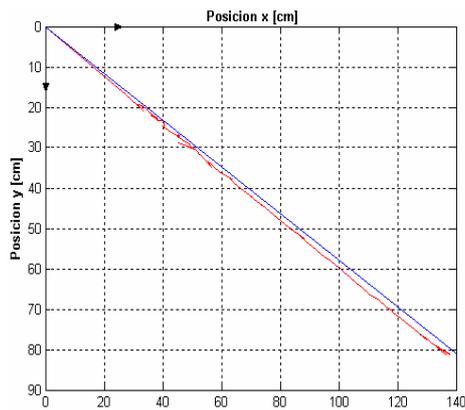
a) con el modelo de color HLS b) con el modelo de color xyY c) con el algoritmo basado en gradientes d) con correlación



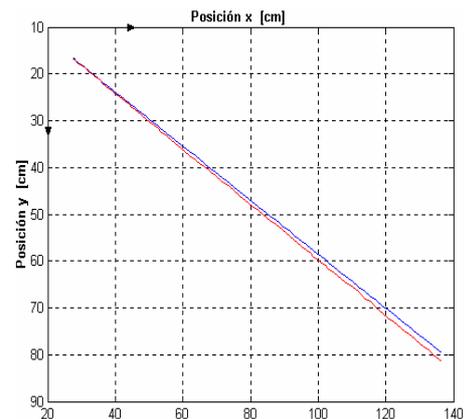
a



b



c



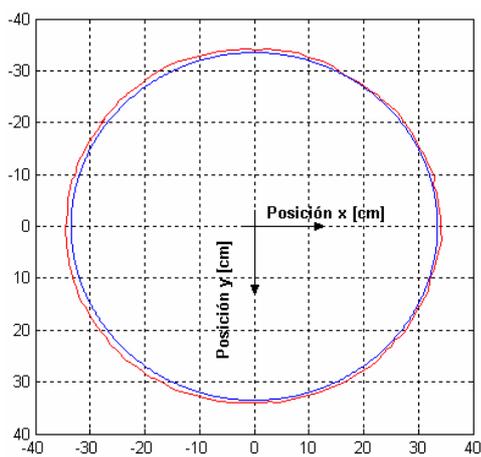
d

Aunque el seguimiento de una trayectoria rectilínea es altamente fiel para los cuatro algoritmos implementados, se observó un leve cambio en la pendiente de la recta. Esto es ocasionado por errores humanos al momento de medir el ángulo de la pendiente en la trayectoria descrita para el objeto en movimiento.

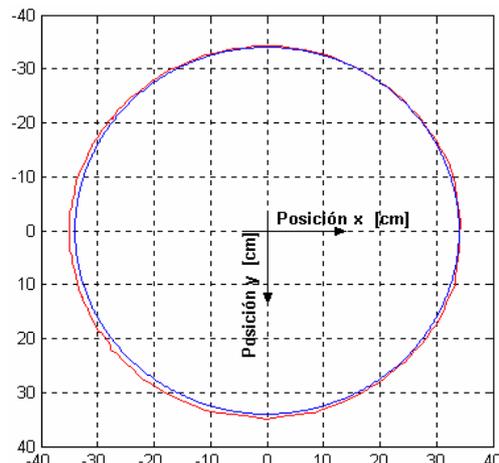
La segunda trayectoria descrita es circular. Los resultados para el seguimiento con los algoritmos implementados se observan en la figura 79.

Figura 79. Seguimiento de una trayectoria circular

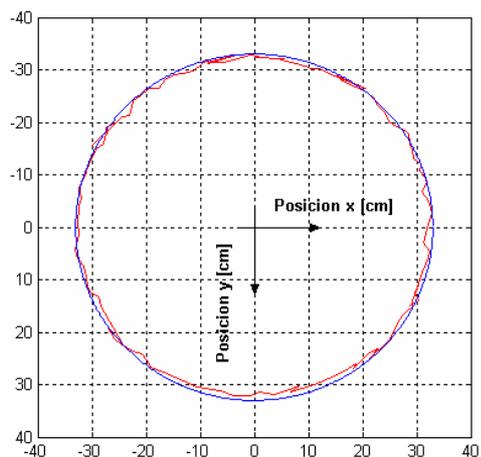
a) con el modelo de color HLS b) con el modelo de color xyY c) con el algoritmo basado en gradientes d) con correlación



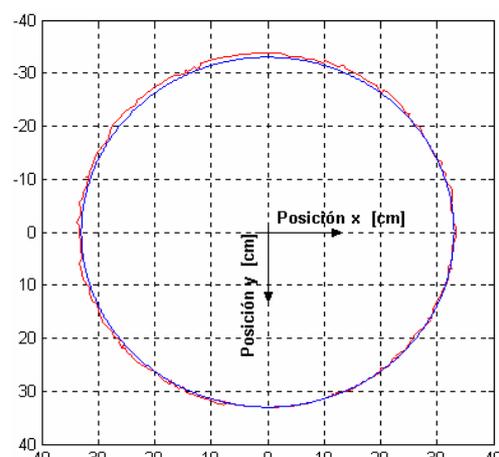
a



b



c



d

Nuevamente en color azul se observa la trayectoria ideal y en color rojo, el seguimiento realizado por la aplicación. En este caso el seguimiento tiene una alta precisión salvo el algoritmo de gradientes donde se presentan variaciones

5.1.4 Tiempo de latencia

Esta prueba tiene como objetivo mostrar la rapidez de respuesta de cada uno de los algoritmos implementados.

Esta prueba se realizó colocado sobre la pista seis objetos móviles y seis obstáculos. La razón de tener el máximo número de objetos móviles y de obstáculos en la escena, es porque de esta manera se obtiene el tiempo crítico para cada algoritmo ya que al incrementar el número de objetos de interés en la escena también incrementa el tiempo de procesamiento.

En la tabla 7 se muestran los tiempos de adquisición, procesamiento y el tiempo total de latencia el cual es la suma de los dos anteriores. Cuando los objetos móviles se detectan por correlación o gradientes, los obstáculos pueden ser detectados con base en el modelo de color xyY o al HLS. Cuando los objetos móviles son detectados por el algoritmo basado en un modelo de color, xyY o HLS, los obstáculos son detectados con base en el mismo modelo de color.

Tabla 7. Tiempos de latencia

Método Tiempo de latencia (seg)	Correlación		xyY	HLS	Gradientes		Móviles
	xyY	HLS			xyY	HLS	Obstáculos
Adquisición	0,0735	0,0735	0,0735	0,0735	0,0735	0,0735	
Procesamiento	0.8108	0.3018	0.9926	0.1661	0.7742	0.3000	
Total	0,8821	0,3748	1,0643	0,2429	0,8515	0,3712	

En la tabla 7 se observa que el tiempo total de latencia varía desde 0.2429s hasta 1.0643s. El algoritmo basado en el modelo de color HLS es el más rápido con un tiempo de latencia de 0.2429s.

El tiempo de latencia limita la velocidad máxima como se observa en la tabla 8, con la cual pueden moverse los objetos para poder ser localizados e identificados correctamente con la aplicación desarrollada. Esto se muestra a continuación:

Tabla 8. Velocidad máxima de procesamiento.

	Correlación		xyY	HLS	Gradientes		Móviles
	xyY	HLS			xyY	HLS	Obstáculos
Velocidad máxima (cm/s)	20,405	48.025	16.912	74.104	21.139	48.491	

5.2 CARACTERIZACION DEL MODULO DE ULTRASONIDO SRF04

Una vez conocido el funcionamiento del modulo de ultrasonido se realizo la caracterización del mismo en el laboratorio, utilizando un generador de señales para producir la excitación y un osciloscopio digital para capturar y observar el pulso de eco entregado por el sensor. En el procedimiento para la calibración se coloco un objeto frente al sensor; se registro la distancia entre el objeto y el sensor y el ancho del pulso de eco medido en el osciloscopio. Para cada distancia se excitó el sensor 5 veces. En la tabla 9 se presenta el promedio de ancho del pulso de eco para cada distancia.

Tabla 9. Resultados del laboratorio en la calibración del SRF04.

Distancia del Objeto (cm)	Ancho del Pulso de Eco (mseg)
3	0.22
6	0.43
9	0.59
10	0.67
12	0.81
15	0.91
18	1.06
20	1.22
21	1.28
24	1.46
25	1.52
44	2.56
64	3.8

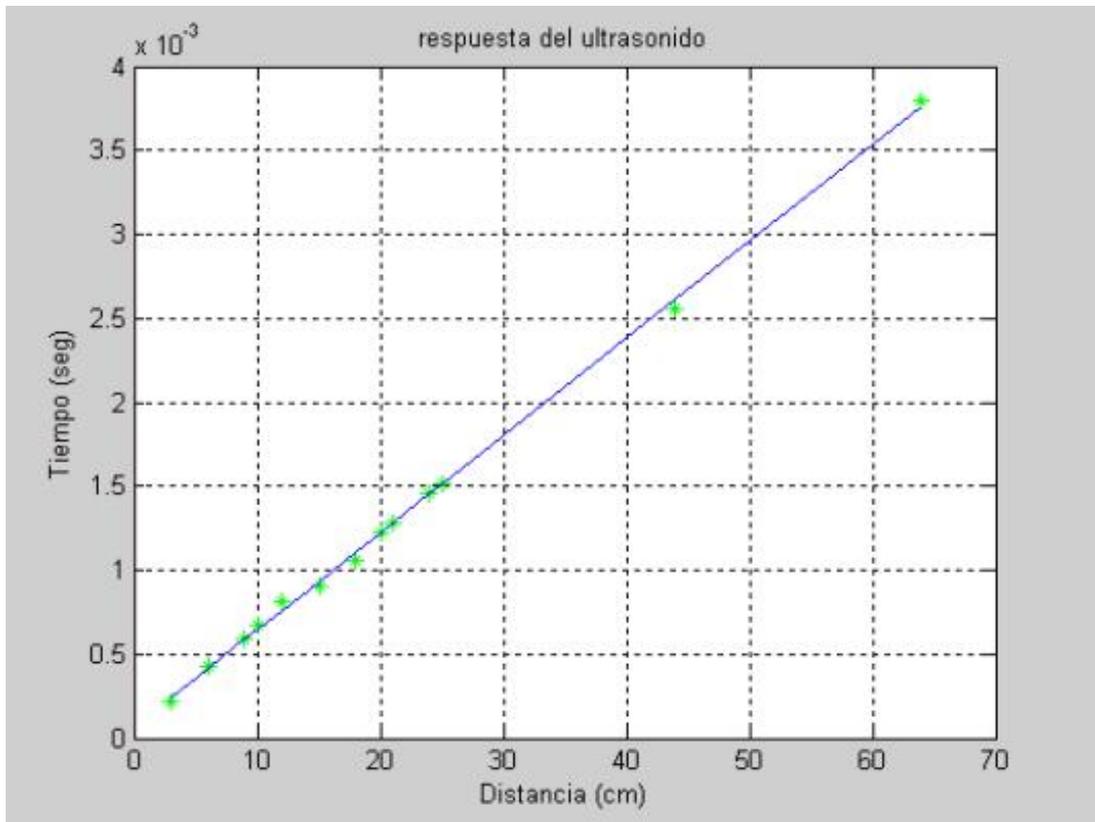
En la figura 80 se puede observar la distancia sensada v/s el tiempo de duración del pulso de eco. Estos datos presentan un comportamiento lineal

En la grafica los puntos en color verde son los datos tomados en el laboratorio y la línea azul es la regresión lineal de esto puntos, que se representan por la siguiente ecuación:

$$Y = 5.78022913408e-5 * X + 6.65830035889e-5 \quad (15)$$

Esta ecuación es utilizada para calcular el ancho del pulso de eco en el rango de distancias que sensa este modulo.

Figura 80. Caracterización del sensor SRF04



5.3 PRUEBAS DE DESEMPEÑO DEL ROBOT MÓVIL

Para el análisis del desempeño de los robots móviles se realizaron diferentes pruebas con el fin de ajustar el control y la comunicación con el PC. Estas pruebas son:

- Comunicación
- Precisión
- Seguimiento de trayectoria
- Repetibilidad
- Evitación de obstáculos

El análisis de cada uno de los resultados obtenidos permitirá medir el desempeño de cada robot móvil.

5.3.1 Comunicación

La primer prueba que se realizo fue la de la comunicación vía radio frecuencia enviando instrucciones desde el PC y observando que el móvil entendiera dichas instrucciones. Como se menciona en el capítulo anterior la comunicación se realizo en grupo de ocho bits con una velocidad de 1200 bits por segundo y un bit de parada.

Una vez obtenida la comunicación se programo el microcontrolador dejando la posibilidad de modificar desde el computador las siguientes variables: ancho de pulso del servomotor derecho, ancho de pulso del servomotor izquierdo, distancia, ángulo de giro.

5.3.2 Precisión

La primera prueba que se realizo fue calibrar el móvil para que ejecutara una línea recta. Primero

Para esta prueba se vario el ancho de pulso a cada servomotor buscando que el móvil siguiera una línea recta que se dibujo en la pista. Después de haber realizado la prueba varias veces se encontró que el móvil siempre se salía de la línea a uno de sus lados. Una solución a esto fue modificar el prescaler del microcontrolador para aumentar la resolución en los pulsos enviados. Se realizo nuevamente la prueba pero el cambio fue mínimo. Se concluyo que este error es debido a la ubicación de los servomotores en el eje ya que están invertidos (es decir que para que el móvil avance hacia delante un servomotor recibe un ancho de pulso mayor a 1.5 ms y el otro uno menor). Para corregir este problema se invirtió la polaridad al motor de continua de uno de los servomotores para que el

móvil avance hacia delante con un ancho de pulso en ambos servomotores mayor a 1.5 ms. Recordemos que para un ancho de pulso igual a 1.5 ms los servomotores no se mueven.

Por ultimo a cada servomotor se le cambio el potenciómetro por un trimer para obtener mayor precisión y también para evitar que se gire accidentalmente uno de estos.

Una vez fijado los trimers se realizo las mediciones de la prueba obteniendo los resultados que se muestran en la tabla 10.

Tabla 10. Prueba de precisión en el robot móvil

MEDICION	DESVIACION DEL PUNTO DE LLEGADA (cm)	ANGULO DE LLEGADA (grados)
1	+1.5	+3
2	-2.2	-1
3	-3.5	-2
4	-2.3	-2
5	+3.5	+6
6	-0.9	+1
7	0	+1
8	-1.1	0
9	-1.4	+1
10	+1.4	+1
11	-2.7	0
12	-0.1	0

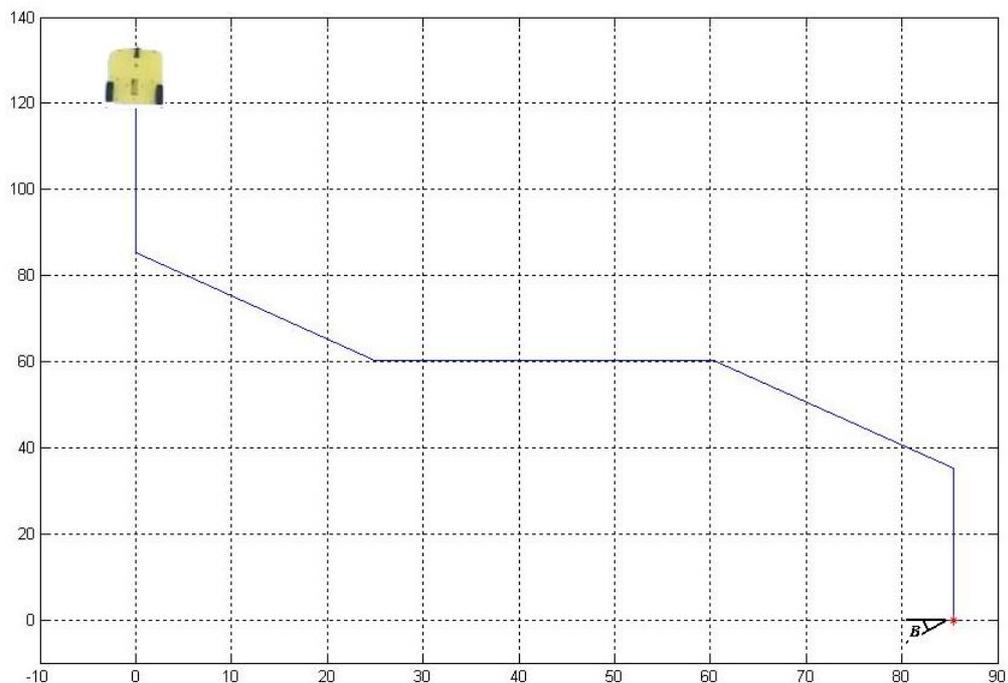
El signo de la desviación del punto de llegada representa la orientación de llegada del móvil. Un signo positivo indica que el movil llego a la derecha del punto esperado, el signo negativo indica que llego a la izquierda.

El signo del ángulo de llegada se tomo como negativo cuando el móvil llega con una orientación en sentido horario, con respecto a la orientación en la partida. Con esta prueba se observa una desviación máxima de +/- 3.5 centímetros y una desviación promedio de 1.91 centímetros para un avance de 80 centímetros. El ángulo de desviación máximo es de 8 grados y el ángulo de desviación promedio es de +/- 3 grados con respecto a la orientación en el punto de partida.

5.3.3 Repetibilidad

Para la prueba de repetibilidad se planifico una trayectoria en la cual se ubica el movil en el mismo punto de partida y con la misma orientación y se mide el punto de llegada, teniendo en cuenta las diferencias con el punto teorico. Para cada medida se tomo el punto de llegada como el centro del eje del movil, y la orientación del móvil como el ángulo de su eje respecto a la horizontal. La trayectoria y el sistema de coordenadas para esta prueba se observa en la figura 79.

Figura 79. Trayectoria y sistema de coordenadas de la prueba de Repetibilidad.



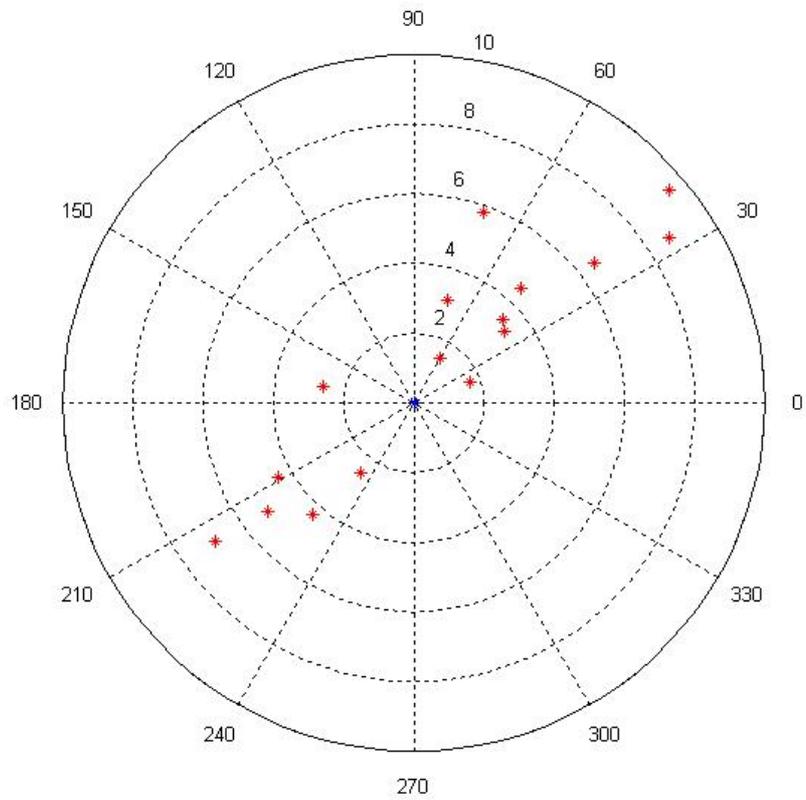
En la prueba el móvil se debe desplazar desde la coordenada (0,120.7cm), hasta la coordenada (85.35cm,0) siguiendo la trayectoria descrita. La ubicación de los puntos de llegada se representa en coordenadas polares definiendo el origen como la coordenada de llegada. Estas medidas se observan en la tabla 11.

Tabla 11. Resultados de la prueba de repetibilidad.

MEDICION	PUNTO DE LLEGADA		ANGULO DE ORIENTACION(β)
	DISTANCIA(cm)	ANGULO(grados)	
1	5.8	70	-1
2	2.6	170	0
3	3.1	72	0
4	4.5	47	-1
5	6.9	215	+4
6	1.5	60	+2
7	3.5	43	0
8	9.5	40	-4
9	3.3	38	+1
10	4.4	209	+3
11	1.7	21	+1
12	8.7	33	-3
13	4.3	228	+2
14	5.2	217	+2
15	6.5	38	-2
16	2.5	233	+1

Con el fin de visualizar la diferencia de los puntos de llegada medidos respecto al punto de llegada teórico, se realizó una gráfica de dispersión que se observa en la figura 81.

Figura 81. Dispersión de los datos en la prueba de repetibilidad.



En esta prueba se obtuvo un error de 4.62cm promedio de diferencia entre el punto de llegada medido y el teórico, y un error máximo en la orientación de +/-4 grados.

5.4 PROGRAMAS ELABORADOS

5.4.1 Campos de potencial

Para este método se desarrolló tres interfaces gráficas dos de las cuales son usadas para planeamiento off-line y otra para planeamiento on-line. Estas interfaces se realizaron usando el GUIDE de Matlab y aparcan en la figura 82 y 83.

Figura 82. Interfaz off-line de campos de potencial

The figure shows a MATLAB GUIDE window titled "guide" with a light beige background. At the top left is the UIS logo. To its right, the text reads "UNIVERSIDAD INDUSTRIAL DE SANTANDER" and "ESCUELA DE INGENIERIAS ELECTRICA, ELECTRONICA Y DE TELECOMUNICACIONES". Below the logo are input fields for "tam pista en x" and "Tam pista en y". Further down are "punto de inicio" and "punto de llegada" fields. At the bottom left, there are three rows of obstacle input fields labeled "obstaculo1", "obstaculo2", and "obstaculo3", each with "pos x" and "pos y" sub-labels. A "numero de obstaculos" field is also present. On the right side, a grid plot with axes from 0 to 1 is shown. At the bottom right, there are two buttons: "cargar datos" and "calcular".

Figura 83. Interfaz off-line que lee los valores de los obstáculos fijos dentro de la pista usando cámara.

posiciones obstaculos		
	pos x	pos y
obstaculo1	22.2215	75.6633
obstaculo2	112.858	105.879
obstaculo3	160.599	32.2382

posicion movil		
movil1	112.351	68.5247
movil2	112.862	112.351

La interfaz para el algoritmo on-line es igual a la de la figura 83, solamente que ella trae ya definida por defecto los valores de tamaño de la pista y el numero de obstáculos. Estos obstáculos son como se enunció en los objetivos del proyecto dos obstáculos fijos y uno móvil.

La captura de los datos se hace mediante visión reconociendo el color azul para los obstáculos fijos y el amarillo para el obstáculo móvil y el carro que realiza el planeamiento.

5.4.2 Algoritmo genético

El algoritmo genético fue implementado en VISUAL C#.NET. La interfaz se puede apreciar en la figura 80, aquí se debe ingresar el peso mínimo, el número de individuos, punto inicial en x , punto inicial en y , y las generaciones.

El algoritmo genético recibe estos datos e inicia el programa teniendo en cuenta que el peso mínimo es el que se lee en pantalla. Este peso mínimo es el criterio de parada y sólo se detiene el algoritmo cuando se alcanza este valor.

El número de individuos hace referencia a cuantos individuos van a cruzarse y a realizar todo el proceso genético a lo largo de las generaciones.

El número de generaciones es el número de veces que los individuos van a ser sometidos a las diferentes operaciones genéticas del algoritmo. Como el único criterio de parada es el peso mínimo, puede suceder que dentro de ese número de generaciones no se llegue al valor de parada, para lo cual se decidió que se inicializa todo otra vez a cero y se vuelve a recalcular desde el principio. Esto fue necesario dado que si la población inicial es muy mala se necesitaría muchas generaciones para obtener el resultado esperado. Este programa se le llamo Freyja dada la relación de la diosa griega de la reproducción con las operaciones trabajadas en el algoritmo. La interfaz es la que aparece en la figura 84. Ejemplos de las trayectorias obtenidas se pueden apreciar en las figuras 85 y 86.

Figura 84. Interfaz gráfica programa de planeación de trayectorias usando algoritmos genéticos.



Figura 85. Ejemplo de trayectoria obtenida con freyja.

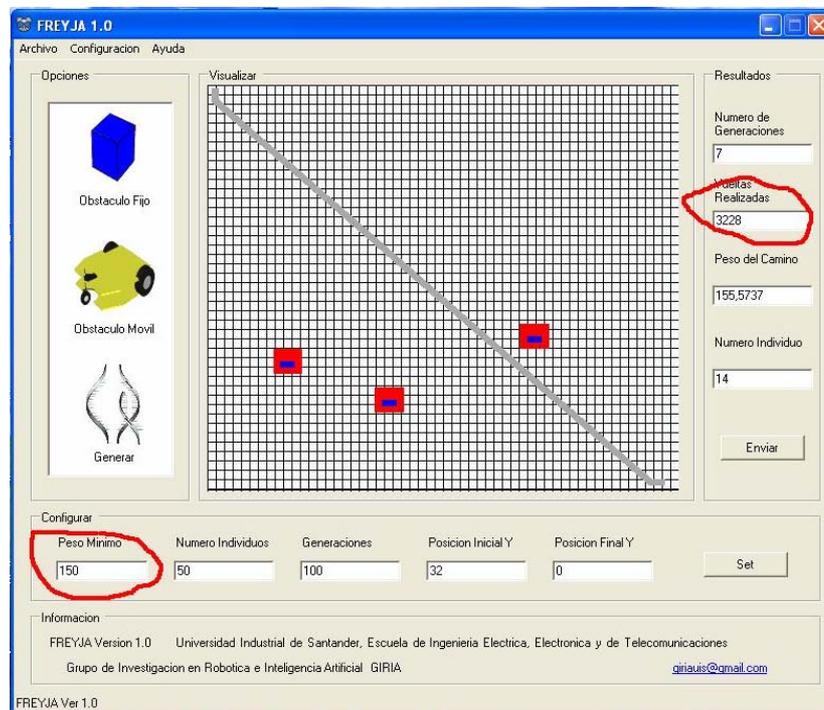
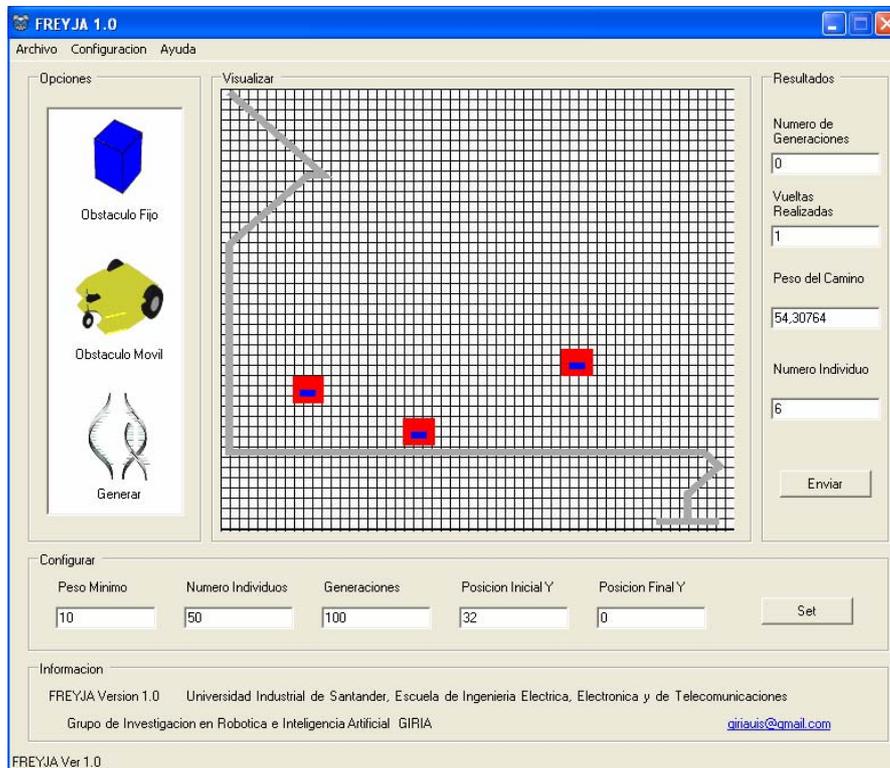


Figura 86. Ejemplo de trayectoria al ejecutar Freyja.

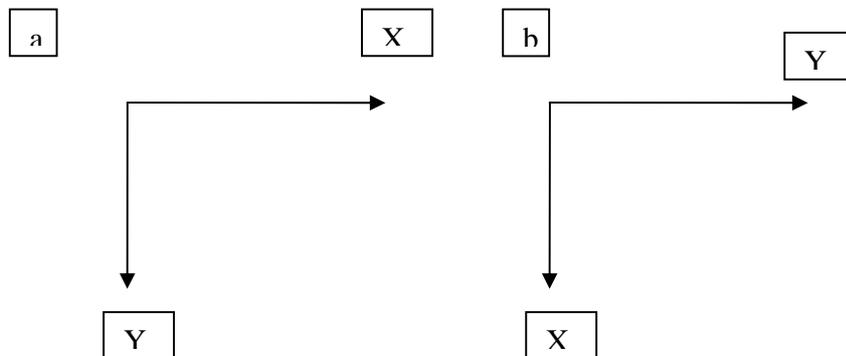


Este algoritmo genético ejecuta trayectorias que van de un punto inicial obtenido de la interfaz a un punto de llegada predefinido evitando los obstáculos y considerando zonas de riesgo alrededor de cada obstáculo. Las trayectorias generadas siempre son diferentes dada la aleatoriedad de la creación de la población inicial, lo cual conduce a obtener caminos válidos en mayor o menor tiempo.

5.4.3 Sistemas de referencia

En la figura 87-a se muestra el sistema de referencia usado en el programa de visión artificial, y en la figura 87-b esta el marco de referencia usado en el algoritmo de campos de potencial. Como puede apreciarse la orientación es diferente, por lo cual fue necesario realizar estos cambios mediante software.

Figura 87. Sistemas de referencia



El origen de coordenadas es posible variarse en el programa fénix de visión artificial. Para cada estrategia desarrollada se trabajó un sistema de referencia variable.

El origen en el programa de campos de potencial se ubica en la esquina de la parte superior izquierda de la pista. Este punto se ubicó dentro del area obtenida por la cámara en el punto: $x=103$, $y=110$. Para algoritmos genéticos el origen se ubicó en la esquina inferior izquierda de la pista.

6. OBSERVACIONES Y CONCLUSIONES

El resultado final del proyecto es una aplicación en Microsoft Visual C++ 6.0, una aplicación en Visual C#.NET y una aplicación en Matlab, con las cuales se puede realizar la localización de los objetos de interés en una escena de condiciones controladas. Como resultado se obtiene la posición para objetos móviles y la posición para obstáculos, generándose trayectorias libre de colisiones con algoritmos genéticos y campos de potencial.

Para realizar la localización de objetos en una zona de interés se utilizan tres algoritmos basados respectivamente en gradientes espacial y temporal, correlación y color. Además se realizó un algoritmo de adquisición que permite capturar de forma eficiente las imágenes necesarias, utilizando una tarjeta digitalizadora DFG/Compress PIRANHA.

Se realizó la configuración de la tarjeta DFG/COMPRESS PIRANHA para adquirir video e imágenes independientes. De esta forma la etapa de adquisición puede interactuar automáticamente con la etapa de procesamiento, de manera que el análisis de movimiento en las imágenes se realiza una vez cada una de ellas es capturada. La etapa de adquisición puede ser utilizada en diferentes aplicaciones, debido al tipo de programación orientado a objetos que se utilizó al implementar FÉNIX, el cual permite modificar y ampliar la aplicación sin alterar las demás funciones. De esta manera, se puede crear nuevas funciones para otras tareas diferentes.

El algoritmo basado en gradientes implementado en este proyecto es una nueva estrategia que se plantea para detectar los objetos en movimiento y avanzar en una futuras investigaciones sobre el flujo óptico[12] con el cual se pueden realizar mediciones mas precisas sobre el movimiento tales como velocidad y aceleración . En comparación con otros métodos para la detección del movimiento como la

extracción del fondo y la diferencia de dos imágenes, el algoritmo basado en gradientes combina información temporal y espacial lo cual solucionó los inconvenientes de ruido en la detección de los objetos en movimiento cuando se encuentran en la escena elementos que no son de interés en la detección.

La etapa de segmentación basada en los modelos de color HLS y xyY, es bastante confiable y robusta gracias al ambiente parcialmente controlado, donde el fondo de la escena tiene un color diferente de los utilizados en el reconocimiento de los objetos móviles y los obstáculos. Además, el color de los obstáculos es diferente al color de los objetos móviles, por lo tanto la identificación entre estos dos tipos de objetos (móviles y obstáculos) se hace fácil. Estos modelos de color tienen la ventaja de que la información cromática se determina en dos componentes, por lo cual el proceso de segmentación con estos modelos se realiza en un menor tiempo respecto a utilizar el modelo RGB, en el cual la información del color se encuentra en sus tres componentes.

Debido a que se trabajó con imágenes de dimensiones 640x480 píxeles sin algún tipo de compresión, la mayoría de las operaciones entre imágenes fueron bastante básicas como operaciones lógicas, sumas, restas y umbralizaciones, con el fin de no consumir demasiado tiempo en procesamiento. Esto se evidenció en la prueba del tiempo de latencia, donde para los tres algoritmos implementados el tiempo de procesamiento es inferior a 1 segundo.

Para el algoritmo basado en correlación, se logró cumplir las restricciones de bajo tiempo en procesamiento y efectividad en la localización, ajustando el nivel de aceptación (valor con el cual se acepta o rechaza un resultado), el tamaño del modelo patrón y el área de búsqueda. Hay que aclarar que a pesar que la función de correlación en imágenes necesita un gran número de multiplicaciones, se logró ganar velocidad a cambio de exactitud en el resultado, aunque manteniendo este último en un nivel bastante aceptable.

Las pruebas realizadas a la aplicación de visión desarrollada arrojaron resultados con un error de posición y de orientación inferiores a 3 cm y 2° respectivamente. Esto se debe a las aberraciones del lente usado. Lo anterior evidencia la confiabilidad en la información de localización obtenida con los tres algoritmos y la utilidad del proceso de calibración para corregir las distorsiones de la cámara utilizada en el proyecto.

Se pueden identificar hasta seis objetos móviles diferentes sin necesidad de marcas específicas, lo cual hace la aplicación atractiva para aplicaciones de robots cooperantes.

La interfaz grafica que acompaña a la aplicación de visión desarrollada es amigable al usuario ya que el entorno es semejante al utilizado en las diferentes herramientas Windows habitualmente usadas. Esto brinda al usuario la posibilidad de manejar las diferentes opciones de la aplicación de una forma sencilla.

La utilidad de los procesos morfológicos en problemas relacionados con tratamiento de imágenes es innegable; sin embargo es importante recalcar que su uso debe ser controlado y orientado a las necesidades del problema a solucionar, considerando que en casos donde se desee conservar los detalles del objeto, la aplicación de un proceso morfológico muy fuerte puede resultar perjudicial y conllevar a un deterioro de la imagen.

La aplicación desarrollada es útil en trabajos relacionados con el tema de planeación de trayectorias de robots móviles en un escenario como el planteado en el presente proyecto, ya que brinda la información confiable de localización de los robots y de los obstáculos presentes en el área de trabajo.

Obtener buenos datos del algoritmo de visión depende en gran medida de realizar una buena calibración.

Se desarrolló una forma alternativa de implementar un método tradicional en robótica, como lo es campos de potencial usando matrices y asignación numérica con excelentes resultados.

Se usó un método de inteligencia artificial para el planeamiento de trayectorias el cual está basado en algoritmos genéticos, con muy buenos resultados.

Dada la complejidad en la programación de un algoritmo genético y de las operaciones realizadas en un campo de potencial, se encontró que la carga computacional es bastante grande, por esto fue necesario usar un computador de uso exclusivo para el trabajo con los algoritmos de control.

El programa basado en campos de potencial es un método sencillo y eficiente. Únicamente es necesario asignar unos pesos a toda el área de trabajo para seguir las celdas con valores numéricos más pequeños y encontrar el punto de llegada.

Es posible ejecutar el programa de planeamiento de trayectorias de campos de potencial para ambientes estáticos o para ambientes dinámicos, lo único que cambia en el algoritmo es la actualización de los datos que puede hacerse luego de obtener los datos con una cámara.

El tamaño de la pista, el número de obstáculos, el punto de inicio y de llegada son ajustables dentro del programa de campos de potencial lo que lo hace más robusto y adaptable a cualquier terreno 2D.

El número de obstáculos depende del tamaño de la pista, además las zonas de riesgo pueden ser aumentadas o disminuidas dependiendo de la forma del obstáculo, tanto para algoritmos genéticos como campos de potencial.

Es posible hacer que cada elemento de la matriz dentro del programa de campos de potencial represente un área dentro de la pista, así como también para el caso de terrenos pequeños hacer que cada coordenada (x,y) corresponda a un elemento de la matriz.

Se encontró que de los dos algoritmos elaborados para el planeamiento de trayectorias, el que se basa en algoritmos genéticos requiere tiempos mayores en la medida que se requiera pesos mínimos más altos.

La velocidad en los algoritmos de planeamiento de trayectorias es afectada también por el número de obstáculos que sea necesario evitar.

Ambos algoritmos toman los datos del computador de el algoritmo de visión usando Ethernet, En el caso de planeamiento off-line sólo se lee al inicio del programa, pero en planeamiento on-line se realiza muestreo de las señales de la pista periódicamente para realizar la actualización de los valores.

El planeamiento on-line requiere de procesadores de alta velocidad, ya que es necesario recalcular periódicamente.

Ambos algoritmos usan radiofrecuencia para el envío de los datos desde el PC a través del puerto serial. Estas señales son recibidas e interpretadas por el microcontrolador y ejecutadas de acuerdo al programa de control que se este ejecutando.

Se realizó una implementación física que puede ser usada para realizar prácticas de laboratorio en materias como robótica e inteligencia artificial.

7. RECOMENDACIONES PARA TRABAJOS FUTUROS

Se recomienda explorar el uso de las funciones de compresión de imágenes de la tarjeta DFG/Compress PIRANHA [15], con el fin de lograr un avance en cuanto a la velocidad de transferencia de datos.

Construir un sistema de iluminación artificial apropiado ayudaría en el proceso de adquisición de las imágenes, segmentación y finalmente en la localización de los objetos de interés en la escena.

Un mayor grado de precisión y exactitud de los resultados podría obtenerse si se investiga a profundidad en los temas relacionados con calibración de cámaras y corrección de las distorsiones producidas por los lentes.

Usar módulos de radiofrecuencia que permitan al algoritmo de control recibir señales de los módulos de ultrasonido para ser incorporados en la hora de la toma de decisiones.

Implementar métodos de planeamiento de trayectorias con otras técnicas de inteligencia artificial, para lograr tiempos de procesamiento menores, así como trayectorias mas suavizadas.

Trabajar estos algoritmos en computadores con mayor velocidad de procesamiento para poder trabajar con móviles más rápidos, así como modificar los algoritmos para que ellos detecten un número no definido de obstáculos y realicen el planeamiento de una manera más óptima .

8. REFERENCIAS BIBLIOGRAFICAS

- [1]. BALLARD, Danna y BROWN, C. Computer Vision [on line]. New Jersey: Prentice Hall, 1982 [citado en mayo de 2004]. disponible en <<http://www.cs.rochester.edu/u/dana/>>
- [2]. CHAPRA, Steven y CANALE, Raymond. Métodos Numéricos para ingenieros. México: McGraw-Hill interamericana editores S.A .1999
- [3]. DE JALON, J. D et al. Aprenda C++ Como Si Estuviera En Primero [on line]. España: Universidad Superior de Ingenieros Industriales de San Sebastián y Universidad de Navarra, 1998[citado en mayo de 2004] disponible en <<http://www.abcdatos.com/tutoriales/tutorial/I2525.html>>
- [4]. GONZALEZ, Rafael y WOODS, Richard. Tratamiento digital de imágenes. Estados Unidos de América: Editorial Addison-Wesley iberoamericana.1996
- [5]. GUREWICH, Nathan y GUREWICH, Ori. Aprendiendo Visual C++ 5 En 21 Días. México: Prentice-Hall Hispanoamericana, S.A. Cuarta edición, 1998.
- [6]. JAIN, Ramesh; RANGACHAR, Kasturi y BRIAN, Shunck. Machine Vision. Singapore: McGraw-Hill Book Co.,1995.
- [7]. KRUGLINSKI, David J; SHEPHERD, George y WINGO, Scot. Programación Avanzada Con Visual C++. España: MCGRAW HILL / INTERAMERICANA DE ESPAÑA. Primera edición en español, 1999.

[8]. KUZINA, T. Y. Investigación e implementación de los algoritmos del campo de visión dinámica por computadora [on line]. México: Universidad de Las Américas Departamento de ingeniería en sistemas computacionales, 2000 [citado en mayo de 2004]. Disponible en http://140.148.3.250/udl_a/servlet/mx.udlap.ict.tales.html.Block?Thesis=42&Type=T

[9]. MAYNERO, Carlos. Desarrollo De Un Sistema De Captura Y Análisis De Imágenes Para La Determinación Del Color Del Esmalte De Dentales. Aplicación A La Confección De Fundas Dentales[on line]. España: Universidad de Valencia, 2002[citado en julio de 2004]. Disponible en <http://www.maynero.com/carlos/proyecto/>

[10]. MATROX IMAGING LIBRARY (MIL). User Guide. Versión 5.1. 1998. Guía en formato digital.

[11]. MATROX IMAGING LIBRARY (MIL). Command Reference. Versión 5.1. 1998. Guía en formato digital.

[12]. PAJARES, Gonzalo y DE LA CRUZ, Jesús. Visión Por Computador: Imágenes Digitales Y Aplicaciones. México: Alfaomega, 2002.

[13]. PEREGRINA, Darío. Seguimiento de objetos por medio de visión activa [on line]. México: Instituto Nacional de Astrofísica, Óptica y Electrónica, 2002[citado en junio de 2004]. Disponible en <http://ccc.inaoep.mx/~labvision/tesis.htm>

[14]. RUSS John. The Image Processing Handbook. Boca Raton, FL. CRC PRESS, IEEE PRESS. 2002.

[15]. THE IMAGING SOURCE DFG / COMPRESS. Users Manual. Versión 1.1 US Edition, 2000. Guía en formato digital.

[16] COOPERATIVE STRATEGY USING DYNAMIC ROLE ASSIGNMENT AND POTENTIAL FIELDS PATH PLANNING. Paul A. Vallejos, Javier Ruiz-del-Solar, Alan Duvost. *Proceedings of the 1st IEEE Latin American Robotics Symposium – LARS 2004, Mexico city, Mexico. October 28 – 29, 2004.*

[17] A BEHAVIOR ARCHITECTURE FOR AUTONOMOUS MOBILE ROBOTS BASED ON POTENTIAL FIELDS. IN: 8TH INTERNATIONAL WORKSHOP ON ROBOCUP 2004 (ROBOT WORLD CUP SOCCER GAMES AND CONFERENCES), Lecture Notes in Artificial Intelligence. Springer, im Erscheinen. Laue, T., Röfer, T. (2005).

[18] Anibal Ollero Baturone, Alfaomega, Marcombo, 2001. ROBÓTICA, MANIPULADORES Y ROBOTS MÓVILES

[19] J. Borenstein , Y. Koren . THE VECTOR FIELD HISTOGRAM-FAST OBSTACLE AVOIDANCE FOR MOBILE ROBOTS. IEEE Journal of Robotics and Automation Vol 7, No 3, June 1991, pp 278-288.

[20] MÉTODO DE APRENDIZAJE SIMPLE PARA NAVEGACIÓN DE MINIROBOTS RODANTES. Gustavo Ramirez. DYNA, Año 70, Nro 138, pp 59-66, Medellín, Marzo de 2003, ISSN 0012-7353.

[21] Kazuo Sugihara and John Smith. "GENETIC ALGORITHMS FOR ADAPTIVE PLANNING OF PATH AND TRAJECTORY OF A MOBILE ROBOT IN 2D TERRAINS". 1999.

[22] Holland, John H. "ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS", University of Michigan Press, 1975, 211 p.

[23] Koza, John R. "GENETIC PROGRAMMING. ON THE PROGRAMMING OF COMPUTERS BY MEANS OF NATURAL SELECTION",The MIT Press, 1992, 819 p.

[24] Goldberg, David E. "GENETIC ALGORITHMS IN SEARCH, OPTIMIZATION, AND MACHINE LEARNING", Addison-Wesley Publishing Company, 1989, 412 p.

[25] Koza, John, "GENETIC PROGRAMMING II : Automatic Discovery of Reusable Programs", The MIT Press, 1992, 746p.

ANEXO A

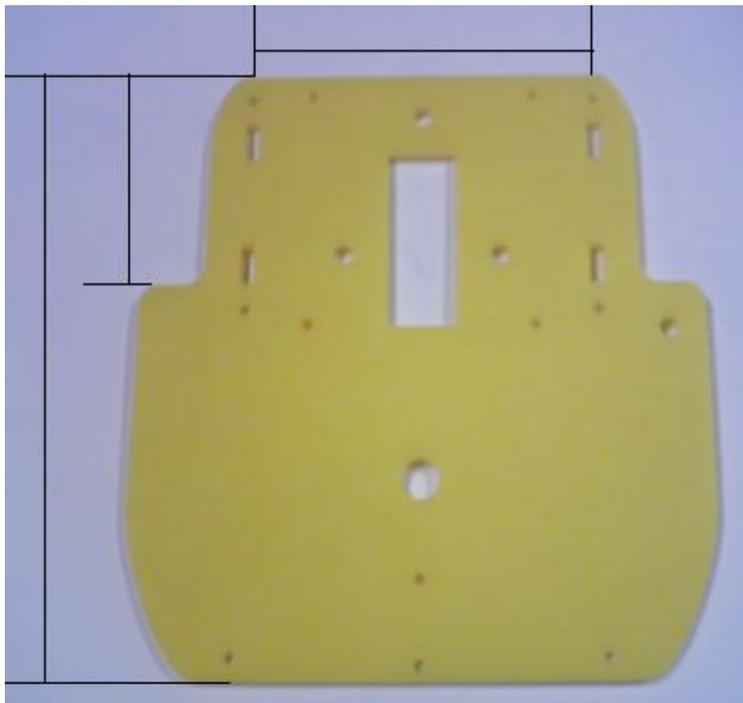
CONSTRUCCIÓN DEL ROBOT MÓVIL

Para el desarrollo del proyecto, se trabajó con unos kits de robotica móvil de la compañía Linxmotion. Estos kits vienen de fabrica por piezas para ensamblar, por lo que fue necesario realizar el ensamble y algunas modificaciones, las cuales se comentarán a lo largo de este capítulo.

A.1 ENSAMBLE GENERAL

El chasis de ambos móviles esta construido por dos placas en acrílico. La forma y dimensiones de las placas se pueden observar en la **figura A.1**.

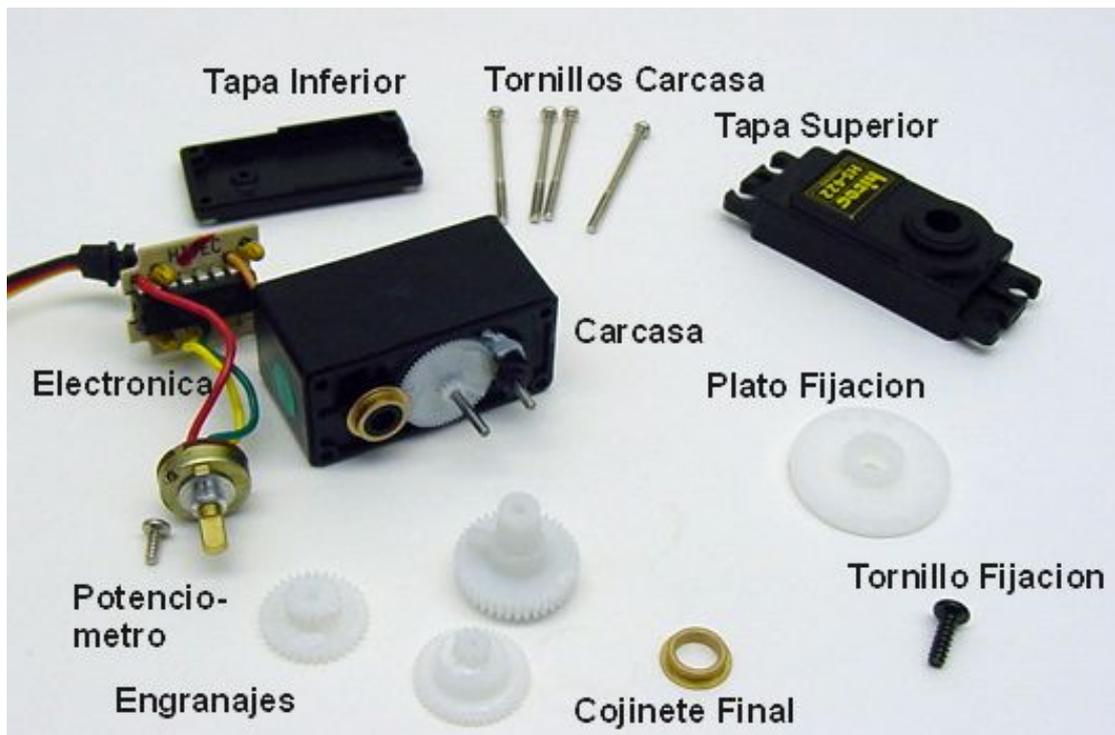
Figura A.1 Placas del chasis de los robots móviles



Para la propulsión del móvil se utilizaron dos servomotores HITEC HS-422 acoplando uno a cada rueda de tracción. Estos servomotores garantizan un alto

torque, necesario para la tracción del móvil sobre la pista de madera. En la figura A.2 se muestran las partes que componen al servomotor.

Figura A.2 Componentes del servomotor.

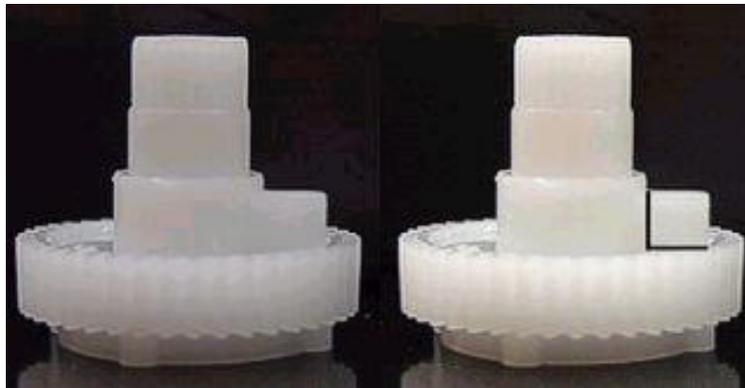


Tomada de superrobotica.com

El servomotor posee en su engranaje principal un tope limitador que no le permite girar más de los 180 grados. Para transformar el servomotor en un dispositivo de rotación continua fue necesario realizar un conjunto de modificaciones que se describe a continuación:

Como primer paso se desarma el servomotor localizando su engranaje principal, luego se procede a remover el tope limitador. Tal como se observa en la **figura A.3**.

Figura A.3 Detalles del tope a cortar.



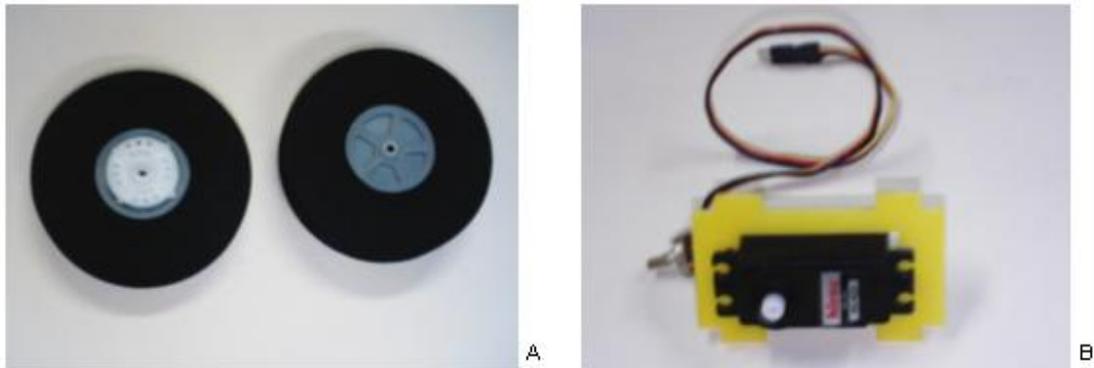
Una vez retirado el tope se debe desacoplar el potenciómetro del eje principal. Existen dos maneras de efectuar este procedimiento. La primera consiste en reemplazar el potenciómetro por otro de igual valor pero que carezca de eje. La otra opción consiste en dejar el potenciómetro original por fuera de la caja conservando sus conexiones. De esta manera es posible fijar la posición neutral enviando el pulso de dicha posición y variando el potenciómetro hasta que el servomotor deje de girar.

Una vez terminada la modificación de los servomotores que van en las llantas se procede a realizar el acople al chasis. Las placas están ubicadas paralelamente una sobre la otra, dando soporte a los motores que están ajustados a una lamina perpendicular a ellas.

Al rotor de los servomotores esta ajustada una rueda de neopreno indicada para el uso en robots. Esta rueda se acopla directamente sobre el plato del servomotor por medio de un adhesivo de doble cara y se atornilla al rotor del servomotor, proporcionando una gran tracción gracias a que esta realizada con neopreno puro. Además, de tracción, esta rueda ofrece un movimiento suave y silencioso gracias

al efecto de almohadilla que amortigua la irregularidades. Las dimensiones de la rueda son de 76 x 19 mm. En la figura A.4 se muestra la rueda por su cara interior y exterior, en la grafica A y en la grafica B podemos observar el rotor del servo.

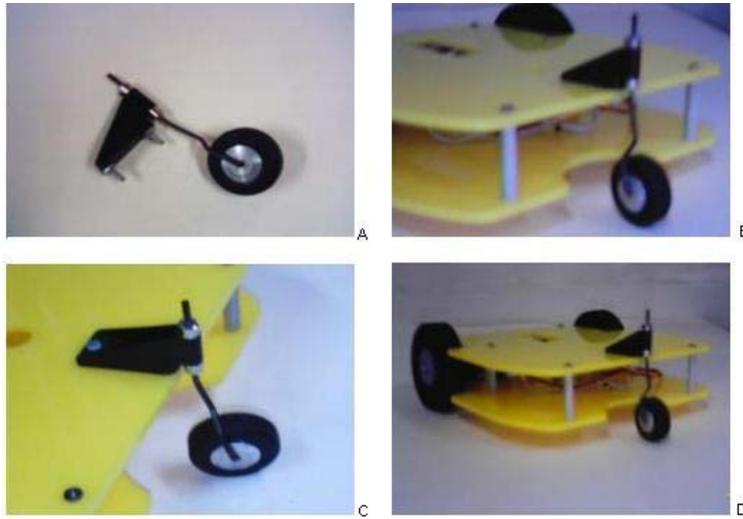
Figura A.4 Rueda de neopreno y servomotor



El chasis posee una rueda giratoria comúnmente llamada rueda loca, la cual es de goma y permite al móvil realizar giros de forma suave y elegante, dependiendo de la tracción dada por las otras dos ruedas.

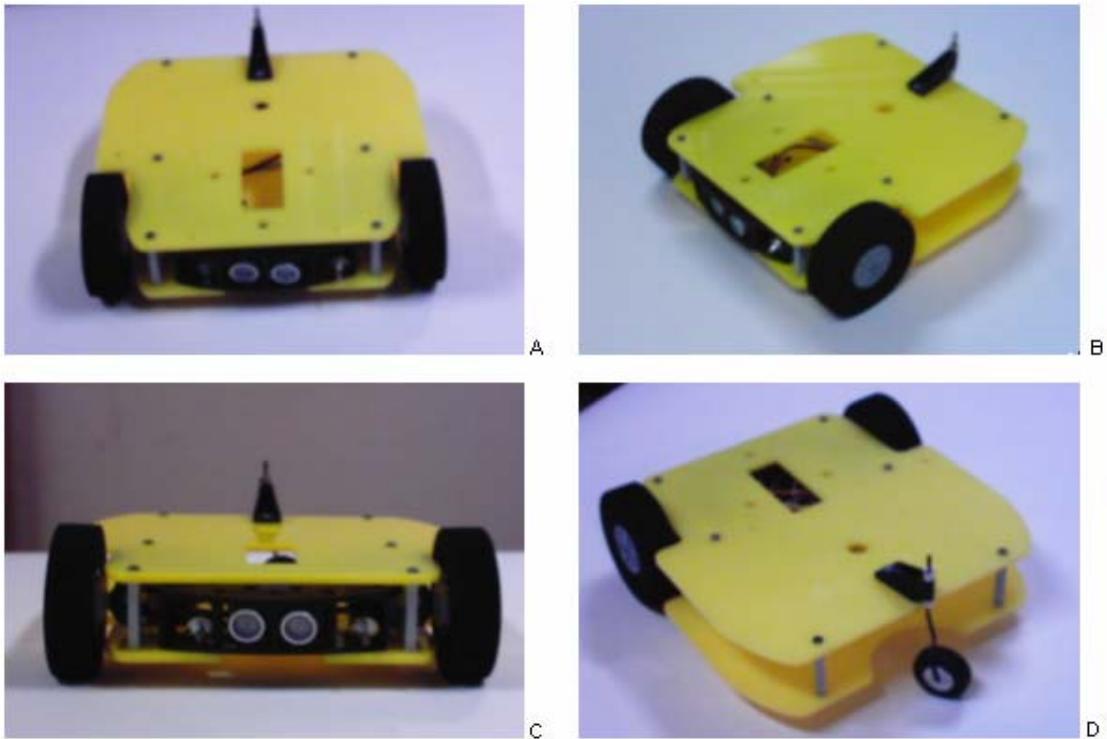
La rueda se atornilla al chasis por medio de un soporte y tiene anillos de fijación que permiten ajustar la altura de trabajo deseada, en este caso, mantener el chasis horizontalmente. En la figura A.5 se puede observar la rueda giratoria con los elementos que la componen.

Figura A.5 Rueda giratoria y su ubicación en el chasis.



El resultado final de la estructura del robot móvil es el que se aprecia en la **figura A.6**

Figura A.6 Móvil completo



En la parte delantera de cada móvil se instaló un módulo de ultrasonido que está atornillado por medio de un soporte al chasis, esto con el fin de realizar la detección de obstáculos. En el caso del robot móvil que va a funcionar como obstáculo, el módulo de ultrasonido es el único sensor que posee para realizar el movimiento sobre la pista, de esta manera se busca hacer que el obstáculo posea movimiento totalmente aleatorio. Para el robot móvil que va a ejecutar la trayectoria, el módulo de ultrasonido sirve como soporte en caso de alguna falla en el sistema que no permita al sistema de control y de visión dar una información rápida del estado actual de la pista.

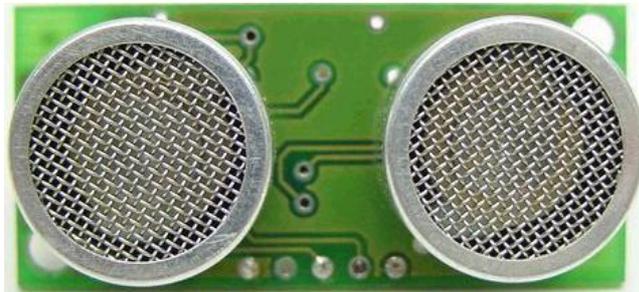
A.2. MODULO DE ULTRASONIDO

El término ultrasonidos designa en principio a los sonidos de alta frecuencia, aunque en realidad corresponde a los sonidos que se encuentran justo por encima de la banda que es capaz de captar el oído humano. El intervalo de frecuencia que corresponde a los ultrasonidos va desde 20 Khz. hasta 500 KHz. Los sonidos a estas frecuencias son más direccionales que los sonidos de frecuencias bajas.

La determinación de la distancia se logra al enviar trenes de pulsos con periodos muy cortos y medir el tiempo transcurrido desde la emisión hasta la recepción del eco. Este tiempo es proporcional a la distancia recorrida por las ondas de presión ultrasónicas.

El modulo de ultrasonido utilizado en los móviles es el SRF04 que se observa en la **figura A.7**.

Figura A.7 Modulo de Ultrasonido SRF04



Este sensor **permite** detectar objetos y calcular la distancia a la que se encuentra en un rango de 3 a 300 cm, por medio de un pulso de tensión cuyo ancho varia entre 100us para 3cm y 18ms para 300cm. Para distancias mayores el sensor envía un pulso de 36ms. Este pulso se obtiene como respuesta (eco) a un pulso de disparo de 10us, enviada por el microcontrolador. Al terminar la recepción del pulso de eco se debe esperar 10ms antes de enviar el siguiente pulso de disparo.

Para su funcionamiento se debe excitar el emisor con una señal adecuada en amplitud y frecuencia de tal modo que este produzca una señal ultrasónica que será enviada al medio externo.

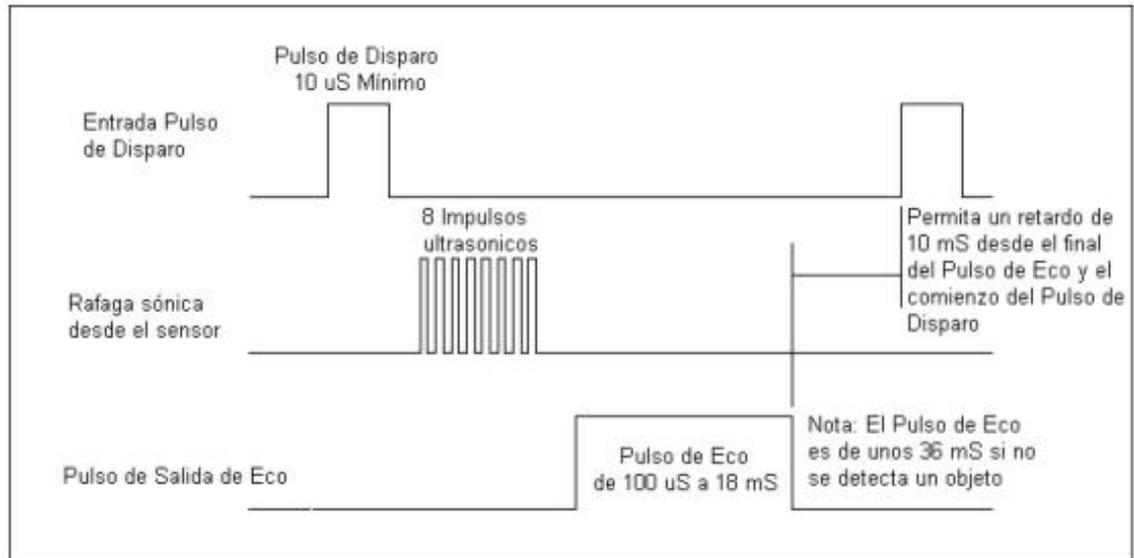
Este modulo contiene toda la electrónica encargada de realizar el proceso de emisión, recepción y medición. Requiere cuatro conexiones para su funcionamiento, dos de ellas son la alimentación del circuito, (0 Volt y +5 Volt dc) y las dos restantes son las conexiones de las señales. El esquema se observa en la **figura A.8.**

Figura A.8. Vista posterior del modulo del ultrasonido SRF04



El diagrama de tiempos del módulo SFRO4 se muestra en la figura A.9.

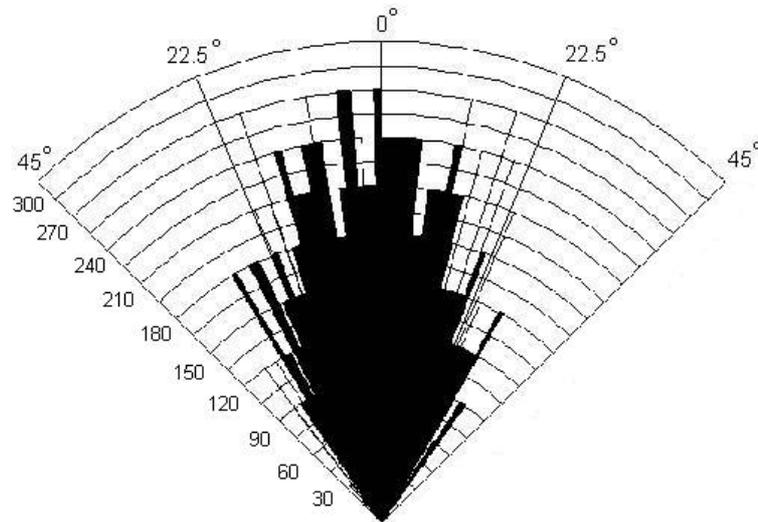
Figura A.9 Diagrama de Tiempos del SRF04



El pulso de 100 μs como mínimo en la salida de eco se debe a que el circuito es desconectado durante este tiempo para evitar el ruido del pulso inicial.

Este modulo de ultrasonido tiene un Rango efectivo de aproximadamente unos 30° como se puede observar en la figura A.10.

Figura A.10 Rango efectivo del SRF04.



Tomada de SupeRobotica.com

La caracterización realizada a este módulo se puede consultar en el capítulo de pruebas.

ANEXO B

PRINCIPALES CARACTERÍSTICAS DE LA TARJETA DIGITALIZADORA DFG/COMPRESS PIRANHA Y DE LA CÁMARA DFK 50H13

B.1 TARJETA DIGITALIZADORA DFG/COMPRESS PIRANHA

La DFG/Compress es una tarjeta de compresión de video de 32 bit PCI, adaptable a multimedia, CTV y aplicaciones médicas. Todas las convencionales fuentes de video con S-VHS (YC) y video compuesto están soportadas por ésta tarjeta de compresión. Las cámaras pueden ser conectadas usando las cuatro entradas de la tarjeta. La transferencia de datos de imagen al PC o a la tarjeta gráfica VGA se hace usando transferencia DMA en burst mode. Usando ésta técnica la CPU no es sobrecargada y de esta manera el PC opera normalmente.

El flujo de imágenes en vivo de salida en el monitor puede ser guardado al disco duro como una secuencia de imágenes AVI en una determinada resolución (máximo 768 x 576) y un número de colores (máximo 32 bit). Adicionalmente, información hardware, como los datos actuales y el tiempo, pueden ser guardados en la imagen de video. Una vez está información ha sido insertada en el flujo de video, no puede ser removida posteriormente. La señal de video sale de un conector de video S-Video (compuesto). Videos previamente grabados en disco duro y el flujo de video en vivo desde una cámara conectada a la tarjeta pueden salir de este conector.

Alternativamente, los datos de video pueden ser escritos a la VGA (y posteriormente al monitor del PC). Para esta funcionalidad, los drivers de la DFG/Compress usan la interfaz DirectDraw de Microsoft.

B.1.1 CARACTERÍSTICAS

- 4 fuentes monocromáticas / 4 fuentes de color compuesto o 4 fuentes Y/C pueden ser conectadas (multiplexadas).
- Soporte multi-estándar (CCIR, PAL; NTSC; RS170, SECAM).
- 4 entradas de disparo.
- 8 entradas y 8 salidas digitales (TTL) – opcionalmente circuitería acoplada ópticamente.
- Soporta distintas resoluciones (representación de la imagen, máximo 768 x 576).
- Conversor de color de Y/C (YUV) a RGB en tiempo real.
- Soporta formatos de verdadero color, RGB de 24 bit y RGB de 32 bit.
- Máxima resolución de 768 x 576 a 50 Hz (PAL / CCIR) y 640 x 480 a 60 Hz (NTSC / RS170).
- Soporte overlay por medio de la interfaz DirectDraw de Microsoft.
- Escalado por hardware de la imagen en vivo.
- EEPROM para escribir y leer datos.
- Interfaz programable para Windows NT 4.0.
- Capacidad de guardar secuencias de video a archivos AVI.
- Compatible con PCI 2.1.
- Transferencia directa de datos a la VGA y a la memoria del sistema.
- Interface maestra PCI DMA.

B.1.2 REQUERIMIENTOS DEL SISTEMA

Los siguientes requerimientos son necesarios para que la DFG/Compress puede ser usada:

- PC con procesador de 100 MHz o mayor.

- Compatibilidad con PCI 2.1.
- Un slot PCI libre de 32 bit.
- Por lo menos 32 MB de RAM (recomendado 64 MB para Windows NT)-
- Windows NT 4.0.

Adicionalmente se recomienda lo siguiente:

- Disco duro SCSI para guardar secuencias de video.
- Monitor externo para verificar el video grabado y la salida de video.

B.1.3 TARJETA VGA

Se recomienda una tarjeta VGA con por lo menos 2 MB de DRAM para aplicaciones monocromáticas y 4 MB de memoria para aplicaciones a color. Solamente con estas especificaciones, se puede lograr resultados con la máxima resolución y número de colores. Es importante que el driver de DirectDraw de la VGA esté instalado. A pesar de que es posible visualizar imágenes sin DirectDraw, existen muchas limitaciones a la frecuencia de repetición de imágenes y al desplegar el overlay. Los mejores resultados pueden ser obtenidos usando tarjetas con chipset ATI (RagePro).

B.1.4 INSTALACIÓN HARDWARE

A continuación se describe como instalar la DFG/Compress en el PC. Antes de la instalación, se debe aislar el PC desconectando el conector de potencia del enchufe en la pared. Para descargar cualquier carga estática almacenada, se debe tocar la cubierta del PC. Luego de esto, cualquier carga estática guardada es

seguramente descargada. La instalación de la DFG/Compress es muy fácil, se debe seguir los pasos mostrados a continuación:

1. Apague el computador y desconecte el cable de potencia. En computadores nuevos con tarjeta madre ATX siempre que el computador se apaga, el bus PCI puede aun estar energizado, por lo cual al insertar la DFG/Compress en la tarjeta madre lleva a la destrucción de ambas tarjetas.
2. Abra el computador quitando los tornillos.
3. Quite el metal del slot PCI el cual no está en uso.
4. Inserte a DFG/Compress en el slot PCI disponible en el computador. Es importante no tocar algún componente en la DFG/Compress.
5. Ajuste la tapa del slot de la tarjeta con un tornillo.
6. Asegúrese que la tarjeta ha sido correctamente instalada en el slot.
7. Tape de nuevo el computador.
8. Conecte la entrada de video.

Se pueden conectar hasta cuatro entradas (compuesta o S-Video) a la DFG/Compress, usando un conector DSUB de 44 pines.

B.1.5 ADQUISICIÓN DE VIDEO

Hasta cuatro fuentes de video pueden ser conectadas a la DFG/Compress. Las entradas pueden ser S-VHS (S-Video / YC, Hi8) o fuentes de video compuesto (VHS, Video8, FBAS, NTSC, RS170, CCIR). LA conexión física entre la fuente de video y la tarjeta se hace con un cable opcional a la salida de un conector DSUB de 44 pines en un lado y un conector BNC (para video compuesto) o conectores Hosiden de 4 pines (S-VHS) en el otro lado.

Resolución de color: La adquisición y compresión de video se hace en un formato de verdadero color YUC 4:2:2. La visualización en la VGA puede ser en diferentes modos: 15 bit, 16 bit (alto color), verdadero color de 24 bit (16.7 millones de colores) o en modo de verdadero color de 32 bit.

Entrada de video: 4 entradas externas: S-Video / YC (S-VHS, Hi8) o compuesto.

Frecuencia de scan de video: PAL/SECAM: 14.75 MHz, NTSC: 12,27 MHz.

Conversión A/D: PAL: 1/50 s (20 ms) por campo, NTSC: 1/60 s (16.7 ms) por campo.

Resolución geométrica: PAL: 768 x 576 píxel en 2 campos, NTSC: 640 x 480 píxel en 2 campos.

B.1.6 ADQUISICIÓN DE IMAGEN CON UNA CÁMARA:

Si se desea adquirir datos de video con solo una cámara, no es necesario usar el conector DSUB de 44 pines. En este caso la DFG/Compress tiene un conector adicional BNC. Los datos desde la cámara pueden ser transferidos en tiempo real (25 imágenes/seg en PAL/SECAM y 30 imágenes/seg en NTSC), por medio de la memoria FIFO, directamente al disco duro del PC. En éste modo de operación, el multiplexor interno no se activará.

B.1.7 SALIDA DE VIDEO

El DFG/Compress tiene dos salidas de video de 50Hz/60Hz que son independientes entre si. Si una fuente 50 Hz SVHS de video se conecta a una

de las entradas de la tarjeta, exactamente la misma señal está disponible en sus salidas. En otras palabras, usando la tarjeta de este modo no se deteriora la señal de video en lo absoluto. Usando la salida de video 1, una de las cuatro entradas de video puede ser colocada directamente a la salida, desviándose así del proceso de digitalización de la tarjeta. La salida de la entrada seleccionada de video no tiene influencia en ninguna grabación de video, la cual puede estar corriendo al mismo tiempo. En la salida de video 2 se fija el video corriente proveniente de una cámara o la señal de video sin comprimir, dependiendo de cual modo se encuentra seleccionado en la tarjeta.

Modo grabación (recording): Entre la entrada de video y la salida de video la señal es análoga. Cualquier revestimiento presente no es enviado a la salida de video.

Modo cinta (playback): La imagen descomprimida es una secuencia de video AVI. Cualquier revestimiento presente esta contenido en la imagen descomprimida. Adicionalmente la salida de video puede ser vista en un monitor VGA.

Resolución en color: La descompresión y la salida de la señal de video 2 es siempre en YUV 4:2:2 formato de color verdadero. El despliegue en la tarjeta VGA, puede ser en 15 bits, 16 bits (alto color), 24 bits en modo de verdadero color (16.7 millones de colores) o 32 bits en modo de verdadero color.

B.1.8 ORGANIZACIÓN DE LA MEMORIA PARA IMÁGENES SIN COMPRIMIR

Las imágenes sin comprimir pueden capturarse singularmente o como una secuencia. La adquisición de la secuencia no tiene nada que ver con el almacenamiento en memoria de la secuencia de video comprimido. Durante la adquisición, la secuencia de imágenes es almacenada en una o más memorias de

imagen (memorias). Antes de esto puede elegirse un lugar de almacenamiento, la memoria de imagen debe ser primero asignada con el driver e insertada dentro de la lista de la secuencia. La preparación de la memoria de imagen toma lugar con el driver, cuando se ha localizado físicamente y de forma lineal (una al lado de la otra) las posiciones de memoria para la secuencia que se almacena. Note que una memoria con estas características puede ser definida solamente cuando su sistema esta iniciando. Solo una limitada parte de la memoria principal esta disponible para la asignación de la memoria de imagen y no puede ser modificada en su tamaño mientras el sistema se encuentra en funcionamiento.

B.1.9 INSTALACION DEL SOFTWARE

La DFG/Compress SDK (funciones API) y el driver de video para Windows están contenidos ambos en el DLL *piranha.dll*. Durante la instalación el driver de video para Windows se introduce en el registro del directorio correspondiente. Este es el mismo durante la instalación de video para Windows y el SDK.

Aviso Importante:

Para instalar la tarjeta DFG/Compress bajo Windows NT, los derechos del administrador se necesitan. En el caso de instalar la tarjeta por primera vez o realizar un cambio de tarjeta en su equipo.

Numero de serial:

El numero serial de la tarjeta DFG/Compress esta impreso en un sticker blanco sobre la tarjeta del circuito y además esta almacenado en la EEPROM en la tarjeta. Cada tarjeta DFG/Compress tiene un único número de serial el cual puede ser mostrado con el programa *lcbInfo.exe*

Instalación por primera vez:

Windows NT 4.0:

Para instalar el software simplemente inserte el disco 1 en su unidad de disquetes y corra el SETUP, el menú de este programa es guiado y auto explicativo. La rutina de instalación copia los drivers en los siguientes directorios:

```
icbvcd32.sys  a  \<windir>\system32\drivers
icbmem32.sys a  \<windir>\system32\drivers
piranha.dll   a  \<windir>\system32
icbhook.dll   a  \<windir>\system32
piranha.ini   a  \<windir>
icblogic.rbf  a  \<windir>
```

La estructura del directorio es creada de la siguiente forma:

<Install_dir>\Piranha\Program	Programas de ayuda
\Piranha\Samples\Exe	ejemplos de programas
\Piranha\Samples\Source	código de los programas de ejemplo
\Piranha\Help	información adicional

(<windir> es el directorio de Windows. <Install_dir> es el directorio en el cual el software de la DFG/Compress es instalado.)

Windows 9X:

Bajo Windows 9x la DFG/Compress será reconocida automáticamente como un nuevo dispositivo multimedia, preguntando así por el driver correspondiente para la instalación, cuando pregunte, por favor inserte el disco en la unidad de

disquetes. El driver de la DFG/Compress se encuentra en este disco, entonces simplemente siga las instrucciones en pantalla y cuando pregunte que hacer, reinicie su sistema, esto es necesario, ya que el driver será reconocido solamente cuando su sistema se encuentre iniciando. Cuando su sistema esta corriendo de nuevo, inserte el disco y ejecute el programa *SETUP* y de nuevo siga las instrucciones en pantalla. Durante la instalación los siguientes archivos son copiados en su PC:

icbvcd32.vxd	a	\<windir>\system
piranha.dll	a	\<windir>\system
icbhook.dll	a	\<windir>\system
piranha.ini	a	\<windir>
icblogic.rbf	a	\<windir>

Una vez su sistema reinicia, inicie el programa de demostración. Si su PC no reinicia correctamente o se bloquea mientras esta iniciando, no se preocupe! , simplemente oprima el botón de reset que se encuentra en el exterior de su PC, apague y espere 60 segundos y de nuevo prenda su PC, Windows deberá comenzar correctamente.

B.1.10 MODOS DE SALIDA DE LA TARJETA DFG/COMPRESS

Modo de mapa de bits (BMP/DIB):

En este modo el video capturado de la DFG/Compress es almacenado en la memoria principal del PC y luego copiado a la tarjeta VGA, aunque de esta forma, la tasa de actualización en pantalla es más lenta, Sin embargo la mayoría de computadoras modernas están en capacidad de desplegar imágenes en tiempo real usando la DFG/Compress. El modo mapa de bits es particularmente ventajoso,

debido a la alta compatibilidad que tiene con todas las tarjetas VGA y directo acceso de los datos de imagen a la memoria principal. Las funciones de cubierta (mezclado en una imagen de texto y gráficos) tienen que ser programadas por el usuario. Como Windows toma el control del despliegue de las imágenes, la imagen en vivo puede ser parcial o totalmente cubierta por muchas otras ventanas y cajas de dialogo.

DirectDraw Primary Surface Mode:

En este modo los datos de la imagen son escritos en el área visible de la tarjeta VGA. Brinda el máximo desempeño de 25 imágenes por segundo con la DFG/Compress. En este caso no es posible usar cubiertas (mezclado en una imagen de texto y gráficos) debido a que los datos de cubierta deben ser constantemente sobrescritos en el siguiente frame lo cual nunca permitiría su visualización. En contraste con muchas otras tarjetas de compresión, en la DFG/Compress es posible posicionar ventanas, cajas de dialogo y otras aplicaciones de Windows sobre la ventana que contiene la imagen en vivo de la DFG/Compress sin perturbarla en lo absoluto, la tarjeta de compresión puede llevar a cabo esto, no desplegando automáticamente la imagen en vivo bajo las ventanas que la cubren y manteniendo una alta frecuencia de actualización.

DirectDraw Overlay Surface Mode:

En este modo, se puede lograr simultáneamente, la velocidad de salida del modo DirectDraw Primary Surface Mode y el despliegue de los datos de cubierta. Esta es la ventaja más grande, ya que las tarjetas VGA las implementan a un costo muy bajo. La imagen de video es digitalizada a un área no visible de la tarjeta VGA. El chip de la VGA permite desplegar la imagen de video o el contenido original de la ventana de salida en la que se despliegan los datos, exactamente cualquiera de estos dos modos se usa dependiendo del color que se haya programado. Definiendo el tipo de color y usando este para dibujar en la ventana

de salida, la imagen solo podrá ser mostrada en las partes donde el tipo de color elegido sea usado, naturalmente si toda la ventana esta llena solamente del tipo de color elegido, la imagen se visualizara completamente. Ahora, textos y gráficos usados en la cubierta que no usen el tipo de color elegido permanecerán disponibles, esto se conoce como *cubierta no-destructiva*. El mezclado de la cubierta es realizado por el chip de la VGA y apenas se requieren algunos ciclos de CPU. Este modo no es soportado por todos los chips VGA y la DFG/Compress solo soporta modos de 15 y 16 bits en los VGA.

B.1.11 COLOR Y FORMATOS DE ALMACENAMIENTO

Cada uno de los formatos de color soportado por la tarjeta DFG/Compress usa diferentes formatos con lo cual se almacenan las imágenes. Como se muestra en la siguiente tabla:

Format	DWORD	Pixel Data [31:0]			
		Byte 3 [31:24]	Byte 2 [23:16]	Byte 1 [15:8]	Byte 0 [7:0]
RGB32	dw0	0	R0	G0	B0
	dw1	0	R1	G1	B1
RGB24	dw0	B1	R0	G0	B0
	dw1	G2	B2	R1	G1
	dw2	R3	G3	B3	R2
RGB16	dw0	{R1[15:11], G1[10:5], B1[4:0]}*		{R0[15:11], G0[10:5], B0[4:0]}*	
RGB15	dw0	{0, R1[14:10], G1[9:5], B1[4:0]}*		{0, R0[14:10], G0[9:5], B0[4:0]}*	

* Cuando una imagen esta en formato RGB15 o RGB16 los bits superiores al bit interno 8 son usados en los colores R G y B.

A. B.2 CÁMARA DFK 50H13

A continuación se muestran las principales características de la cámara DFK 50H13.

Formato de la señal: NTSC

Sensor de imagen: 1/3" interlineado CCD.

Imagen efectiva: 768 (H) x 494 (V).

Frecuencia Horizontal: 15.734 kHz

Frecuencia Vertical: 59.94 Hz

Mínima iluminación: 1 lux.

Radio S/N: mas de 48dB (AGC=OFF).

γ : 0.45.

Balance blanco: automático / manual.

Montaje de lente: CS o C (adptador).

AGC: ON / OFF, seleccionable.

Salida de Video: VBS 1.0 Vp-p a 75 Ω , Y/C.

I/O serial: RS232C.

Alimienación: 10 ~ 12 V DC.

Potencia consumida: 180 mA.

Temperatura de operación: -10°C ~ +40°C.

Dimensiones: 51mm (ancho) X 51 mm (alto) X 55.5 mm (profundidad).

Peso: 190g.

ANEXO C

INSTALACIÓN E INTERFASE GRAFICA DE “FENIX”

FENIX es una aplicación software desarrollada en *Microsoft Visual C++ 6.0*, la cual tiene como objetivo el análisis dinámico de imágenes en una escena particular.

C.1 INSTALACIÓN

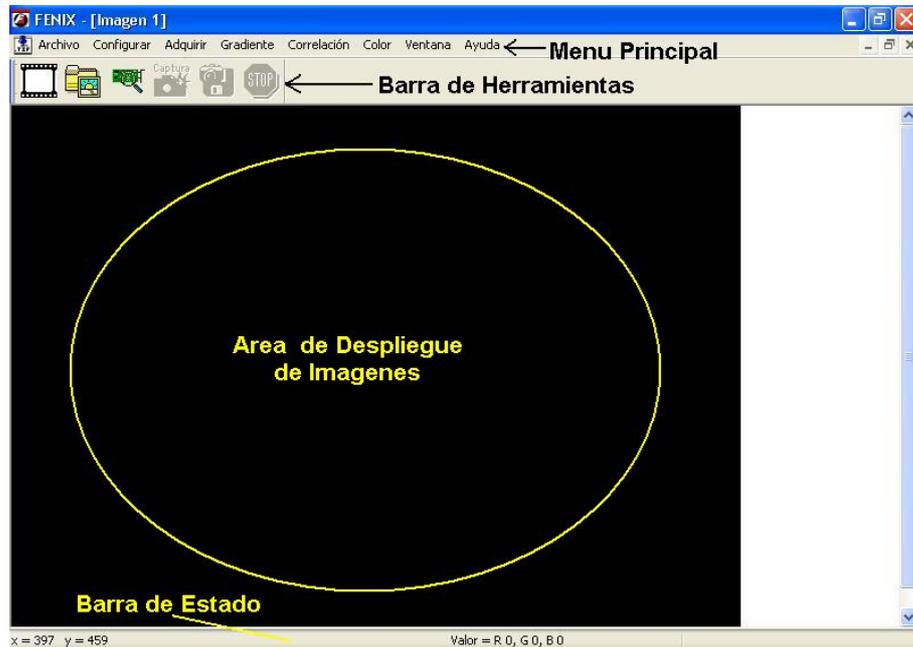
A continuación se describen los pasos a seguir para una correcta instalación de la aplicación FENIX.

1. Instale correctamente la tarjeta digitalizadora PIRANHA. En el anexo B se encuentra de forma detallada la instalación de la misma.
2. Copie la carpeta *Fenix* con todos sus componentes, en algún lugar del disco duro del PC.
3. Ejecute el archivo *fenix.exe*, el cual se encuentra dentro de la carpeta *Fenix* copiada a su PC.

C.2 INTERFASE GRAFICA

En la figura C.1 se observa la ventana principal de FENIX.

Figura C.1. Ventana principal de FENIX

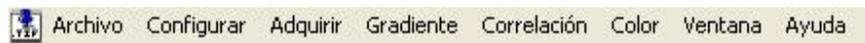


La ventana principal de FÉNIX tiene diferentes características: en la parte superior se encuentra el menú principal, del cual se puede seleccionar todas las funciones de la aplicación. También se muestra la barra de herramientas que tiene acceso directo a algunas funciones. En la parte central se observa el área para desplegar imágenes. Finalmente en la parte inferior se muestra la barra de estado.

C.2.1 MENÚ PRINCIPAL

En la figura C.2 se observan las diferentes opciones del menú principal las cuales son: archivo, configurar, adquirir, gradiente, correlación, color, ventana y ayuda.

Figura C.2. Menú principal



C.2.1.1 Archivo

En la figura C.3 se aprecian las opciones del ítem archivo.

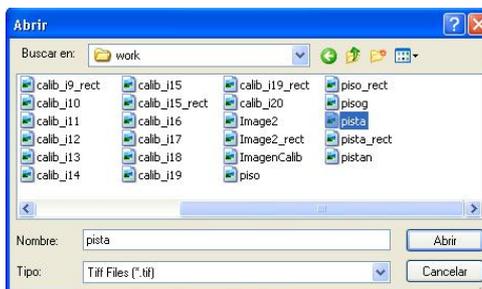
Figura C.3. Ítem archivo



Nuevo: Crea una nueva área (ventana) para desplegar una imagen.

Abrir: Carga una imagen en formato tiff ,desde una archivo existente, por medio del dialogo *abrir* de Windows (figura C.4).

Figura C.4. Abrir Imagen



Cerrar: Cierra una imagen activa.

Salir: Cierra la aplicación en su totalidad.

C.2.1.2 Ítem Configurar

Permite fijar los parámetros necesarios para la correcta ejecución de los algoritmos implementados en la aplicación. Consta de las siguientes opciones: adquisición, video, calibración, modelo de color y resultados. En la figura C.5 se muestra la manera como se despliegan la opciones en pantalla.

Figura C.5. Ítem configurar



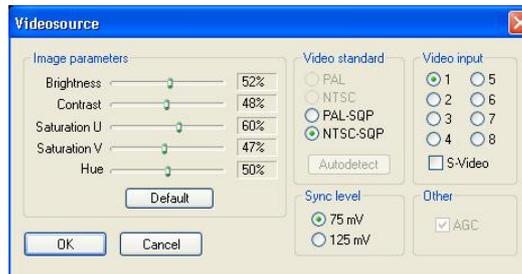
Adquisición: permite seleccionar por medio de una ventana de dialogo (figura C.6) la tarjeta con la cual se va a realizar el proceso de digitalización. Para el proyecto desarrollado la única tarjeta configurada fue la PIRANHA, aunque se dejan las opciones de otras tarjetas en el dialogo para posteriores mejoras de la aplicación.

Figura C.6. Dialogo de selección de la tarjeta digitalizadora



Video: permite manipular por medio de una ventana de dialogo (figura C.7) en la tarjeta piranha las propiedades de video como contraste, brillo, selección del canal de video, estándar de video etc.

Figura C.7. Dialogo de propiedades del video de la tarjeta digitalizadora PIRANHA



Calibración: En esta opción (figura C.8) se fijan los parámetros intrínsecos obtenidos en el proceso de calibración de la cámara (figura C.9) y el origen de coordenadas del mundo real en la imagen. (figura C.10).

Figura C.8. Ítem calibración



En la figura C.9 se observa el dialogo de entrada para los parámetros intrínsecos de calibración. En este dialogo también se introduce a la aplicación el factor de escala necesario para la conversión de píxeles a centímetros.

Figura C.9. Dialogo de entrada para los parámetros de calibración

Parámetros de Calibración

Fenix

Parámetros Intrínsecos

Longitud Focal
fc1 1077.18144 fc2 1074.90424

Punto Principal
cc1 328.63616 cc2 194.06424

Coefficientes de Distorsión
kc1 -0.3163 kc2 -0.31778 kc3 0.00186 kc4 0.00094

Factor de Escala
0.2564102 cm/pixel

Aceptar Cancelar

en la figura C.10 se muestra el dialogo de entrada para las coordenadas en píxeles del origen del sistema de referencia del mundo real en la imagen.

Figura C.10. Dialogo de entrada para el origen de coordenadas

Sistema de Referencia

Fenix

Coordenadas (píxeles)

Xo 320 Yo 240

Aceptar Cancelar

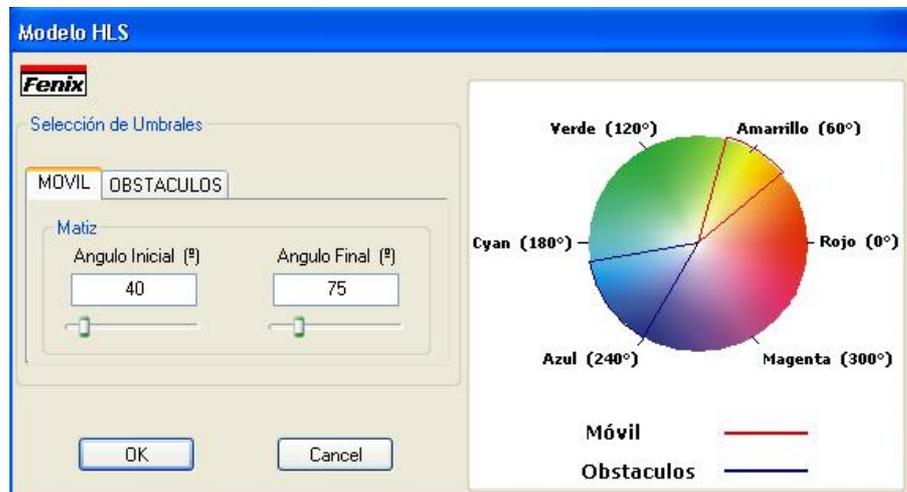
Modelo de color: Con este ítem (figura C.11) se puede seleccionar y configurar el modelo de color a utilizar. Se tienen las opciones de modelos HLS y xyY

Figura C.11. Ítem modelo de color



En la figura C.12 se observa el dialogo de configuración del modelo de color HLS. En la parte izquierda se configura el matiz del color a localizar en los móviles y los obstáculos. En la parte derecha se muestra de forma grafica y dinámica los cambios realizados en los valores de configuración del modelo

Figura C.12. Dialogo de configuración del modelo de color HLS.

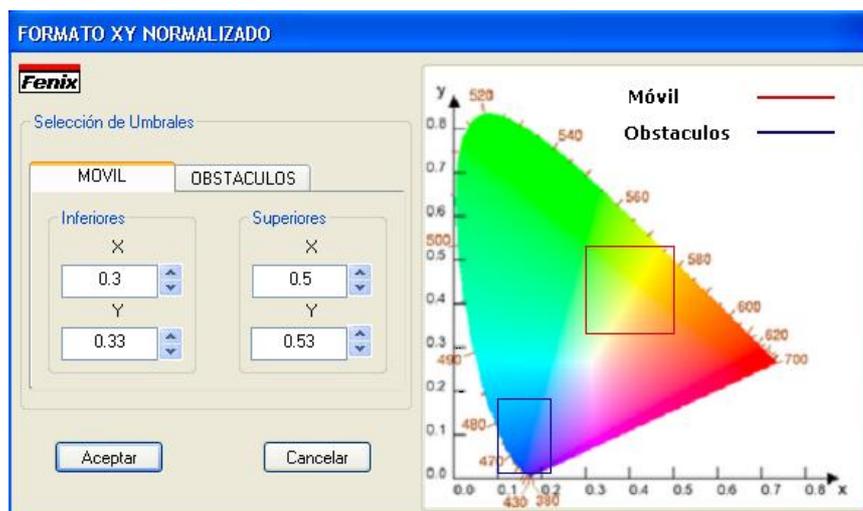


Al seleccionar el modelo de color xyY, se despliega la ventana de dialogo mostrada en la figura C.13, la cual tiene dos partes principales.

La primera se ubica a la izquierda y contiene la información de los umbrales utilizados para la segmentación de los objetos móviles y de los obstáculos. Para las componentes x e y del modelo xyY se tiene un umbral inferior y un umbral superior.

La segunda está ubicada a la derecha y contiene información visual del color con el cual se desea segmentar cada tipo de objetos de la escena. Para esto se tiene una imagen de fondo del modelo de color xyY y sobre ésta se dibujan dos rectángulos. Uno de ellos encierra el color deseado para los objetos móviles y el otro encierra el color para los obstáculos.

Figura C.13. Dialogo para seleccionar umbrales en el modelo de color xyY



Resultados: este ítem tiene dos opciones: la primera se refiere a la unidad en la cual se desea desplegar la información de posición para los objetos de interés en la escena. Esta unidad de medida puede ser píxeles o centímetros (figura C.14).

Figura C.14. Selección de la unidad de medida para la posición de los objetos de Interés



La segunda indica el tipo de referencia (fija o móvil) para calcular la orientación de los objetos móviles (figura C.15).

Figura C.15. Selección de la referencia para calcular la orientación



La aplicación por defecto selecciona medidas de posición en centímetros y una referencia móvil para la orientación.

C.2.1.3 Ítem Adquirir

Este ítem tiene tres opciones: adquirir video, capturar un frame y guardar un frame desplegado, como se muestra a continuación:

Figura C.16. Ítem adquirir



Video: Esta opción permite adquirir video en una ventana de dialogo como se muestra en la figura C.17.

Figura C.17. Ventana de dialogo para desplegar el video



Frame: captura un frame (imagen) y lo despliega en la parte central de la ventana principal de la aplicación (figura C.18).

Guardar: guarda una imagen en formato tiff, por medio de la ventana “*guardar*” de windows (figura C.19).

Figura C.18. Frame capturado y desplegado

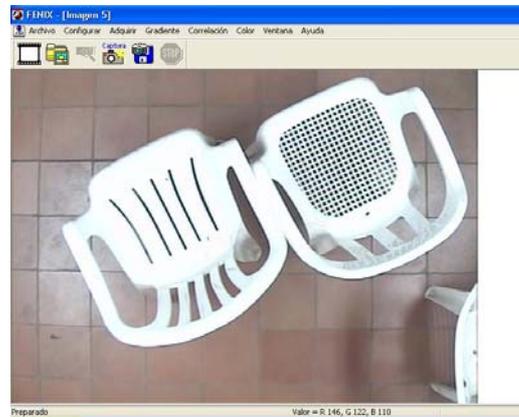


Figura C.19. Ventana para guardar una imagen.



C.2.1.4 Ítems gradiente, correlación y color

Estos ítems son los encargados del inicio de ejecución de los tres principales algoritmos implementados por medio de la opción *calcular* (figura C.20). Cabe resaltar que la aplicación solo deja ejecutar solo un algoritmo a la vez.

Figura C.20. Gradiente, correlación y color



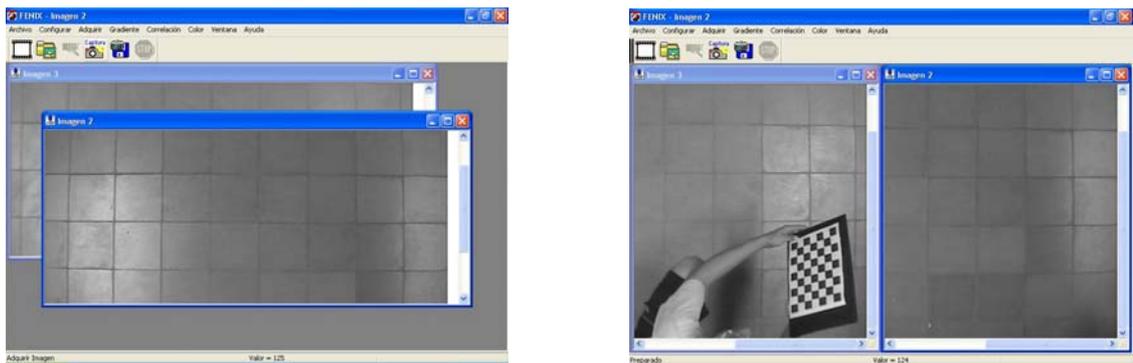
C.2.1.5 Ítem Ventana

Con este ítem se organizan las imágenes que se encuentran desplegadas en diferentes ventanas en forma de cascada (figura C.22-a) o de mosaico (figura C.22-b). También se muestra un lista con las imágenes actualmente desplegadas.

Figura C.21. Ítem ventana



Figura C.22 Organizar imágenes en forma de a) cascada b) mosaico.



a

b

C.2.2 BARRA DE HERRAMIENTAS

Esta barra está compuesta de iconos con los cuales tienen acceso directo a algunas de las funciones descritas anteriormente (figura C.23).

Figura C.23. Barra de herramientas



A continuación se relaciona cada icono con la función que ejecuta.



Despliega una nueva imagen en la cual se puede adquirir una imagen



Abre una imagen ya existente.



Abre el dialogo de selección de la tarjeta digitalizadora.



Captura una imagen desde la cámara



Guarda una imagen

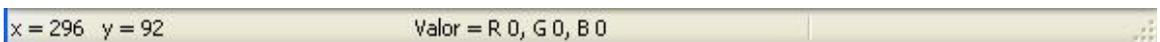


Detiene el procesamiento del algoritmo en ejecución.

C.2.3 BARRA DE ESTADO

La barra de estado muestra dos tipos de información: a la izquierda las coordenadas por donde pasa el cursor del mouse sobre la imagen y la derecha el valor RGB del píxel en dichas coordenadas, como se muestra en la figura C.24.

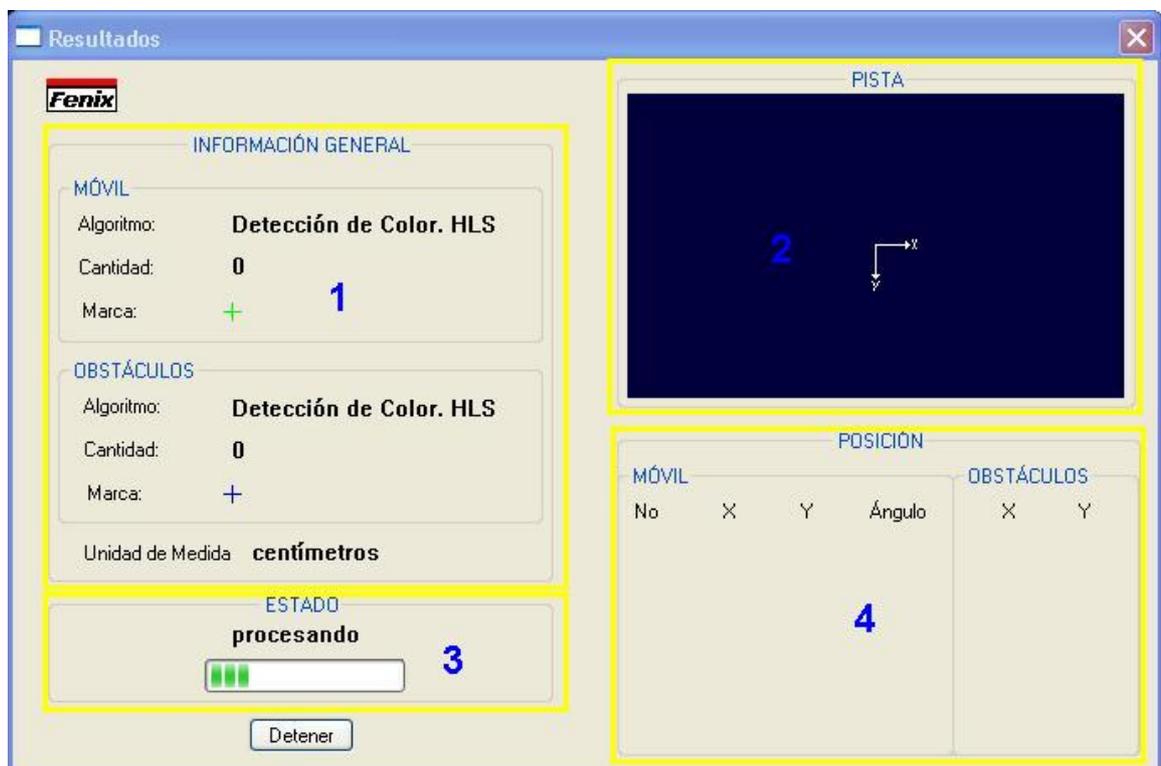
Figura C.24. Barra de estado



C.2.4 DIALOGO DE RESULTADOS

Es donde se despliega la información final de la aplicación FENIX. Este dialogo se divide en cuatro zonas como se muestra en la figura C.25.

Figura C.25 Dialogo de resultados



Zona 1. Información General: Muestra la información del algoritmo utilizado, cantidad de objetos de interés en la escena y unidad de medida para la posición.

Zona 2. Pista: área en la cual se simula la escena real. Los objetos de interés en la escena se simbolizan por medio de marcas.

Zona 3. Estado: muestra el estado de ejecución de los algoritmos (procesando / detenido).

Zona 4. Localización: se muestra la información de posición en la unidad de medida seleccionada, para los objetos móviles y obstáculos en la escena. Además el ángulo de orientación respecto al eje x para cada uno de los objetos móviles.