

**METODOLOGÍA PARA UN SISTEMAS DE VISIÓN ARTIFICIAL
APOYADO CON REALIDAD AUMENTADA EN BASE A MARCADORES
PARA UN ROBOT MÓVIL**

Ing. Carlos Alberto Valencia Hernández

MAESTRÍA EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL

FACULTAD DE INGENIERÍAS

INSTITUTO TECNOLÓGICO METROPOLITANO

MEDELLÍN

2015

**METODOLOGÍA PARA SISTEMAS DE VISIÓN ARTIFICIAL APOYADO
CON REALIDAD AUMENTADA EN BASE A MARCADORES PARA UN
ROBOT MÓVIL**

Ing. Carlos Alberto Valencia Hernández

Tesis de grado para optar al título de: Magíster en Automatización y Control Industrial

Director: PhD. Alejandro Restrepo Martínez
Codirector: MSc. Nelson David Muñoz

MAESTRÍA EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL

FACULTAD DE INGENIERÍAS

INSTITUTO TECNOLÓGICO METROPOLITANO

MEDELLÍN

2015

DEDICATORIA

A mi familia, porque creyeron en mí y porque me sacaron adelante, dándome ejemplos dignos de superación y entrega; porque en gran parte, gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí fue lo que me hizo llegar hasta el final. Va por ustedes, por lo que valen, porque admiro su fortaleza y por lo que han hecho de mí. Gracias por haber fomentado en mí el deseo de superación y el anhelo de triunfo en la vida.

CONTENIDO

INTRODUCCIÓN	1
1. MARCO TEÓRICO Y ESTADO DEL ARTE	5
1.1. Marco teórico	5
1.1.1. Robótica	5
1.1.2. Algoritmo de navegación	8
1.1.3. Visión artificial	12
1.1.4. Ángulo de visión	13
1.1.5. Calibración de parámetros intrínsecos de las cámaras	13
1.1.6. Realidad aumentada	17
1.2. Estado del arte	21
1.2.1. Visión artificial en robótica móvil	21
1.2.2. Realidad aumentada	23
1.2.3. Marcadores en Realidad Aumentada	29
1.2.4. Aplicaciones de la realidad aumentada	32
1.2.5. Realidad aumentada y navegación de robots móviles	33
2. MÉTODOS Y PROCEDIMIENTOS	38
2.1. Arquitectura mecánica y electrónica de la plataforma robótica	39
2.1.1. Descripción del sistema locomotor	40
2.1.2. Sistemas electrónicos de la plataforma	42
2.1.3. Sistemas de potencia	43
2.1.4. Circuito de control y comunicaciones	44
2.1.5. Suministro de potencia	45
2.2. Caracterización de sistema de visión	45
2.2.1. Evaluación de marcadores	47
2.2.2. Calibración de parámetros intrínsecos y extrínsecos de la cámara	49
2.2.3. Estimación de error de la realidad aumentada con respecto al mundo real	58
2.2.4. Ángulos y distancias de detección de marcadores.	62

2.2.5.	Volumen de visión	72
2.3.	Pruebas de realidad aumentada y marcadores	74
2.4.	Parámetros de iluminación	85
2.5.	Ambiente de pruebas	92
2.6.	Algoritmos de navegación	97
2.7.	Estrategia de integración de la realidad aumentada con marcadores, el sistema de visión artificial y el algoritmo de navegación	107
2.8.	Evaluación de trayectorias	109
3.	RESULTADOS Y ANÁLISIS	110
3.1.	Pruebas de navegación	110
3.1.1.	Análisis estadístico de trayectorias	136
3.2.	Planteamiento de metodología	139
4.	CONCLUSIÓN	143
5.	TRABAJOS FUTUROS	145
	REFERENCIAS	146
	ANEXOS	155

LISTA DE TABLAS

Tabla 1. Elementos de cumplimiento para cada objetivo específico	3
Tabla 2. Recorrido tecnológico y los eventos críticos en la historia de la Realidad Aumentada..	24
Tabla 3. Cuadro que refleja los comentarios hechos en esta sección de la presente tesis.....	35
Tabla 4. Resultados obtenidos con los marcadores de apoyo.....	49
Tabla 5. Parámetros intrínsecos cámara del robot	52
Tabla 6. Vértices del marcador Pica en las dos imágenes de la Figura 29 (i, j) y los vértices del recuadro que se quiere re proyectar sobre ellas (X, Y.).....	54
Tabla 7. Puntos de referencia de las imágenes para re proyección de cubo	57
Tabla 8. Parámetros intrínsecos cámara del robot	61
Tabla 9. Parámetros intrínsecos de RA	61
Tabla 10. Ángulos mínimos de detección de marcadores	64
Tabla 11. Distancias de operación empleando <i>BuildAR</i>	67
Tabla 12. Distancias de operación empleando <i>BuildAR Pro</i>	68
Tabla 13. Acercamiento del robot al marcador desde 1mtr hasta 17 cm.....	69
Tabla 14. Distancia y estados de objetos 3D.....	71
Tabla 15. Dimensiones de los marcos de visión obtenidos	73
Tabla 16. Evaluación de software de RA	76
Tabla 17. Descriptores de forma, grados mínimos de detección y distancias máximas de detección discriminada por marcadores.....	80
Tabla 18. Centroides de imágenes internas de los marcadores	82
Tabla 19. Distancias máximas de detección para prueba de solidez	84
Tabla 20. Comportamientos de marcador frente a variaciones de iluminación en fuentes superiores	89
Tabla 21. Comportamientos de marcador frente a variaciones de iluminación en fuentes superiores y reflector	90
Tabla 22. Elementos utilizados en las pruebas de navegación	93

Tabla 23. Criterios de evaluación de trayectorias algoritmo reactivo.....	118
Tabla 24. Criterios de evaluación de trayectorias algoritmo de navegación por campos potenciales	123
Tabla 25. Criterios de evaluación de trayectorias algoritmo reactivo en ambiente industrial	126
Tabla 26. Criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente industrial.....	129
Tabla 27. Criterios de evaluación de trayectorias algoritmo reactivo en ambiente rectangular..	132
Tabla 28. Criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente rectangular	135
Tabla 29. Prueba de hipótesis con algoritmo reactivo.....	137
Tabla 30. Prueba de hipótesis con algoritmo de campos potenciales	138

LISTA DE FIGURAS

Figura 1. Diagrama de flujo algoritmo de navegación reactivo básico	9
Figura 2. Representación de interacción de campos potenciales artificiales	10
Figura 3. Modelo de fuerzas de campos potenciales	11
Figura 4. Niveles de procesamiento de imágenes	12
Figura 5. Ángulo de visión	13
Figura 6. Patrón de calibración de tableo de ajedrez	14
Figura 7. Taxonomía de Milgram	18
Figura 8. Clasificación de la Realidad Aumentada según la técnica de reconocimiento	18
Figura 9. Sistema de Realidad Aumentada. 1) Cámara, 2) Pantalla,3) Elemento Virtual, 4) Ordenador, 5) Marcador, 6) Usuario	19
Figura 10. Proceso general de la Realidad Aumentada	20
Figura 11. Marcador estándar	29
Figura 12. Marcadores binarios	30
Figura 13. Marcadores a color	30
Figura 14. Marcadores jerárquicos	31
Figura 15. Marcadores circulares	31
Figura 16. Flujo de información en el robot	39
Figura 17. a) Oruga simple, b) Oruga, paral lateral y amortiguador vista lateral, c) Orugas montadas en paral frontal con canastillas y amortiguador vista frontal, d) Orugas montadas en paral frontal con canastillas y amortiguador vista superior.	40
Figura 18. a) Chasis vista frontal, b) chasis vista trasera, c) chasis vista superior, d) chasis vista lateral.	41
Figura 19. Plataforma completa	42
Figura 20. Vista superior de la plataforma robótica	43
Figura 21. Diagrama de flujo de sistemas electrónicos de la plataforma robótica.....	45
Figura 22. Ubicación de cámara en el robot	46
Figura 23. Registro de distancia mínima de visión	47

Figura 24. Disposición de marcador central y de apoyo	48
Figura 25. Montaje para la obtención de las imágenes de calibración.....	50
Figura 26. Imágenes de calibración de la cámara frontal del robot	51
Figura 27. Imágenes de referencia para modelación y validación de modelo de la cámara del robot	53
Figura 28. Ejemplo de extracción de coordenadas de vértices del marcador <i>pica</i> y del recuadro que se re proyectará en la imagen	53
Figura 29. Imágenes con re proyección de esquinas	55
Figura 30. Vértices de cubo re proyectado en las coordenadas del mundo	56
Figura 31. Imágenes con proyección de cubo	57
Figura 32. Patrón de calibración real	58
Figura 33. Patrón de calibración con inserción de marcador de RA y delimitación de área elegida para la calibración (encerrada en recuadro rojo)	59
Figura 34. Imágenes de calibración a) imágenes cámara real b) imágenes con RA	60
Figura 35. Escenario y parámetros de medición de ángulos críticos de trabajo con múltiples marcadores 1) Robot, 2) Telón, 3) Marcador, 4) semicírculo que delimita el posicionamiento del robot.....	63
Figura 36. Representación de los ángulos críticos de visión (en rojo) entre los cuales no son detectados los marcadores.....	65
Figura 37. Robot frente a marcador.....	66
Figura 38. Marcadores empleados en la prueba	70
Figura 39. Proceso de medición de cada marco de visión	73
Figura 40. Volumen de visión construido con los marcos de visión obtenidos experimentalmente	74
Figura 41. Gráfica con la evaluación de software de Realidad Aumentada	76
Figura 42. Diagrama de pruebas experimentales de marcadores	77
Figura 43. Disposición de elementos en prueba de escala	78
Figura 44. Descriptores de forma y distancias máximas de detección.....	81

Figura 45. Gráficas de relación de descriptores de forma y ángulos de detección	81
Figura 46. Gráficas de centroides de imágenes internas de marcadores y distancia de desestabilización representada en el radio de las esferas	83
Figura 47. Estructura de pruebas de iluminación	85
Figura 48. Ambiente de prueba con un solo marcador	86
Figura 49. Ambiente de prueba con dos marcadores	87
Figura 50. Disposición de Luz 1 y Luz 2 con respecto al marcador	88
Figura 51. Disposición de los marcadores en el ambiente de pruebas de múltiples marcadores ..	91
Figura 52. Ubicación de las fuentes de iluminación en el ambiente de prueba industrial	92
Figura 53. Vista superior del ambiente de prueba con objetos reales	94
Figura 54. Laboratorio de Mecatrónica, Institución Universitaria Pascual Bravo	95
Figura 55. Máquina llenadora de granos y cortadora industrial	95
Figura 56. Dimensiones de ambiente de prueba industrial	96
Figura 57. Ambiente de prueba para prueba sin corredores	96
Figura 58. Artículos que emplean el algoritmo reactivo como estrategia de navegación	98
Figura 59. Casos de navegación	99
Figura 60. Distancia de detección de obstáculos y su representación en la imagen tomada por la cámara frontal del robot	100
Figura 61. Sección reactiva del algoritmo de navegación reactivo híbrido	101
Figura 62. Sección híbrida del algoritmo de navegación reactivo híbrido	102
Figura 63. Plantilla experimental para algoritmo de campos potenciales	103
Figura 64. Plantilla con tratamiento digital de imágenes para algoritmo de campos potenciales	104
Figura 65. Ejemplo del efecto de un obstáculo sobre el algoritmo de campos potenciales implementando la plantilla obtenida experimentalmente	104
Figura 66. Algoritmo general de campos potenciales aplicado a la plataforma robótica	106
Figura 67. Estrategia de integración de la Realidad Aumentada con marcadores, el sistema de visión artificial y los algoritmos de navegación	108

Figura 68. Diagrama de flujo de extracción de trayectorias del robot en cámara aérea	111
Figura 69. Procesos generales que componen el algoritmo de trazado de trayectorias	112
Figura 70. Ambiente real de pruebas objetos reales (cámara aérea).....	113
Figura 71. Ambiente real de pruebas objetos reales (cámara frontal del robot).....	114
Figura 72. Gráficas de trayectorias de robot con objetos reales empleando el algoritmo de navegación reactivo.....	114
Figura 73. Ambiente de con ubicación de marcadores para la inserción de los objetos virtuales (cámara aérea)	115
Figura 74. Ambiente de pruebas real con los objetos virtuales insertados por la Realidad Aumentada (cámara frontal del robot)	116
Figura 75. Gráficas de trayectorias de robot con objetos insertados por la RA empleando el algoritmo de navegación reactivo	116
Figura 76. Gráfica de resultados empleando algoritmo de navegación reactivo aplicando métricas de evaluación	118
Figura 77. Gráficas de trayectorias de robot con objetos reales y algoritmo de campos potenciales	119
Figura 78. Imágenes del obstáculo cilindro real captadas por la cámara del robot en el ambiente de pruebas	120
Figura 80. Ejemplo de efecto óptico de obstáculos independientes unidos en una masa única aparente	121
Figura 79. Gráficas de trayectorias de robot con objetos insertados por la RA y algoritmo de campos potenciales	122
Figura 81. Gráfica de resultados empleando algoritmo de navegación de campos potenciales aplicando métricas de evaluación.....	123
Figura 82. Gráficas de trayectorias de robot en ambiente industrial con objetos reales empleando algoritmo reactivo	124
Figura 83. Gráficas de trayectorias de robot en ambiente industrial con objetos virtuales empleando algoritmo reactivo.....	125

Figura 84. Gráfica de criterios de evaluación de trayectorias algoritmo reactivo en ambiente industrial.....	126
Figura 85. Gráficas de trayectorias de robot en ambiente industrial con objetos virtuales empleando algoritmo reactivo.....	127
Figura 86. Gráficas de trayectorias de robot en ambiente industrial con objetos virtuales empleando algoritmo de campos potenciales	128
Figura 87. Comparación de trayectorias del robot en ambiente industrial empleando objetos virtuales a) algoritmo reactivo b) algoritmo por campos potenciales	128
Figura 88. Gráfica de criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente industrial.....	129
Figura 89. Gráficas de trayectorias de robot en ambiente rectangular con objetos reales empleando algoritmo reactivo.....	131
Figura 90. Gráficas de trayectorias de robot en ambiente rectangular con objetos virtuales empleando algoritmo reactivo	131
Figura 91. Gráfica de criterios de evaluación de trayectorias algoritmo reactivo en ambiente rectangular	133
Figura 92. Gráficas de trayectorias de robot en ambiente rectangular con objetos reales empleando algoritmo por campos potenciales	134
Figura 93. Gráficas de trayectorias de robot en ambiente rectangular con objetos virtuales empleando algoritmo por campos potenciales	134
Figura 94. Gráfica de criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente rectangular	135
Figura 95. Etapa 1 de la metodología para sistemas de visión artificial apoyadas con Realidad Aumentada con base en marcadores para un robot móvil.....	139
Figura 96. Etapa 2 de la metodología para sistemas de visión artificial apoyadas con Realidad Aumentada en base a marcadores para un robot móvil	140
Figura 97. Etapa 3 de la metodología para sistemas de visión artificial apoyadas con Realidad Aumentada en base a marcadores para un robot móvil (completa).....	141

LISTA DE ECUACIONES

Ecuación 1. Potencial incidente sobre el robot.....	10
Ecuación 2. Fuerza incidente sobre el robot	11
Ecuación 3. Sumatoria de fuerzas en el robot	11
Ecuación 4. Matriz de parámetros intrínsecos KK	15
Ecuación 5. Matriz de parámetros extrínsecos de la cámara	15
Ecuación 6. Transformación de un punto 2D del mundo al plano 2D de la cámara	16
Ecuación 7. Matriz de transformación mundo cámara	16
Ecuación 8. Ecuación de λ	16
Ecuación 9. Matriz de rotación R_t	16
Ecuación 10. Vector de rotación P	16

LISTA DE ANEXOS

Anexo 1. Imágenes de calibración de cámara aérea.....	155
Anexo 2. Tabla de parámetros intrínsecos cámara aérea	155
Anexo 3. Plano esquemático de circuito de comunicaciones del robot	156
Anexo 4. Plano esquemático de sistema de control de motor del robot.....	156
Anexo 5. Plano esquemático de sistema de control del robot.....	157
Anexo 6. Modelo cinemático del robot.....	157
Anexo 7. Código de algoritmo de extracción de fotogramas de videos	158
Anexo 8. Código de algoritmo de identificación del robot y trazado de trayectorias	159
Anexo 9. Código de algoritmo de impresión de trayectorias y medición de distancias en pixeles	161
Anexo 10. Código algoritmo de campos potenciales	163
Anexo 11. Código para el análisis morfológico de marcadores.....	167
Anexo 12. Código algoritmo reactivo.....	169

AGRADECIMIENTOS

A todas las personas que de una u otra forma se vieron envueltas en este proyecto de investigación, especialmente a los directores de tesis Alejandro Restrepo Martínez y Nelson Muñoz, por su paciencia, colaboración y asesoría durante la ejecución de la investigación.

RESUMEN

En la presente investigación, se propone una metodología para integrar sistemas de visión artificial apoyada con realidad aumentada en base a marcadores, para robots móviles, con el fin de evaluar la navegación en ambientes de trabajo, combinando objetos virtuales y reales en un solo escenario, de modo que no sea obligatorio tener los elementos físicos para estudiar el comportamiento de los robots, permitiendo diferentes pruebas en los prototipos, obteniendo la flexibilidad del mundo virtual y la fiabilidad de los ambientes de prueba reales.

Se desarrolló un robot móvil, el cual se desplazaba con un par de orugas simples. Al este, se le implementó un sistema de visión artificial monocular; posteriormente se realizaron pruebas para definir las características de volumen de visión, condiciones de trabajo de distancias y ángulos de visión. Igualmente, se realizó una calibración de la cámara empleando el *toolbox* de calibración Calib del software MATLAB, para caracterizar la óptica del sistema. Con esto, se procedió a seleccionar el software de realidad aumentada y otro software en el cual se implementarían los algoritmos de visión artificial y de navegación.

Los tipos de software seleccionados para este trabajo fueron: para el ambiente de desarrollo *Build AR pro*, para la realidad aumentada; y el software MATLAB para la implementación, tanto de los algoritmos de tratamiento digital de imágenes, de navegación de campos potenciales y reactivo, como de comunicación y control.

Un aspecto que se estudió fue la influencia de las condiciones de iluminación en los marcadores de RA, al igual que se realizó un análisis morfológico de los mismos (envolvente convexa, número de elementos, porcentaje de área convexa y área simple), con el fin de definir la influencia de la forma en los parámetros de detección y plantear unas recomendaciones básicas para la selección y diseño de marcadores para RA. También se tratan los problemas encontrados en los procesos anteriores,

como la pérdida de los marcadores por parte de la cámara del robot y sus implicaciones en la representación del ambiente aumentado, mostrando y sugiriendo soluciones al respecto.

Al final, se realizaron pruebas con dos algoritmos de navegación (reactivo y por campos potenciales artificiales) en tres ambientes de prueba; el primero, simulando un corredor en el cual se trabajó tanto con realidad aumentada como con objetos reales; el segundo, un ambiente industrial real con lo que se analizó la efectividad de la metodología en este tipo de ambientes; y el último, una trayectoria no recta.

Se evaluaron las trayectorias en los ambientes de prueba empleando criterios de desempeño estándar (longitud de trayectoria recorrida, periodos de control y distancia mínima obstáculos), con lo que se pudo demostrar, por medio de la prueba estadística de hipótesis (*t student*), que el comportamiento de un robot en un ambiente de prueba empleando realidad aumentada es estadísticamente similar al presentado en uno real).

Como aportes significativos de esta tesis se resalta la demostración de que los comportamientos del robot, tanto con los objetos reales como los aumentados, son estadísticamente similares. Igualmente, se pudo proponer una forma de calibración de sistemas de realidad aumentada con base en marcadores, empleando patrones virtuales y reales. Por último, la obtención de esta metodología que simplifica y/o estandariza el desarrollo y la validación de algoritmos de navegación, en donde se representen ambientes que fusionen realidad aumentada y lo real.

ABSTRACT

In the present research, there is a proposition of a methodology forwarded to integrate artificial vision systems, supported with amplified reality, based on markets for mobile robots, trying to evaluate its navigation in working environments, mixing both virtual and real objects in only one scenery in a way that possibilities or makes no mandatory, the existence of the objects in order to study the behavior of the robots, allowing different kinds of test in the prototypes, obtaining by this way, the flexibility of the virtual environment, and the trustiness of the actual test environments.

Has been developed a mobile robot, which moves around by a couple of simple caterpillars, equipped with an artificial monocular vision system, lately tested in order to define the characteristics of the vision volume, working conditions of angles of vision and distances; meanwhile, the camera was calibrated by using the calibration toolbox CALIB from the software MATLAB, to characterize the optic of the system, with this results, the following step consisted in selecting the appropriate software of expanded reality and the software to implement the algorithms of artificial vision and navigation.

The selected software were Build art pro for the environment development, for the expanded reality, and the software MATLAB as implementation software for the algorithms of digital treatment of image and also for the navigation of potential fields, reactive and control and communication.

One of the studied aspects, was the influence of light conditions on the RA markers, as was done a morphological analysis of them (surrounding, convex, number of elements, convex area percentile and simple area), trying to define the influence of the form in the parameters of detection, and also proposes a few basic recommendations for the selection and design of RA markers. Are also treated the problems found in the previous process, like the loss of markers from the robot's camera and its implications in the expanded environment representation and suggesting possible solutions about it.

At the end, tests were done with two different navigation algorithms (reactive and by potential artificial fields) in three different testing environments, the first simulating a corridor, in which were worked with expanded reality with real objects, the second an actual industrial environment used to analyze the effectiveness of the methodology in this kind of environments, and the last a not straight trajectory.

The trajectories were evaluated by standard facts of performance (like covered distance, control periods and minimal distance with obstacles.) facts that allows to demonstrate by the statistics test of hypothesis (*t student*), that the behavior of a robot in an environment test using expanded reality is statistically similar to the one in an actual one.

As significant inputs from this thesis, should be stand out, the demonstration of the fact that the robot behavior also in real objects as in expanded ones, are statistically similar; by the same way, was possible to propose a way to calibrate the systems of expanded reality based on markers using virtual and actual standards.

INTRODUCCIÓN

Los ambientes de Realidad Aumentada (RA) se pueden definir como aquellos que mezclan elementos reales y virtuales en un ambiente único. Esta tecnología se ha usado tradicionalmente en los campos del entretenimiento, la medicina, la educación, la publicidad y la robótica (Krevelen & Poelman, 2010).

En el campo de la robótica, la RA se ha empleado tanto en robots manipuladores como en robots móviles. En los primeros, su aplicación más común es la validación y apoyo al diseño de trayectorias y ambientes de trabajo (Kozlov, Macdonald & Wünsche, 2007; Luo & Lai, 2014); en los segundos, en el enriquecimiento de interfaces de usuario en robots tele operados (Mekni & Lemieux, 2014), concentrándose en el mejoramiento de los algoritmos de seguimiento de marcadores y corrección de errores por oclusión (I. Y. H. Chen, MacDonald & Wünsche, 2008).

Volviendo a los robots móviles, la necesidad de contar con todos los componentes físicos de un ambiente de trabajo al momento de desarrollar e implementar robots que naveguen en ambientes estáticos, se convierte en un obstáculo para agilizar dichos procesos al igual que para flexibilizar los ambientes de pruebas. En este trabajo se propondrá una metodología que ayudará a subsanar este problema, integrando ambientes reales y virtuales en un ambiente de realidad aumentada.

El trabajo está dividido esencialmente en cinco secciones: la primera tratará los temas de marco teórico y estado del arte; la segunda mostrará los métodos y procedimientos; la tercera expondrá los resultados y el análisis de las pruebas de navegación; la cuarta presentará la metodología propuesta; y por último, en la quinta, se expondrán las conclusiones de este trabajo.

A continuación se presenta la hipótesis y los objetivos que dan forma a este trabajo:

Hipótesis

Se propondrá una metodología para un sistema de visión artificial apoyada por realidad aumentada en base a marcadores, que permitirá la navegación de un robot en ambientes cambiantes sin utilizar, en algunos casos, los obstáculos físicos.

Objetivo general

Proponer una estrategia de navegación por visión artificial con realidad aumentada en base a marcadores para un robot móvil en ambientes cambiantes.

Objetivos específicos

- Caracterizar el sistema de visión artificial del robot.
- Determinar una estrategia de integración entre la realidad aumentada con marcadores, el sistema de visión artificial y el algoritmo de navegación.
- Evaluar la metodología en un ambiente cambiante controlado con el sistema de realidad aumentada en base a marcadores, utilizando métricas dimensionales.

Alcance y limitaciones

Se debe tener en cuenta que este proyecto está planteado para ambientes *In-door*. Se hace referencia sobre el concepto de ambiente cambiante: que a los obstáculos que interactúan en los escenarios se les puede cambiar su posición espacial, es decir, que aunque el ambiente es estático permite el intercambio espacial de los obstáculos, y que el único que cambia su posición en el transcurso de las pruebas es el robot. Asimismo, se aclara que los obstáculos que se empleará para las pruebas

serán de formas geométricas simples, como cubos y cilindros sólidos, excepto el elemento Poliedro, el cual se describirá más adelante, y se implementará para revisar los límites del sistema y su comportamiento ante objetos con geometrías complejas.

A continuación, se especifica con cuáles elementos se le dio cumplimiento a cada objetivo específico:

Tabla 1. Elementos de cumplimiento para cada objetivo específico

OBJETIVO ESPECÍFICO	ELEMENTO DE CUMPLIMIENTO
Caracterizar el sistema de visión artificial del robot.	<ul style="list-style-type: none"> • Volumen de visión de la cámara • Estimación de parámetros intrínsecos y extrínsecos de la cámara • Ángulos y distancias de detección de marcadores • Selección de software de realidad aumentada • Estimación del error en la representación de objetos reales en la realidad aumentada.
Determinar una estrategia de integración entre la realidad aumentada con marcadores, el sistema de visión artificial y el algoritmo de navegación.	<ul style="list-style-type: none"> • Selección de software de programación del algoritmo de navegación y tratamiento digital de imágenes. • Diseño y evaluación de marcadores. • Elección de canal de comunicación entre el software de realidad aumentada y el software de programación que alberga el algoritmo de navegación y tratamiento digital de imagen. • Establecer tipo de iluminación.

	<ul style="list-style-type: none"> • Diseño e implementación del algoritmo de navegación y tratamiento digital de imágenes.
<p>Evaluar la metodología en un ambiente cambiante controlado con el sistema de realidad aumentada en base a marcadores, utilizando métricas dimensionales.</p>	<ul style="list-style-type: none"> • Evaluación de trayectorias basándose en criterios estándar de evaluación y pruebas de similitud estadísticas (<i>T student</i>).

Fuente: Autor

1. MARCO TEÓRICO Y ESTADO DEL ARTE

El contenido de este capítulo se divide en dos áreas principales: el marco teórico y el estado del arte. En el marco teórico, se verán los temas concernientes a la robótica móvil, los algoritmos de navegación aplicados en este trabajo, la visión artificial y los aspectos técnicos, como son el ángulo de visión y las técnicas de calibración de cámaras. Por último, se mostrarán los conceptos referentes a la Realidad Aumentada.

El segundo tema (estado del arte), se concentró en la búsqueda bibliográfica de proyectos realizados en las áreas de la visión artificial aplicada a la robótica y la Realidad Aumentada haciendo énfasis en los proyectos que integren estas dos tecnologías.

1.1. Marco teórico

1.1.1. Robótica

El *Robot Industries Association* (RIA) define que: «un robot es un manipulador funcional reprogramable, capaz de mover material, piezas, herramientas o dispositivos especializados mediante movimientos variables programados, con el fin de realizar tareas diversas» (Association Robotic Industries, 2015). Este segmento se concentrará en los elementos generales que componen los robots móviles como son: locomotor, electrónico, potencia, control y comunicaciones.

Robots móviles. Se pueden definir como una máquinas automáticas que se pueden desplazar por un ambiente determinado (Martínez-Gómez, et al. 2014). En los años recientes, ha habido un auge de la robótica móvil gracias a los avances en la tecnología empleada para su diseño e implementación. Las áreas de aplicación de estos dispositivos son vastas, solo por citar algunos ejemplos tenemos aplicaciones: pedagógicas, militares, de monitoreo y manipulación de materiales

peligrosos (Bermúdez, 2002), humanitarias como búsqueda y rescate, distribución de correo, limpieza de recintos cerrados, exploración espacial, investigación científica, etc.

Un robot móvil se puede dividir en los siguientes sistemas:

Sistema locomotor. Hay una gran variedad de modos de moverse sobre una superficie sólida; entre los robots, las más comunes son las ruedas, orugas y patas.

Los vehículos de ruedas son, con mucho, los más populares por varias razones prácticas: estos son más sencillos y fáciles de construir, además, la carga que pueden transportar es considerable. Algunos limitantes de estos sistemas son la pérdida de tracción en superficies blandas (Ollero Baturone, 2005) y la limitación de superar obstáculos con altura mayor al radio de sus ruedas. Los robots con ruedas presentan varias configuraciones, las cuales les brindan diferentes características respecto a carga útil, consumo de potencia y maniobrabilidad. Las configuraciones más comunes son la configuración diferencial, de triciclo y de Ackerman (Orozco, Valencia y Parra, 2006).

Los robots que emplean patas como sistema de locomoción son ideales para terrenos irregulares, pero presentan grandes desventajas, como algoritmos complejos para su desplazamiento al igual que la falta de espacio para la instalación de sensores y ubicación de carga útil (Cano, 2007) .

En los robots que utilizan orugas se pueden considerar estas como ruedas de gran tamaño, en donde su funcionamiento es análogo a los robots con ruedas tipo *Skid Steer* (Ollero, 2005), en donde la dirección y el desplazamiento responden a la velocidad de cada sistema de ruedas o cintas deslizantes.

Los robots que emplean este sistema locomotor son ideales para ambientes agrestes y con obstáculos grandes. Gracias a que distribuyen su peso en una superficie más grande que las ruedas y las patas, pueden desplazarse en terrenos donde la superficie es blanda como el lodo, al igual que facilita su desplazamiento en terrenos que se erosionan con facilidad. Una de sus desventajas es la

gran demanda de potencia debido a la pérdida de energía, ocasionada por la fricción con la superficie de desplazamiento, además el derrapaje que generan las orugas hace muy difícil determinar la posición del robot por los métodos de odometría convencionales. A pesar de esto las orugas se emplean en este trabajo, pues el terreno por el que se moverá el robot es uniforme y el rastreo, como se verá más adelante, se hará por medio de una cámara aérea, por lo cual el derrapaje no afectará el cálculo de la posición del robot.

Sistema sensorial. Le permite al robot obtener información tanto de sí mismo como del exterior. En el primer caso sirve para conocer variables relacionadas con su funcionamiento, como nivel de batería o la posición de sus actuadores (sensores propioceptivos); mientras que en el segundo caso se emplea para reconocer el mundo que lo rodea, censando variables como distancia a obstáculos u orientación con respecto al polo magnético (sensores exteroceptivos) (Aracil, 2002); generalmente está compuesto por un conjunto de sensores de diferentes tipos como: infrarrojos, ultrasónicos, laser, cámaras, etc.

El éxito de una tarea radica en un buen porcentaje de la calibración de estos elementos, su ubicación en el robot y la capacidad de subsanar las deficiencias que cada uno posee con las ventajas de otros, por ejemplo, los sensores ultrasónicos presentan dificultades al identificar eficientemente obstáculos huecos o que absorban las ondas sonoras, desventaja que es corregida mediante la implementación en conjunto con sensores infrarrojos. En este trabajo el reconocimiento del mundo exterior está a cargo de una cámara web, la cual reemplaza los sensores tradicionales, como los infrarrojos o los ultrasónicos, con el fin de facilitar la implementación de la Realidad Aumentada (RA) y disminuir costos.

Sistema de comunicaciones. El sistema de comunicaciones es el conjunto de componentes por medio de los cuales el robot interactúa con elementos externos. Estos elementos no solo son computadores o dispositivos móviles, sino que también pueden ser otros robots. En la actualidad, los protocolos más comunes en sistemas alámbricos son los seriales como RS232, I2C y el LIN (Cano, 2007). En los sistemas inalámbricos son el protocolo *Zigbee* y el TC/IP vía *wireless*.

Generalmente están compuestos por un emisor o codificador, un medio de transmisión y un receptor o decodificador. En el caso de este trabajo se decidió implementar un sistema de comunicaciones basado en protocolo *Zigbee*, por su fácil conexión y baja complejidad de hardware, requerido para ser implementado tanto con un microcontrolador como con un PC. Adicionalmente, la robustez de dicho protocolo contribuye a la estabilidad del sistema robótico.

Sistema de control. El sistema de control es el encargado de gobernar el robot definiendo las rutas a seguir basándose en el algoritmo de navegación y la información suministrada por el sistema sensorial. Normalmente está representado por un elemento o dispositivo de cómputo como un microcontrolador, un microprocesador o un PC, aunque en algunos casos también está conformado por elementos no reprogramables como compuertas electrónicas simples que implementan algoritmos de baja complejidad.

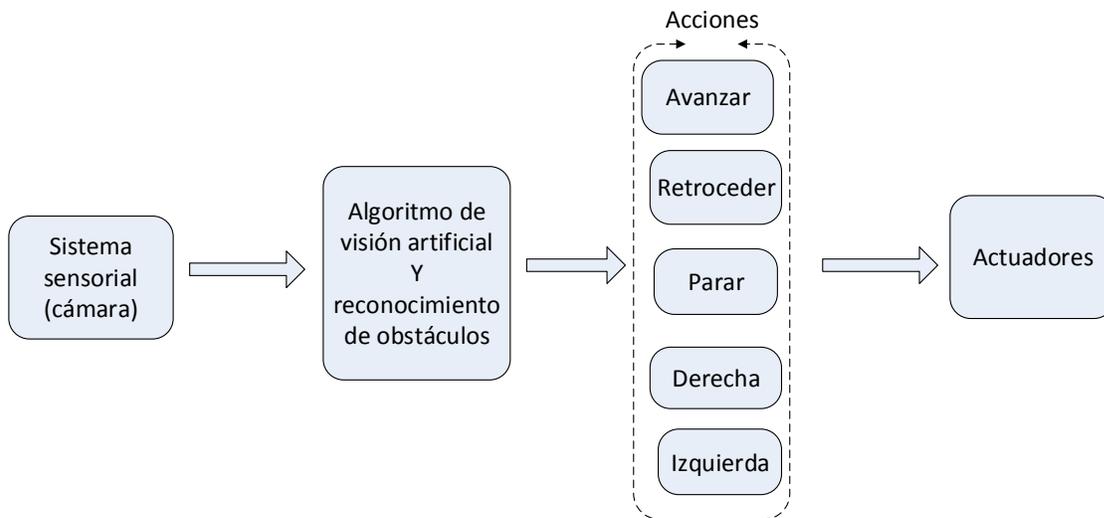
En este caso el corazón del sistema de control está representado por un PC. Este elemento se eligió debido a la complejidad que implica, primero, procesar las imágenes provenientes de la cámara; segundo, implementar la RA; y tercero, correr el algoritmo de navegación. Todo esto de forma simultánea.

1.1.2. Algoritmo de navegación

Algoritmos de navegación. Es el elemento encargado de la planificación de las trayectorias y de llevar al robot desde un punto de inicio hasta un punto final. Generalmente están basados en métodos clásicos de planificación como: grafos de visibilidad, diagramas de Voronoi, modelado del espacio libre, descomposición en celdas, campos potenciales y comportamiento reactivo. Estos últimos son ideales para tareas en las cuales no se tiene un mapa confiable del terreno, o la información del ambiente de pruebas es incompleta (González, Rodríguez, Sánchez, López y García, 2006).

Algoritmo de navegación reactivo. La parte comportamental en este algoritmo está determinada por la información suministrada por los sensores, los cuales de acuerdo con su estado detonan acciones simples como avanzar, retroceder, girar, etc. La capa deliberativa es simple o inexistente, por lo cual se emplea en tareas con bajo nivel de dificultad como evasión de obstáculos en ambientes cambiantes.

Figura 1. Diagrama de flujo algoritmo de navegación reactivo básico



Fuente: Autor

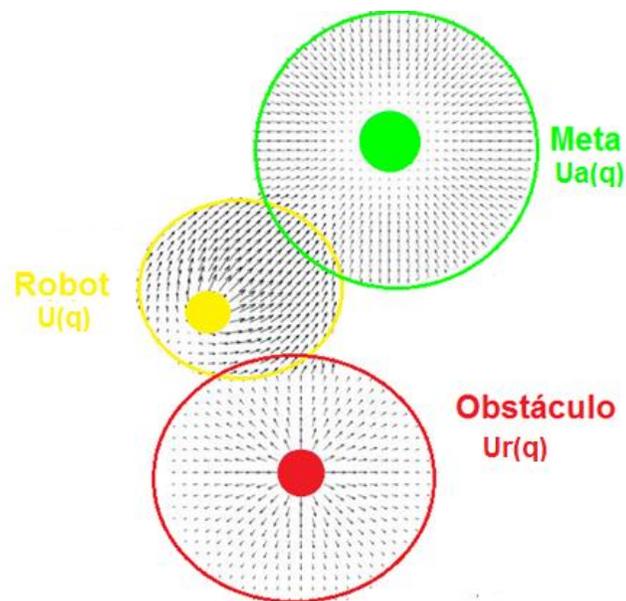
Algunas de las ventajas de los algoritmos reactivos son sus bajos requerimientos de cómputo, facilidad de desarrollo modular y que no requiere un modelo del mundo para su implementación, pero tiene problemas al tratar de implementar tareas complejas y seguir una trayectoria predefinida.

Algoritmo de campos potenciales. Se basan en el principio del tratamiento del robot como una partícula influenciada por campos potenciales artificiales, que emanan tanto de los elementos presentes en el mundo como del punto de destino (Osuna, 2010), asignando potenciales de repulsión μ_r a los obstáculos y de atracción μ_a a la meta.

Cada obstáculo emana un campo de repulsión distinto, el cual al interactuar con los otros dos campos (el del robot y la meta) generan una fuerza resultante que dirige el robot.

A continuación se muestra la Figura 2, que ilustra la interacción de los campos entre sí.

Figura 2. Representación de interacción de campos potenciales artificiales



Fuente: Autor

Basado en lo anterior, se puede decir que el potencial incidente sobre la partícula representada por el robot en el punto q $\mu(q)$ se puede expresar de esta forma:

Ecuación 1. Potencial incidente sobre el robot

$$\mu(q) = \mu_a(q) + \mu_r(q)$$

Teniendo en cuenta lo anterior, la fuerza es calculada como se ve en la siguiente ecuación:

Ecuación 2. Fuerza incidente sobre el robot

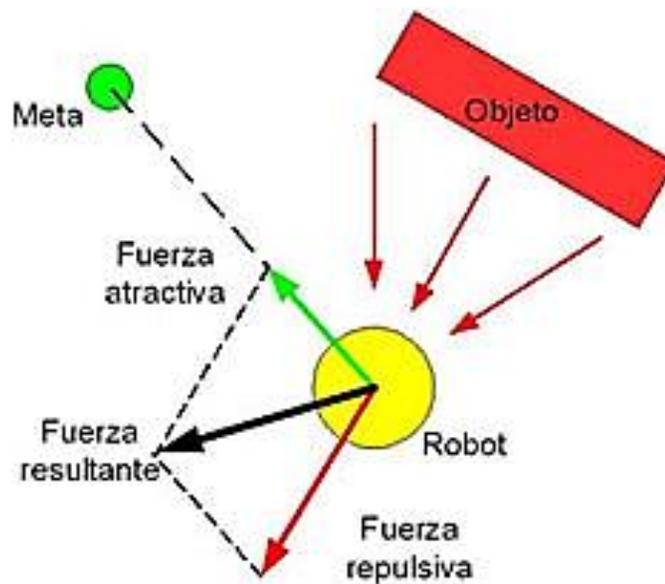
$$\mathbf{F} = -\nabla\mu(\mathbf{q})$$

Lo que significa que la fuerza resultante sobre el robot es la suma de las fuerzas de repulsión F_r más la suma de las fuerzas de atracción generadas por la meta F_a . Ver ecuación 3.

Ecuación 3. Sumatoria de fuerzas en el robot

$$F(\mathbf{q}) = \sum F_a + \sum F_r$$

Figura 3. Modelo de fuerzas de campos potenciales



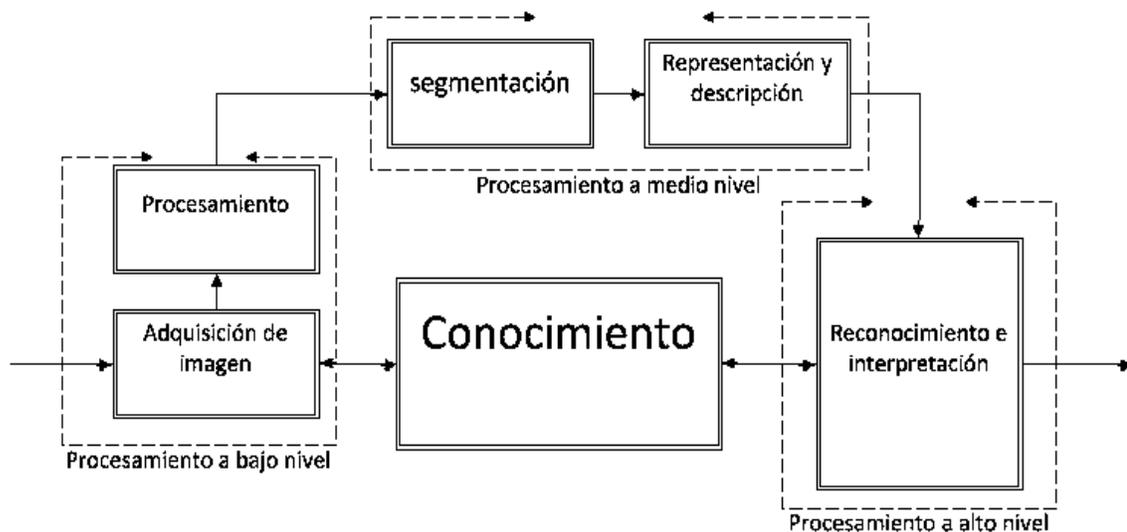
Fuente: Autor

1.1.3. Visión artificial

La visión artificial es una rama de la inteligencia artificial, la cual analiza las imágenes digitales captadas por un sistema óptico, buscando información relevante para una tarea o la interpretación automática de la imagen.

La visión artificial reúne desde los procesos y técnicas empleadas para la captura de la imagen y su procesamiento (procesamiento de imágenes a bajo nivel), pasando por la segmentación y descripción (procesamiento de imágenes a nivel medio), hasta la identificación de características relevantes y el reconocimiento de patrones (procesamiento de imágenes a alto nivel).

Figura 4. Niveles de procesamiento de imágenes



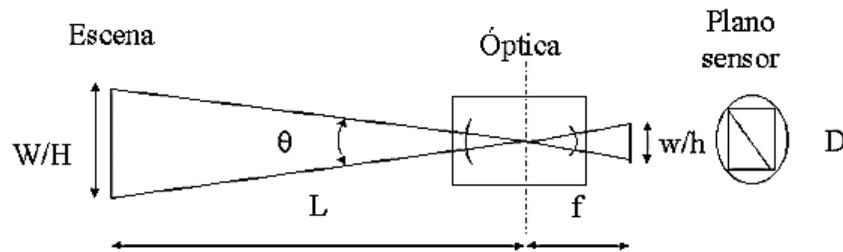
Fuente: Autor

La visión artificial en este trabajo cumple el papel de sistema sensorial, el cual permitirá identificar marcadores de RA, generar volúmenes y tomar decisiones de desplazamiento, esto basado en procesamiento a bajo nivel, es decir, aplicando solo las etapas de adquisición de la imagen y su procesamiento.

1.1.4. Ángulo de visión

El ángulo de visión es la proporción que determina la extensión del escenario que un sensor óptico en conjunto con un lente puede cubrir, y generalmente está compuesto por una medida vertical, una horizontal y una transversal. El ángulo de visión se obtiene del ángulo que forman dos rayos que inciden sobre los bordes externos de la zona sensible de la imagen, cuando la imagen se encuentra enfocada a una distancia infinita.

Figura 5. Ángulo de visión



$$\frac{w}{W} = \frac{h}{H} = \frac{f}{L}$$
$$\theta = 2 \arctan\left(\frac{D}{2f}\right)$$

En el caso de este trabajo, el concepto de ángulo de visión contribuirá en la construcción del volumen de visión de las cámaras empleadas y servirá para establecer recomendaciones para el uso de marcadores de RA en sistemas robóticos integrados con visión artificial.

1.1.5. Calibración de parámetros intrínsecos de las cámaras

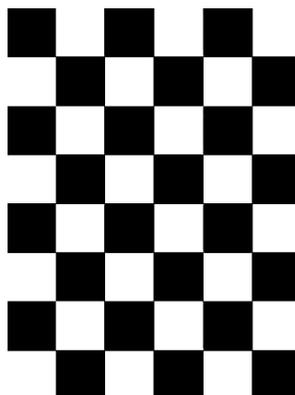
La calibración de la cámara es el procedimiento mediante el cual se busca establecer los parámetros intrínsecos o inherentes a la cámara (distancia focal, centro óptico, modelo de deformación del lente).

Aunque existen muchas herramientas para realizar la calibración de una cámara, en general todas las técnicas siguen los mismos pasos básicos:

1. Establecer con precisión las coordenadas de un punto 3D del mundo exterior, el cual servirá de referencia para el sistema.
2. Fijar la correspondencia del punto anterior con la representación 2D dada por el sensor óptico.
3. Obtener la solución a la obtención de los parámetros intrínsecos y extrínsecos de la cámara por medio de procedimientos matemáticos de optimización, lo que servirá para poder calcular un punto 3D del mundo con una imagen 2D de la cámara.

Para este procedimiento se emplean, generalmente, patrones simples monocromáticos con el fin de facilitar la identificación de los puntos de calibración, el más común es el patrón de tablero de ajedrez debido a su simplicidad y la fácil fijación de las esquinas por el algoritmo de calibración (Barranco, 2010; Fetić & Jurić, 2012; Mejía & López, 2014).

Figura 6. Patrón de calibración de tablero de ajedrez



Fuente: Autor

Este procedimiento arroja los parámetros intrínsecos de la cámara, entre los que se cuentan:

- Focal length (fc): distancia focal
- Principal points (cc): puntos principales
- Skew (alpha c) : coeficiente de asimetría que define el ángulo entre los pixeles del X y del eje Y

Con estos parámetros se contrallega la matriz de parámetros intrínsecos KK, como se ve en la ecuación 4.

Ecuación 4. Matriz de parámetros intrínsecos KK

$$KK = \begin{bmatrix} fc(1) & \alpha_c * fc(1) & cc(1) \\ 0 & fc(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix}$$

Con la ecuación KK se puede obtener la ecuación de los parámetros extrínsecos, como se ve a continuación:

Ecuación 5. Matriz de parámetros extrínsecos de la cámara

$$\begin{bmatrix} Xp \\ Yp \\ 1 \end{bmatrix} = KK \begin{bmatrix} X_a(1) \\ X_a(2) \\ 1 \end{bmatrix}$$

Para validar la matriz KK se realiza una re proyección del mundo 2D de la cámara al mundo real en 3D sobre una imagen real. El primer paso para ello es calcular la matriz H de transformación 2D a 2D, que transforma un punto en el plano del mundo $X_{x,y}$ al plano de la cámara $X_{i,j}$ de tal forma que:

Ecuación 6. Transformación de un punto 2D del mundo al plano 2D de la cámara

$$X_{x,y} = HX_{i,j}$$

Donde la matriz H 3x3 se puede llamar matriz de transformación mundo-cámara, teniendo así:

Ecuación 7. Matriz de transformación mundo cámara

$$\begin{bmatrix} i \\ j \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Para llevar a cabo la re proyección, se obtienen la matriz de rotación R_t y el vector de traslación P que constituyen los parámetros extrínsecos de la cámara, como se aprecia a continuación (Mejía & López, 2014) :

Ecuación 8. Ecuación de λ

$$\lambda = \frac{\|K^{-1}h_1\| + \|K^{-1}h_2\|}{2}$$

Ecuación 9. Matriz de rotación R_t

$$R_t = \frac{K^{-1}H}{\lambda}$$

Ecuación 10. Vector de rotación P

$$P = k[r_1|r_2|r_3|t]$$

Los parámetros obtenidos con este procedimiento son comúnmente utilizados para percepción y navegación visual, pues permiten relacionar un punto 2D de la imagen de la cámara con un punto 3D del mundo real. En el caso de este trabajo, este procedimiento se utilizará para la validación de parámetros de calibración y la estimación del error presente en la representación de objetos virtuales en una imagen real.

1.1.6. Realidad Aumentada

Se puede decir que «la Realidad Aumentada es el término que se usa para definir una visión directa o indirecta de un entorno físico del mundo real, cuyos elementos se combinan con elementos virtuales para la creación de una realidad mixta a tiempo real» (Hernández, 2012), igualmente se puede decir que «un sistema de Realidad Aumentada es aquel que complementa el mundo real mediante el uso de elementos virtuales generados por computador que parecen coexistir en el mismo espacio que los elementos reales» (Rodríguez, 2011).

Complementando lo anterior:

La Realidad Aumentada (RA) es una rama de la realidad mixta, la cual tiene como objetivo el enriquecimiento de una imagen del mundo real con información u objetos virtuales generados por computadora, a diferencia de la realidad virtual la cual se basa en la inmersión del individuo en un ambiente puramente virtual o sintético (Jaramillo, 2011).

Esto se puede ver mejor en la taxonomía propuesta por Milgram, en donde la Realidad Aumentada se ubica en medio de la realidad virtual y el mundo real, acercándose más a este último; esto debido a que usa de base ambientes reales incorporando objetos virtuales solo como un complemento.

Figura 7. Taxonomía de Milgram



Fuente: Jaramillo, 2011.

Los sistemas de RA se pueden clasificar, según su técnica de reconocimiento, como patrones basados en imágenes y basados en la localización (Cheng & Tsai, 2012), en donde los sistemas basados en imágenes utilizan marcadores o imágenes del ambiente para emplearlos como referencias al momento de aumentar su entorno; y los sistemas basados en localización emplean coordenadas de GPS, brújulas y/o acelerómetros para referenciar el elemento aumentado ya sea videos, sonido u objetos 3D.

Figura 8. Clasificación de la Realidad Aumentada según la técnica de reconocimiento

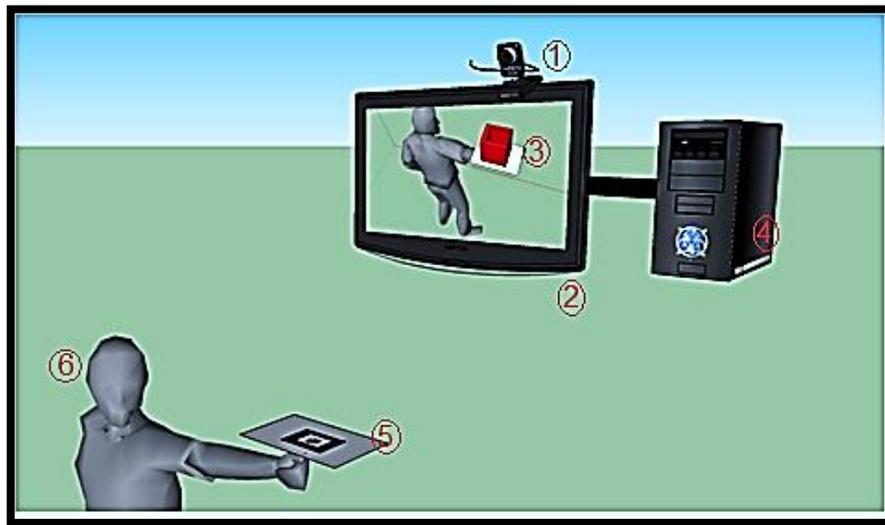


Fuente: Cheng & Tsai, 2012.

Según esta taxonomía el presente trabajo se puede clasificar como RA basada en imágenes, la cual está conformada generalmente por los siguientes elementos:

- Cámara: es el elemento que se encarga de captar la imagen.
- Ordenador: es en el que se aloja el software de RA y al cual llega la imagen captada por la cámara; se encarga de calcular la posición y la orientación del objeto virtual en la imagen.
- Elemento visualizador: se encarga de mostrar el resultado de la integración del mundo real y el elemento virtual.
- Marcador: es un código de barras bidimensional monocromático dentro de un cuadrado impreso sobre una superficie plana el cual sirve de referencia para la ubicación del objeto virtual (Garrido & García-Alonso, 2007).

Figura 9. Sistema de Realidad Aumentada. 1) Cámara, 2) Pantalla, 3) Elemento Virtual, 4) Ordenador, 5) Marcador, 6) Usuario



Fuente: González Morcillo, López, & Jiménez, 2009.

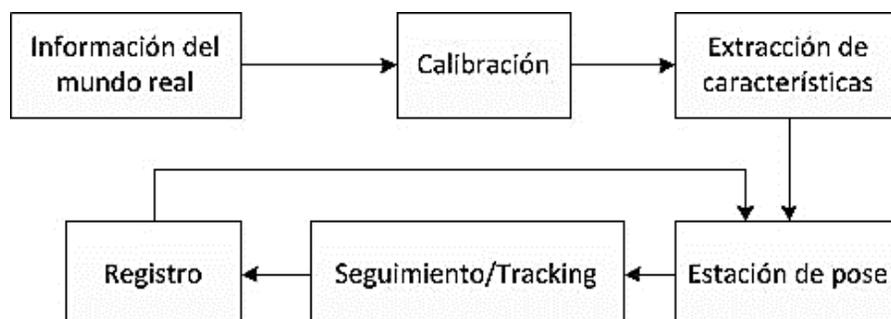
Los elementos antes mencionados pueden cambiar ligeramente según el tipo de sistema, tal es el caso de los sistemas binoculares donde se utilizan dos cámaras en vez de una o en los sistemas de

RA sin marcadores, en donde este último es remplazado por un conjunto de características extraídas de una imagen del mundo real (I. Y. H. Chen et al., 2008).

En esta tesis la cámara estará ubicada en la parte frontal del robot; la pantalla no solo servirá de indicación para la RA sino que se utilizará como interfaz de comunicaciones entre el software de RA y el software donde se alojará el algoritmo de tratamiento digital de imágenes; el elemento virtual representará objetos u obstáculos con los que el robot interactuará; el marcador estará ubicado en el ambiente de pruebas con el fin de fijar la posición y orientación de los objetos virtuales; y por último, el usuario será el mismo robot, pues es quien tomará esta Realidad Aumentada y la integrará a su sistema de navegación.

El proceso general de la Realidad Aumentada comienza con la obtención de información del mundo real por parte de la cámara (imagen), posteriormente se efectúa una calibración de la misma, la cual en la mayoría de los software de RA es realizada de forma automática. Acto seguido, se hace la extracción de características para calcular la pose de la cámara real y generar el modelo de la cámara virtual. Con este modelo se procede a realizar el seguimiento del marcador registrando e insertando el objeto virtual.

Figura 10. Proceso general de la Realidad Aumentada



Fuente: Jaramillo Rojas, 2011.

El proceso de calibración, mencionado en la Figura 10 puede ser realizado por el usuario o como en el caso de este trabajo, se realiza de forma automática por el mismo software de RA; adicionalmente se aclara que este proceso se aplica a sistemas monoculares, binoculares con o sin marcadores. Entre los software más usados en el desarrollo de aplicaciones de Realidad Aumentada para PC son el *ARToolKit*, *BuildAR* y *BuildAR Pro*, los cuales ofrecen herramientas tanto de visualización como de diseño.

1.2. Estado del arte

En este subcapítulo se mostrarán las diferentes tendencias de los temas pertinentes a este trabajo como son la robótica móvil y la Realidad Aumentada.

1.2.1. Visión artificial en robótica móvil

En general, la visión en la robótica se refiere principalmente a la capacidad de un robot para percibir visualmente el medio ambiente circundante (Martínez-Gómez et al., 2014). También se puede decir que es el estudio del uso de cámaras como sensores para la robótica. Su función principal es reconocer y localizar objetos en el entorno, mediante el procesamiento de las imágenes y con el objetivo de «entender» una escena o extraer características de una imagen para tomar decisiones.

En la actualidad, los sistemas de visión en robótica móvil están compuestos principalmente de una o dos cámaras (digital, analógica o híbrida, un procesador (puede ser un PC o un sistema embebido) y un software que procesa la imagen basado en una técnica de inteligencia computacional, como las redes neuronales, lógica borrosa o híbridas (Adrian, 2004). El software descompone la imagen para extraer las características necesarias y cumplir la tarea asignada, un ejemplo de ello es discriminar elementos como paredes, pisos y objetos estáticos basándose en la textura, eliminando de la imagen los elementos no deseados y conservando solo los relevantes (Jiménez et al, 2008).

Uno de los sistemas ópticos más usados es el binocular, el cual se emplea en la reconstrucción en 3D de ambientes estáticos y funciona bajo los parámetros de la visión humana; estos sistemas fusionan las dos imágenes captadas por las cámaras para medir el desfase y calcular los puntos tridimensionales, utilizando la distancia conocida entre ellas y los ángulos de rotación e inclinación de cada una. Su implementación en sistemas de navegación de robots se ha dado gracias a su bajo costo, comparada con otras alternativas como los rastreadores láser, además la visión artificial facilita la obtención de modelos tridimensionales de los ambientes de prueba manteniendo o mejorando la precisión del sistema, optimizando los cálculos de trayectorias y mejorando los comportamientos dinámicos de los robots (Luo & Lai, 2014). Aunque los sistemas binoculares presentan grandes ventajas, estos generan un gran costo computacional, tanto para la reconstrucción de los elementos como para la interpretación de los mismos. También se tienen los sistemas de visión 2 ½ D, conformados por una cámara y un elemento que ayuda a la percepción de profundidad, este puede ser un sensor láser, infrarrojo o ultrasónico (K. Wang, Liu & Li, 2014), simplificando el procesamiento de las imágenes, pero dependiendo exclusivamente de estos sensores para aportar el componente de profundidad. Lo anterior implica que el sistema de visión se torna vulnerable a errores causados por las falencias en sus principios de funcionamiento (dificultad de detección de los elementos translucidos en los sensores ópticos y los espacios cóncavos en los acústicos) (Carrizo et al. & Mut, 2006; Jiménez et al., 2008; Pérez, et al., 2009).

Otra técnica se basa en la distribución de la velocidad del movimiento aparente de los patrones de brillo en la imagen para brindar información espacial en corredores, teniendo como resultado un sistema que solo requiere una cámara para tener una perspectiva de profundidad (Sanahuja et al., 2011; Soria & Carelli, 2006). Todas las técnicas vistas anteriormente presentan problemas en la interpretación de imágenes cuando existe pobre iluminación (Castillo, Andrés, López & León, 2006).

Con respecto a la navegación de robots utilizando los sistemas de visión, esta se realiza de diferentes formas, la más común es tomando la imagen de un sistema de visión monocular y trazando la ruta basada en el promedio de los espacios libres de la superficie por la cual se desplaza

el robot (Jiménez et al., 2008). También se puede realizar un trazado de la trayectoria por hitos o puntos clave, los cuales deben ser fácilmente identificables por el sistema de visión.

Igualmente es posible lograr que el robot trace una ruta empleando algoritmos inteligentes, alimentados con la información captada por el sistema de visión artificial. Los algoritmos más comunes empleados para ello son las redes neuronales o lógica borrosa (Bonilla, Novelo, Enríquez & Cortés, 2009).

Una técnica de navegación no muy común, utilizando sistemas monoculares, es la que emplea cámaras omnidireccionales, las cuales dan una vista de 360 grados optimizando al máximo el empleo de una sola cámara para la navegación (X. Wang & Kwok, 2007). No obstante, estos sistemas presentan grandes dificultades al momento de la interpretación de las imágenes debido al alto grado de aberraciones ópticas que se presentan.

Analizando los sistemas de visión artificial se observan varias ventajas, entre las cuales se cuentan la gran cantidad de información que se puede extraer de una imagen y el poder cambiar un anillo de sensores compuesto por una gran cantidad de elementos, por solo una o dos cámaras.

Para concluir, se debe tener en cuenta que en todas las técnicas mencionadas se requiere que los elementos físicos estén presentes en los ambientes de entrenamiento de los robots, en caso contrario, dicho entrenamiento no se puede realizar o presenta grandes deficiencias bajo los parámetros y estándares de estas técnicas.

1.2.2. Realidad Aumentada

Los ambientes de Realidad Aumentada (RA) se pueden definir como aquellos que mezclan elementos reales y virtuales en un ambiente único (Jaramillo, 2011 y Rodríguez, 2011).

Los sistemas de Realidad Aumentada no son nuevos, en realidad existen desde hace varias décadas, pero solo hasta algunos años la tecnología ha avanzado lo suficiente para que esta entre a un mayor uso debido a los avances en los computadores y dispositivos móviles, como tablets y smartphones (Abdulmuslih, 2012).

Desde el punto de vista cronológico, la evolución de la Realidad Aumentada en sus inicios, 1960 a 1969, tuvo pocos cambios, al igual que en las décadas de los 70 y 80 cuando la mayoría de los sistemas se basaban en cascos donde se implementaba la RA: En las décadas subsiguientes, 90 y 2000, se ve un mayor crecimiento en el desarrollo de la RA, pasando de sistemas voluminosos y de difícil manipulación a sistemas compactos y de dominio público, esto propiciado por el auge de Internet.

En el siguiente cuadro se puede apreciar de forma general el recorrido tecnológico y los eventos críticos en la historia de la Realidad Aumentada

Tabla 2. Recorrido tecnológico y los eventos críticos en la historia de la Realidad Aumentada

Fecha	Evento
1960	Morton Heilig, Telesphere Mask, crea el primer dispositivo para ser montado en la cabeza con la pantalla, que proporciona estereoscópica (3D) TV, amplia visión y sonido estéreo real [Morton Heilig].
1969	Morton Heilig, Experiencie Theater, es un cine con una gran pantalla semiesférica, que muestra en 3D imágenes en movimiento, con imágenes periféricas, sonido direccional, aromas, viento, las variaciones de temperatura y la inclinación del cuerpo del asiento. El público está sentado en el punto de enfoque en los asientos de la escena [Morton Heilig].
1973	Ivan Sutherland inventa el <i>display</i> de cabeza (HMD).
1985	Nacimiento de Hurto. Myron Krueger crea Videoplance, que permite por primera vez a los usuarios interactuar con objetos virtuales
1990	Jaron Lanier acuña el termino realidad virtual y crea la primera actividad comercial en torno a los mundos virtuales.

1992	Tom P. Caudell y David W. Mizell crean el término Realidad Aumentada.
1994	Steven Feiner, Blair MacIntyre y Doree Seligmann hacen la primera utilización importante de un sistema de Realidad Aumentada en un prototipo.
1999	Hirokazu Kato desarrolla <i>ARToolKit</i> en el <i>HitLab</i> y se presenta en SIGGRAPH ese mismo año.
2000	Bruce H. Thomas desarrolla <i>ARQuake</i> , el primer juego al aire libre con dispositivos móviles de Realidad Aumentada.
2008	AR Wikitude Guía sale a la venta el 20 de octubre de 2008 con el teléfono Android G1.
2009	<i>AR Toolkit</i> es portado a <i>Adobe Flash (FLARToolkit)</i> por Saqoosha, con lo que la Realidad Aumentada llega al navegador Web.
2009	Se crea el logo oficial de la Realidad Aumentada con el fin de estandarizar la identificación de la tecnología aplicada en cualquier soporte o medio por parte del público general.
2012	<i>Google</i> se lanza al diseño de unas gafas que crearían la primera Realidad Aumentada comercializada. Bautiza a su proyecto como <i>Project Glass</i>

Abdulmuslih Alsirhani, 2012.

En sus primeros años, la presentación de la Realidad Aumentada se daba gracias a un casco o módulo dotado con lentes en los cuales se proyectaba la información, un elemento de rastreo de marcadores o patrones y todo esto conectado a un sistema de cómputo de gran poder para su época. Debido a su complejidad y difícil implementación, esta tecnología solo se encontraba en los laboratorios de las universidades o en aplicaciones fuera del alcance del público, como en el sector aeroespacial (Caudell & Mizell, 1992) .

Años después se presentaron los primeros sistemas de RA móviles, los cuales en su mayoría compartían la estructura básica de los sistemas iniciales diferenciándose solo en la portabilidad brindada por un equipo de cómputo portátil, el cual a pesar de su condición seguía siendo voluminoso y sin la potencia necesaria para presentar gráficos en 3D (Feiner, Macintyre, Höllerer & Webster, 1997; Piekarski & Thomas, 2003).

Solo hasta años recientes los avances en los PC portátiles y la implementación de las tabletas y *Smartphone* la Realidad Aumentada ha podido masificarse y mostrar todo su potencial.

En la actualidad la RA ha migrado en gran medida hacia los dispositivos móviles, aunque se ha regresado al concepto de los lentes, gracias a las nuevas películas semitransparentes que convierten un lente normal en un mini *display* a color, prueba de ello son las *Google Glass* que retoma los conceptos tecnológicos iniciales de la Realidad Aumentada aprovechando las nuevas tecnologías para llevar al extremo el concepto de RA portable.

Existen varias formas de implementar los sistemas de Realidad Aumentada, pero todas estas se pueden dividir en sistemas basados en marcadores y sistemas sin marcadores; los primeros son los más comunes y los más fáciles de implementar, pues la carga del cálculo de la posición y orientación de los objetos virtuales se vuelca en la medición de la deformación óptica del marcador (Garrido & García, 2007; Jaramillo, 2011). Esta técnica presenta gran robustez y facilidad de aplicación debido a que el procesamiento del marcador es más simple que la extracción de características de una imagen compuesta de un escenario (Jaramillo, 2011), aunque la obligatoriedad de la presencia del marcador, previamente diseñado, presenta una desventaja.

Los sistemas sin marcadores se pueden encontrar ahora para todos los dispositivos móviles y PC (Garrido & García, 2007), pero el proceso de extracción de características de la imagen de referencia demoran la ejecución de las aplicaciones, lo que las hace menos eficientes que los sistemas con marcadores pues requieren mayor poder de procesamiento (Jaramillo, 2011).

Los sistemas de RA sin marcadores también pueden ser complementados con sensores externos como acelerómetros, brújulas o sensores comerciales como el Kinect para mejorar su rendimiento y brindar mayor robustez (Sacramento de Britto & López Apolinário Júnior, 2013). Sin embargo estos sistemas generan mayor costo computacional que los implementados con marcadores; igualmente si en un sistema de RA sin marcadores se toma como plantilla una imagen de un ambiente susceptible cambios, como un corredor industrial o el interior de una oficina, las

posibilidades de que esta sea alterada por la intervención humana es mucho mayor que las de un marcador impreso, por lo cual es una opción poco atractiva para estos ambientes.

Respecto a software de diseño de aplicaciones de RA se puede decir que entre los más usados se encuentran *ARToolkit*, *ATOMIC Autoringtool* y *BuildAR*, con los cuales se pueden realizar todo tipo de aplicaciones en Realidad Aumentada además de que su gran ventaja es que son de distribución libre.

Se ha encontrado que en años recientes la aplicación de la RA se ha extendido al mejoramiento de ambientes de simulación, al igual que el enriquecimiento de ambientes de prueba con objetos virtuales en robots tele operados para validar su comportamiento, no solo con objetos inmóviles sino también con otros robots de naturaleza virtual (BędkowskiMas & Masłowski, 2009).

Este enfoque en la mayoría de los casos no explota todo el potencial de esta tecnología, empleando solo la RA como un apoyo visual y no como un componente determinante de la navegación autónoma, desaprovechando la posibilidad que el robot entienda este mundo híbrido (real-virtual) y limitándose solo a intervenir los sensores tradicionales como el láser o el ultrasónico. Adicionalmente, en estos sistemas rara vez son validados los comportamientos derivados de objetos virtuales con sus pares reales.

Asimismo se ha utilizado la Realidad Aumentada para ayudar en la ubicación de robots móviles con sistemas de visión omnidireccionales y monoculares estándar, en donde los marcadores sirven como puntos de referencia para ubicar el robot en el espacio empleando los datos proporcionados por el software de RA (Ross & Bradshaw, 2012; X. Wang & Kwok, 2007) . Cabe aclarar que las aplicaciones de esta técnica en particular no muestran indicios del aumento del ambiente con obstáculos virtuales, limitándose a instrumentalizar el software de RA para extraer datos útiles destinados a la ubicación y la navegación. Otro enfoque propone el mejoramiento del diseño de las trayectorias ampliando las zonas de seguridad con Realidad Aumentada, simplificando el problema al encerrar en cubos virtuales los obstáculos relevantes escogidos por el usuario ayudando al

sistema y complementando la inteligencia del robot con el criterio y la capacidad de análisis del ser humano (Carff Johnson, El-Sheikh, & Pratt, 2009).

Este tipo de enfoque es bastante prometedor, pero obliga a la supervisión constante del proceso restándole importancia o aún más eliminando casi por completo la autonomía del robot. Adicionalmente, la visión del escenario captada por la cámara del robot es puramente informativa pues las decisiones con respecto a la trayectoria son tomadas según los objetos seleccionados y aumentados por el usuario.

La RA no solo se ha integrado con robots móviles terrestres sino que también se ha incorporado a sistemas robóticos aéreos, con el fin de estudiar el comportamiento de la RA en vehículos que se mueven en 3 ejes. Sus aportes están enfocados a la caracterización de los sistemas para el desempeño óptimo de este tipo de vehículos cuando se relacionan con la RA en ambientes controlados, también se centran en el mejoramiento de los algoritmos de rastreo de características de paisajes y cómo la escala de las marcas afectan el comportamiento de la RA (I. Y. Chen, Macdonald & Burkhard, 2008). Cabe anotar que aunque su enfoque es para trabajar en exteriores (*outdoor*), todas las pruebas de validación se hacen a nivel de laboratorio.

Se debe considerar en la literatura que la mayor parte de trabajos están concentrados en robots teledirigidos y en sistemas que no toman la visión artificial como una herramienta viable para el estudio de la RA en la robótica. Adicionalmente los robots implementados no poseen inteligencia a bordo o, en el mejor de los casos, el procesamiento de los datos se lleva a cabo en una estación de trabajo y es devuelta al robot de forma inalámbrica extendiendo el tiempo de respuesta y aumentando la complejidad del sistema.

Se puede decir que aunque existen sistemas de robot móviles empleando Realidad Aumentada, esta no es utilizada como un elemento fundamental en la navegación, relegándola a complementar la visión de los robots y participando solo como un apoyo del trazado de las rutas. Igualmente los sistemas que emplean la RA como complemento a la visión artificial son complejos y difíciles de

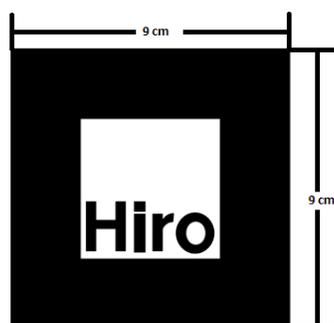
implementar, dejando por fuera el uso de esta tecnología a la mayoría de los usuarios potenciales, como las universidades y el sector productivo en general.

Se debe resaltar la importancia de la integración de la Realidad Aumentada con la navegación, pues esta es una herramienta útil al momento del entrenamiento y validación de algoritmos de robots móviles al unir la robustez de los ambientes físicos con la flexibilidad de los ambientes aumentados. Adicionalmente, la interpretación de elementos como elementos reales en los sistemas de visión y navegación facilita la construcción de ambientes de prueba, al igual que optimiza tiempos de desarrollo al dinamizar el cambio de escenarios con la representación de objetos sólidos de gran peso y tamaño solo con un marcador y un software de RA. Otro aspecto interesante de esta técnica es que elimina la necesidad del modelo cinemático del robot el cual es fundamental en las técnicas de simulación en ambientes puramente virtuales.

1.2.3. Marcadores en Realidad Aumentada

Los marcadores están constituidos por un recuadro de color negro dentro del cual se inserta una figura distintiva (letras, dibujos simples o códigos de barras bidimensionales), la cual relaciona un elemento creado por software como un objeto 3D, un video o una imagen (Garrido & García-Alonso, 2007; White, Feiner & Kopylec, 2006).

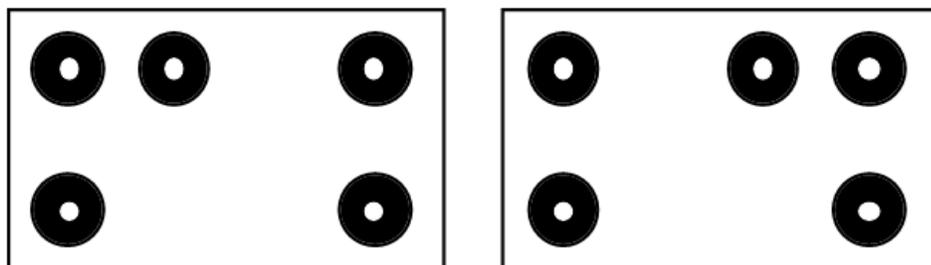
Figura 11. Marcador estándar



Fuente: Autor

Una variación de los marcadores cuadrados son los marcadores binarios circulares, los cuales se disponen sobre una superficie blanca (Hoff & Nguyen, 1996).

Figura 12. Marcadores binarios



Fuente: Hoff, Nguyen, & Lyon, 1996

También existen otros tipo de marcadores, como los coloreados o con figuras de colores (Lee & Woo, 2008; State, Hirota, Chen, Garrett & Livingston, 1996).

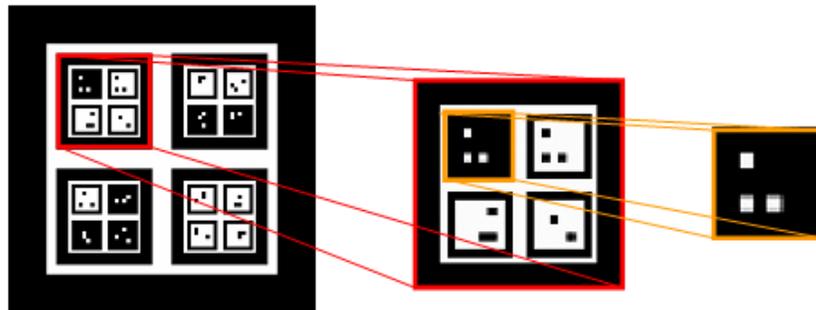
Figura 13. Marcadores a color



Fuente: State, Hirota, & Chen, 1996; b) Lee & Woo, 2008.

Otro enfoque es utilizar marcadores anidados o insertar dentro del recuadro principal varios marcadores, con el fin de mejorar la fijación de los mismos por el sistema. Estos marcadores son conocidos como marcadores jerárquicos (Tateno, Kitahara & Ohta, 2007). Ver Figura 15.

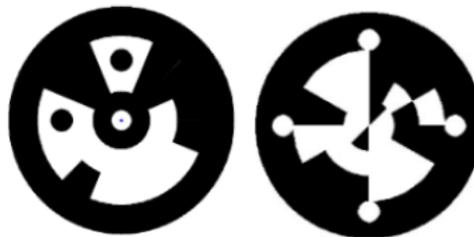
Figura 14. Marcadores jerárquicos



Fuente: Tateno et al., 2007

Desde el punto de vista de la forma, un marcador cuadrado no es la única opción, también se pueden generar de forma circular lo que mejora los problemas de oclusión, aunque presenta problemas de ambigüedad en la pose de la cámara cuando se emplea un solo marcador (Köhler, Pagani, & Stricker, 2011).

Figura 15. Marcadores circulares



Fuente: Koehler, Pagani & Stricker, 2010

1.2.4. Aplicaciones de la Realidad Aumentada

Refiriéndonos a los campos de aplicación, en este momento la RA ha tenido un gran auge en diferentes áreas. A continuación se ilustrarán cada una de ellas.

Un sector en el cual la Realidad Aumentada ha tenido alto crecimiento es el de los dispositivos móviles, que se utilizan para varias aplicaciones entre las que se encuentran entretenimiento, videojuegos, reconocimiento de patrones faciales (Rohs, 2007), (Henrysson, Marshall & Billinghurst, 2007), interfaces de usuario, cartografía y, en el caso particular del iPhone 3GS, posicionamiento de antenas para comunicación satelital.

Las aplicaciones de RA no solo se limitan al iPhone. En este momento se pueden encontrar para la gran mayoría de los sistemas operativos de dispositivos móviles, como son Android, Symbian SO, Windows Mobile, iPhone SO, entre otros (López Pombo, 2010).

No solo las aplicaciones han crecido en los dispositivos móviles sino también en los sistemas de cómputo tradicionales, en ellos podemos encontrar aplicaciones en educación con libros complementados con Realidad Aumentada (Abdulmuslih, 2012), en la medicina con sistemas para la simulación y entrenamiento en procedimientos quirúrgicos (Wen, Tay, Nguyen, Chng & Chui, 2014; Wu, Wang, Liu, Hu & Lee, 2014), investigación y desarrollo (Garrido & García-Alonso, 2007), desarrollo de herramientas interactivas para la educación a distancia en robótica industrial (Jara, Candelas, Fernández & Torres, 2009), en robótica industrial se ha empleado para mejorar las interfaces de usuario de los robots manipuladores y facilitar su programación en tareas complejas, al igual que para facilitar a los programadores sin experiencia la implementación de este tipo de robots (Bischoff & Kazi, 2004). En el campo de los videojuegos los principales fabricantes de consolas han incorporado esta tecnología, tales son los casos de Sony con el PlayStation Move para la consola Playstation 3, Nintendo con el Wiimote para la consola Wii y Microsoft con el Kinect para la consola Xbox 360 (Mendoza & Henao, 2012). De igual forma la cirugía robótica emplea la RA para mejorar el planeamiento de las mismas (Baker, Fryberger & Ponce, 2015).

Un caso más concreto se presenta en la asistencia de la cirugía laparoscópica, en la cual la Realidad Aumentada puede mejorar la visión del cirujano resaltando con gráficas virtuales el área o el órgano a operar (Marescaux, 2004), (Peters et al., 2008).

En la robótica manipuladora, la RA se ha empleado para ayudar en la programación de brazos robóticos industriales (Chong, 2007) al igual que para mejorar las interfaces de usuario.

También se ha utilizado para evaluar las trayectorias y el desempeño de los manipuladores (off-line) sin poner en peligro los robots ni sus operarios (Novak-marcincin, Janak, Barna & Novakova, n.d.), de igual forma la RA se ha empleado para evaluar la destreza de los técnicos a cargo de los robots en empresas tan importantes en el mundo de la robótica como KUKA (Schulze-Ferebee, 2008).

En la robótica móvil también podemos encontrar la utilización de la Realidad Aumentada como forma de enriquecer la interfaz de usuario en los robots teleoperados (X. Wang & Kwok, 2007) que poseen un sistema de visualización remota, ayudando al operador a tomar decisiones más acertadas en la navegación (Dominguez, Essabbah, Cheaib, Otmane & Dinis, 2012; Portilla & Basañez, 2007; Yu, Jin, Luo & Lai, 2010), sin embargo sigue habiendo problemas en esta área en ambientes dinámicos complejos (Carff et al., 2009; Collett & MacDonald, 2006).

1.2.5. Realidad Aumentada y navegación de robots móviles

Se pueden encontrar aplicaciones muy interesantes, en las cuales se utiliza la Realidad Aumentada como instrumento de enriquecimiento de escenarios para la navegación de robots, tanto terrestres como aéreos, en los cuales se busca analizar la interacción de estos con el ambiente a nivel de laboratorio, aunque aún presentan fallas en la fijación y representación de los objetos aumentados, al igual que en el control efectivo de las vibraciones producidas por la inercia del aparato (I. Y. Chen et al., 2008). Adicionalmente, en la mayoría de estos trabajos los robots no interactúan con

los objetos aumentados; y en otros los operadores están a cargo de la navegación de los aparatos, lo que se hace de forma teleoperada. En otros casos, la navegación autónoma se realiza por visión artificial con una cámara pero se limitan a interactuar con el ambiente real sin aumentarlo (Bekris, Argyros & Kavraki, 2004).

Pocas veces la integración de la Realidad Aumentada con marcadores y la navegación se realiza de forma funcional. En la mayoría de los casos esta integración se realiza por medio de los marcadores, los cuales no representan objetos tridimensionales sino que aportan información de posición en los ambientes de prueba (Kim, An & Lee, 2011),(Aleksandrovich & Gennadievich, 2013), (Ross & Bradshaw, 2012).

Otra forma de manejo de los elementos de Realidad Aumentada en la navegación es la utilización de marcadores para dar información direccional al robot en el ambiente de prueba, de modo que solo se le indica en qué dirección seguir al encontrarse con un determinado marcador, lo cual normalmente está acompañado del mapeo del entorno en tiempo real (SLAM), (Zhang, 2014).

Aunque los dos casos anteriores (representación de posición con marcadores y la inserción de información direccional en los marcadores) constituyen un empleo válido de Realidad Aumentada, no explotan el verdadero potencial de esta tecnología en la robótica móvil, que es la representación de objetos tridimensionales en un ambiente real y la interacción con los mismos para impactar la navegación de robots.

Tampoco se analizan las trayectorias de los robots con objetos reales y aumentados para verificar la efectividad de la representación de los objetos aumentados bajo los comportamientos cinemáticos de los robots. Por ultimo no se aprecia una metodología unificada para la implementación de sistemas robóticos que integren la Realidad Aumentada, la navegación y la visión artificial.

A continuación se muestra un cuadro que refleja los comentarios hechos en esta sección de la presente tesis.

Tabla 3. Cuadro que refleja los comentarios hechos en esta sección de la presente tesis

Artículo	Tema	Falencias
Chen, I. Y.-H., MacDonald, B. & Wünsche, B. (2008). Markerless Augmented Reality for Robots in Unprepared Environments. <i>ACRA 2008: Australasian Conference on Robotics and Automation</i> . Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Markerless+Augmented+Reality+for+Robots+in+Unprepared+Environments#1	Realidad Aumentada sin marcadores en campo abierto. Corrección de problemas con la visualización y recuperación de objetos virtuales.	No se concentran en la interacción con los objetos. Pruebas en lugares cerrados y robot tele operado.
Stilman, M., Michel, P. & Chestnutt, J. (2005). Augmented reality for robot development and experimentation. ... <i>Rep. CMU-RI-TR-05-55</i> , 1–11. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.71.9737&rep=rep1&type=pdf .	Realidad Aumentada en campo abierto sin marcadores. Interacción con objetos virtuales, visualización global del campo de experimentación, simulación de robots móviles en el ambiente de experimentación.	Altos recursos técnicos. Elevado poder de cómputo requerido. Se trabaja sin marcadores.
Bekris, K. E., Argyros, A. A. & Kavraki, L. E. (2004). Angle-based methods for mobile robot navigation: reaching the entire plane. <i>IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004</i> , 3(April), 2373–2378. doi:10.1109/ROBOT.2004.1307416	Navegación con robots móviles con registro de trayectorias con cámara panorámica.	No maneja Realidad Aumentada.
Howard, a., & Kitchen, L. (1997). Fast visual mapping for mobile robot navigation. <i>1997 IEEE International Conference on Intelligent Processing</i>	Navegación por visión artificial con una cámara en corredores, haciendo mapas en tiempo real.	No emplea Realidad Aumentada.

<i>Systems (Cat. No.97TH8335), 2. doi:10.1109/ICIPS.1997.669197</i>		
Kim, Y.-G., An, J., & Lee, K.-D. (2011). Localization of Mobile Robot Based on Fusion of Artificial Landmark and RF TDOA Distance under Indoor Sensor Network. <i>International Journal of Advanced Robotic Systems</i> , 8(4), 1. doi:10.5772/45698.	Navegación por visión artificial y red de sensores empleando una cámara apuntando al techo en interiores, (navegación por hitos con marcadores).	Los marcadores no integran Realidad Aumentada solo información de navegación.
Aleksandrovich, Y. D. & Gennadievich, P. G. (2013). Mobile Robot Navigation Based on Artificial Landmarks with Machine Vision System, 24(11), 1467–1472. doi:10.5829/idosi.wasj.2013.24.11.7010	Navegación por hitos, en donde estos son marcadores con códigos de barras bidimensionales (en este caso QR), los cuales contienen información sobre la localización. Delimitación de ambiente con cintas negras.	Los marcadores no tienen objetos 3d relacionados sino información que es decodificada por un software.
Zhang, Y. (2014). <i>Real-time SLAM for Humanoid Robot Navigation Using Augmented Reality</i> by. Simon Fraser University.	Navegación con robot NAO empleando Realidad Aumentada en entornos desconocidos, o sea sin información <i>a priori</i> sobre el ambiente donde se desplaza. Emplea SLAM (navegación y mapeo simultáneamente en tiempo real). La Realidad Aumentada se utiliza para mejorar el desempeño de un algoritmo EKF-SLAM. Emplea marcadores circulares y un módulo de identificación creado para el robot nao el cual devuelve un número asociado al marcador en cuestión, que le aporta información de posición al robot.	(No integra tecnología solo incorpora la información de los números al SLAM y no presenta objetos 3D para interactuar con ellos). La navegación es con el sensor láser del NAO y la visión solo se emplea para la detección de los marcadores (módulo NAOmark)
Gini, G. C., & Marchi, A. (2002). Indoor Robot Navigation With Single Camera Vision. <i>Pris</i> , 67–76. doi:10.1.1.97.4390	Navegación con una cámara empleando un algoritmo de tratamiento digital de imágenes diferenciando el piso de los obstáculos presentes, identificando	No emplea Realidad Aumentada para la navegación ni

	los obstáculos como cuerpos negros en la imagen y modificando su trayectoria de acuerdo con esto. Establece como premisa que el suelo es de textura uniforme, distinguiendo los obstáculos por cambios de textura. Emplea navegación reactiva para evadir obstáculos.	la interacción con objetos virtuales.
Ross, L., & Bradshaw, K. (2012). Fiducial Marker Navigation for Mobile Robots (November).	Emplea marcadores para navegación, un algoritmo de navegación con marcadores de referencia, emplea ARToolKit.	Los marcadores se emplean como complemento de la navegación, no tienen objetos 3D.

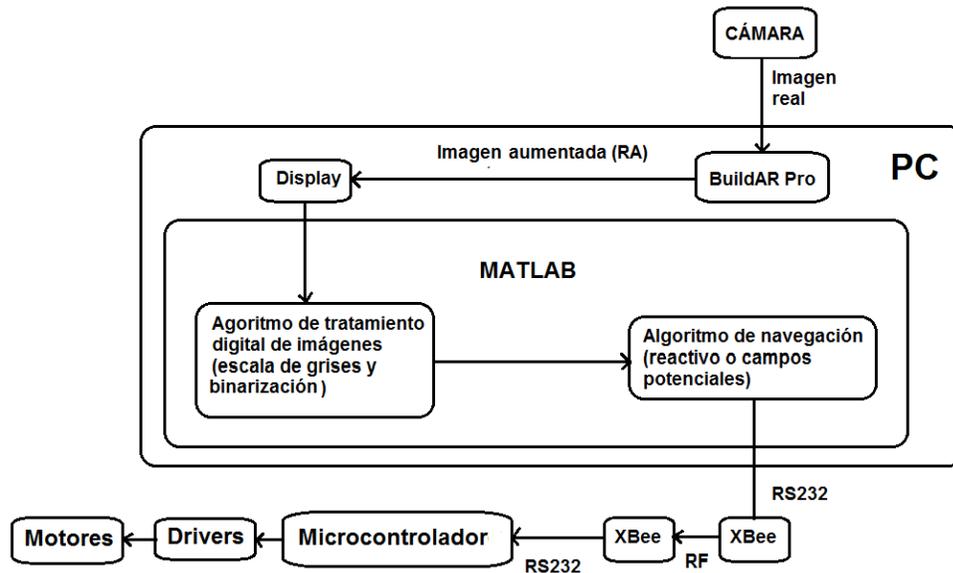
Fuente: Autor

2. MÉTODOS Y PROCEDIMIENTOS

Al momento de desarrollar el sistema de navegación de robots móviles empleando Realidad Aumentada y visión artificial se tuvieron en cuenta algunos de factores como: los modelos 2D cámara robot -2D del mundo, los diferentes algoritmos de navegación de robots compatibles con visión artificial, los posibles canales de comunicación entre el software de RA, el software de control y el robot, el tipo de iluminación para el funcionamiento del sistema, las características morfológicas de los marcadores empleados, etc. Igualmente, se implementaron 30 pruebas relacionadas con descriptores de forma de marcadores para estudiar la influencia de la forma en la detección de los marcadores de RA , 20 pruebas para estudiar la influencia de la iluminación en la detección de marcadores y el comportamiento de los mismos frente a cambios de escala, ángulo de registro y distancias de detección, 168 pruebas de navegación en ambientes de laboratorio, 128 en corredores y 20 en un laberinto simple, al igual que 20 pruebas en ambientes industriales. Todo esto ayudará a la comprensión del comportamiento de todos los elementos constitutivos del proyecto y su puesta a punto.

Un punto crítico en este proceso es la identificación del flujo de información en el sistema, lo cual en forma general comienza con el registro de la imagen del mundo real por parte de la cámara, que es tomada por el software de RA (*BuildAR*) para insertar los objetos virtuales y ser re proyectada en la pantalla del ordenador; posteriormente, el algoritmo de navegación toma dicha imagen e identifica el obstáculo, si lo hay por medio de un algoritmo básico de tratamiento digital de imágenes, emitiendo la señal correctiva según el tipo de algoritmo de navegación implementado (campos potenciales o reactivo). A continuación se muestra un diagrama general del flujo de información en el robot. Ver Figura 16.

Figura 16. Flujo de información en el robot



Fuente: Autor

Con esto se desglosarán los diferentes componentes y se analizarán por separado cada uno de ellos. Las pruebas y procedimientos para este fin se ilustrarán de forma separada a continuación.

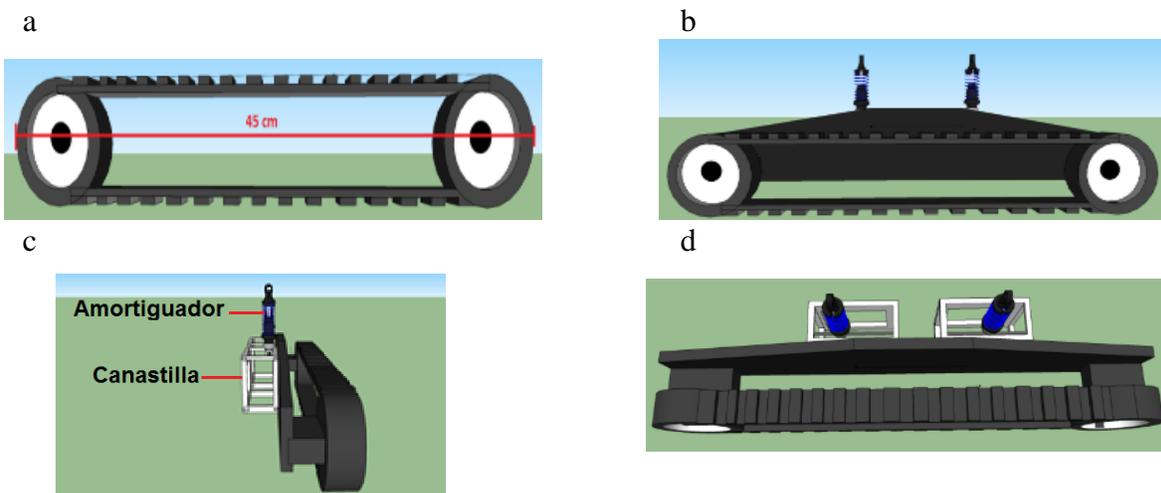
2.1. Arquitectura mecánica y electrónica de la plataforma robótica

La plataforma robótica utilizada fue suministrada por el grupo de investigación en Instrumentación, Control Automático y Robótica –ICARO- del Politécnico Jaime Isaza Cadavid. Esta consta básicamente de un chasis de acrílico y un sistema mecánico de locomoción por orugas. A esta plataforma básica se le agregaron otros componentes como: los sistemas de comunicación, de control, de potencia, de visión y procesamiento digital de imágenes, los cuales fueron diseñados e implementados por el autor.

2.1.1. Descripción del sistema locomotor

El sistema locomotor está constituido por un par de orugas simples situadas a cada lado del robot, estas orugas son impulsadas por cuatro servomotores *Futaba S3003* dos a cada lado, los cuales fueron modificados, para que funcionen como motorreductores, estos servos están sujetos cada uno a un paral lateral de un material plástico semirrígido. Dicho paral está anclado al chasis por medio de un par de canastillas oscilantes, que son estabilizadas con un par de amortiguadores que garantizan el máximo agarre a la superficie de desplazamiento. Ver Figura 17.

Figura 17. a) Oruga simple, b) Oruga, paral lateral y amortiguador vista lateral, c) Orugas montadas en paral frontal con canastillas y amortiguador vista frontal, d) Orugas montadas en paral frontal con canastillas y amortiguador vista superior.



Fuente: Autor

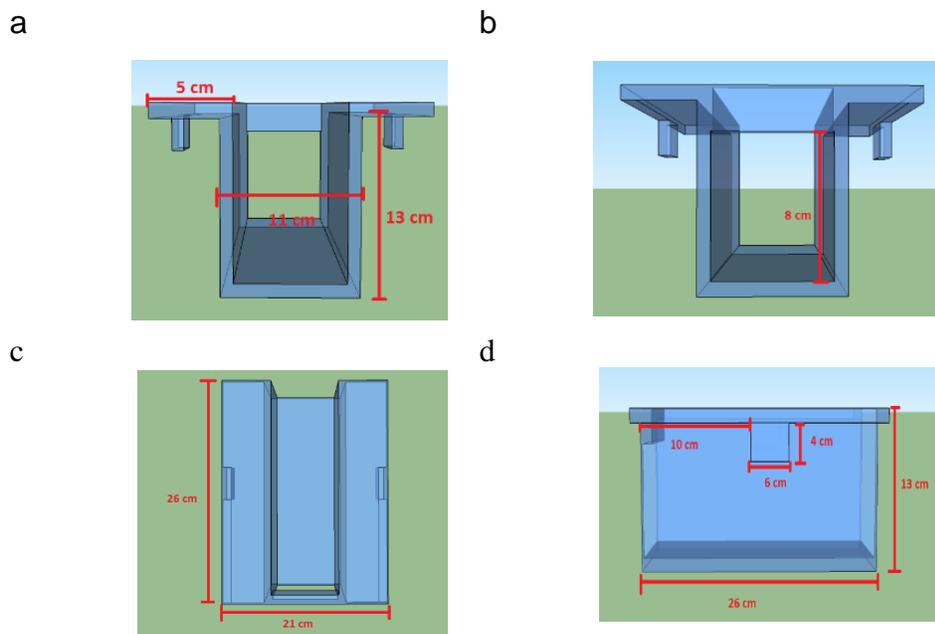
Dicha configuración con los servomotores permite que el robot tenga una capacidad de carga útil de 2 kg, la suficiente para soportar un computador tipo *notebook*, que se le instaló en su parte superior y en donde se implementaron los algoritmos de visión, navegación y RA. Adicionalmente, esta forma de locomoción garantiza que el robot pueda dar giros en su propio eje, pues las orugas

se comportan como un par de ruedas gigantes, es decir, si se quiere ir hacia adelante se direccionan las dos orugas hacia adelante, para ir a la derecha solo basta con direccionar la oruga derecha hacia atrás y la izquierda hacia adelante, y para que el robot se dirija a la izquierda basta con invertir el movimiento anterior.

En el caso de necesitar moverse alrededor de un obstáculo, el robot primero se moverá en su propio eje fijando el ángulo de desplazamiento y después avanzará hacia adelante o hacia atrás, esto se debe a que la arquitectura de los actuadores empleados en el robot no permite la variación de velocidad en las orugas, por lo cual no se puede contar con esto para realizar correcciones de dirección sobre la marcha.

El chasis está construido de un material acrílico translúcido de 0.5 cm de espesor; 22 cm de ancho por 13 cm de altura y 26 cm de profundidad, como se ilustra a continuación. Ver Figura 18.

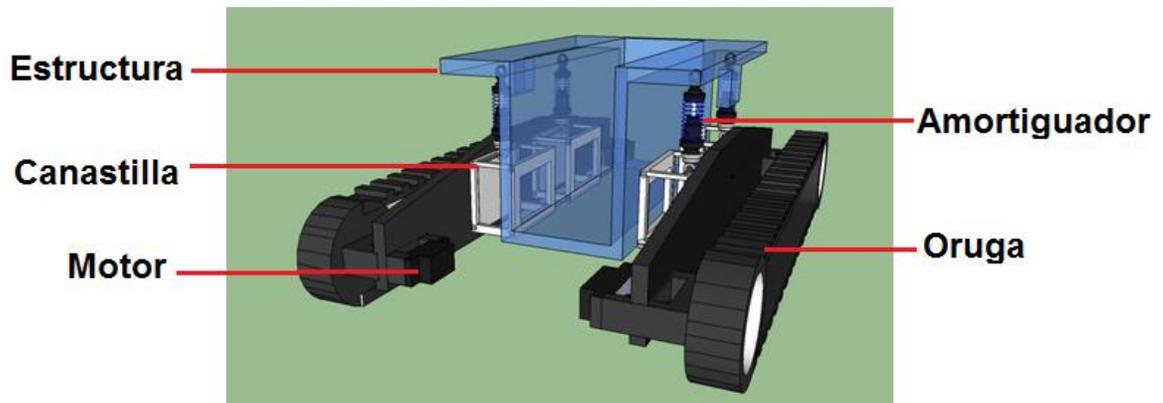
Figura 18. a) Chasis vista frontal, b) chasis vista trasera, c) chasis vista superior, d) chasis vista lateral.



Fuente: Autor

Uniendo todos los componentes anteriores se tiene la plataforma completa.

Figura 19. Plataforma completa



Fuente: Autor

2.1.2. Sistemas electrónicos de la plataforma

Los sistemas electrónicos de la plataforma están divididos en 3 circuitos:

- Circuito de interfaz de potencia
- Circuito de control
- Circuito de comunicaciones

Dichos circuitos fueron diseñados e implementados de forma modular, de modo que fuera fácilmente reemplazables en caso de algún percance. Esto también facilita la revisión individual de cada sección de los circuitos volviendo más eficiente la depuración de los mismos.

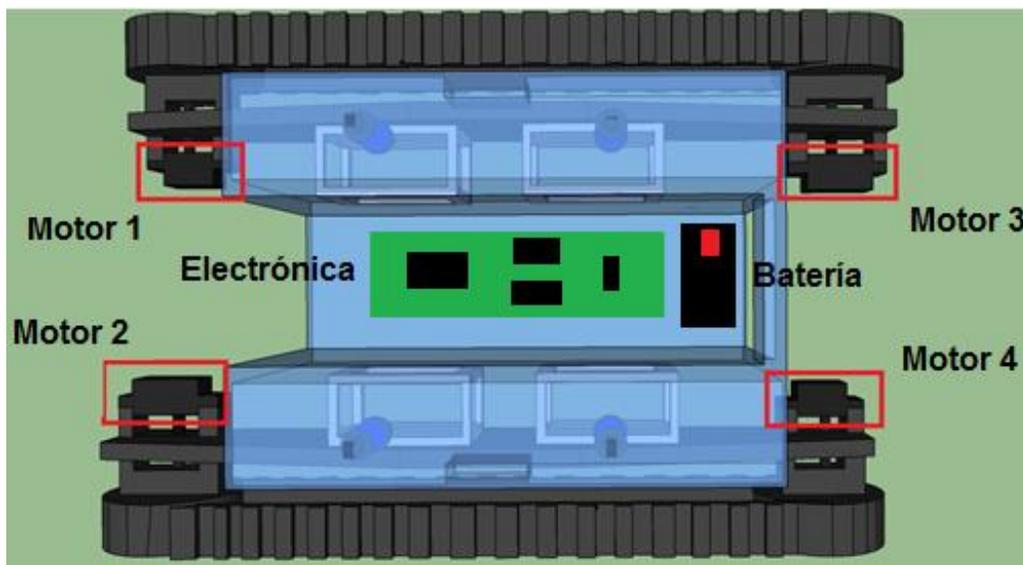
A continuación, se toma cada circuito describiendo sus componentes y la función que cada uno desempeña.

2.1.3. Sistemas de potencia

Basado en las características de funcionamiento de los servomotores que impulsan la plataforma y el hecho de que estos se comportan como motorreductores, se determinó que el elemento adecuado para gobernarlos corresponde al puente H L298.

La interfaz está compuesta por dos puentes H L298, los cuales gobiernan dos motorreductores cada uno. Asimismo posee dos resistencias variables para la regulación de velocidad y además tiene dos leds; el primero indica la presencia de voltajes de alimentación de los motorreductores, y corresponde a 6V, y el segundo indica la presencia del voltaje de funcionamiento de los integrados, en este caso 5V (Anexo 4). Cada puente h controla dos motores ubicados en las esquinas del robot.

Figura 20. Vista superior de la plataforma robótica



Fuente: Autor

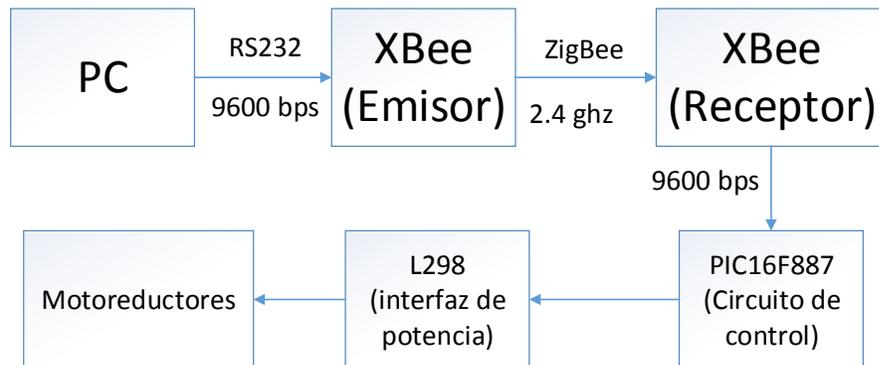
2.1.4. Circuito de control y comunicaciones

El circuito tiene como elemento principal un PIC16f887 con una frecuencia de procesamiento de 4 MHz, el cual se comunica con el circuito de interfaz de potencia por un *jumper* asociado al puerto B, y con el circuito de comunicaciones por un *jumper* asociado a los pines RC6 y RC7. Adicionalmente el circuito tiene un conector para la alimentación de trabajo de 5V, un led D1 que indica la presencia de voltaje de polarización y un pulsador *de reset*. También posee un puerto de programación *in-circuit* utilizado para programar el microcontrolador, sin necesidad de retirarlo del impreso (Anexo 5).

El circuito de comunicaciones está conformado por un XBee serie2, con un alcance de 100 metros a plena vista, el cual recibe los comandos de otro XBee de iguales características conectado a un computador. También se posee un regulador LD33 que reduce el voltaje de alimentación a 3.3V, adecuado para el XBee serie 2 con una capacidad de transferencia 9600 baudios; igualmente cuenta con un led D1, que indica la presencia de voltaje de alimentación y un led D3 que indica la presencia del voltaje regulado en el circuito. Por último, el jumper de comunicación lleva integrado un divisor de voltaje en el pin de recepción, de forma que el XBee no reciba un voltaje mayor a 3.3V ni menor a 1.5V (Anexo 3).

El robot carece de sistema de posicionamiento para registrar las trayectorias, esto se hará por medio de un sistema independiente al robot con el fin de que este se centre sólo en la generación de Realidad Aumentada, procesamiento de la imagen y navegación, con el fin de evitar la saturación del sistema de cómputo; el sistema de registro de trayectorias se tratará en detalle en el apartado 4.1. A continuación se muestra el diagrama de los sistemas electrónicos del robot.

Figura 21. Diagrama de flujo de sistemas electrónicos de la plataforma robótica



Fuente: Autor

2.1.5. Suministro de potencia

La potencia requerida para el funcionamiento de todos los elementos de la plataforma (cuyo consumo nominal en corriente es de 1.3 A) es suministrada por una batería seca de 6V a 4.5 Ah, la cual le da una autonomía de funcionamiento óptimo de alrededor de 3 horas a full carga. Los elementos que requieren 5V están conectados a un regulador LM7805, el cual estabiliza el voltaje a estos niveles.

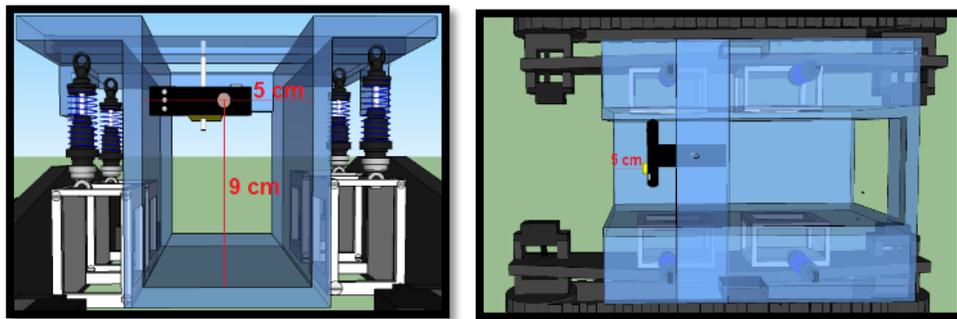
2.2. Caracterización de sistema de visión

En el desarrollo de este proyecto se pueden encontrar dos sistemas de visión: el primero implementado en el robot como herramienta de navegación; y el segundo en la parte superior del escenario de pruebas, el cual se implementó solo para evaluar la trayectoria del robot a través de este ambiente.

En el caso de los sistemas de visión en robots móviles, puede decirse que son de dos tipos: binoculares y monoculares. En este caso se decidió implementar este último, teniendo en cuenta su simplicidad en la implementación y la compatibilidad del mismo con los software de RA.

La cámara del robot está sujeta a un rectángulo de acrílico en la parte superior- frontal, a una altura de 9 cm de la placa inferior del chasis y a 5 cm de la placa lateral izquierda, de forma que el lente de la cámara estuviera centrada con respecto a al ancho del chasis; esto con el fin de mantener centrado el volumen de visión con respecto al robot.

Figura 22. Ubicación de cámara en el robot



Fuente: Autor

Con la cámara en su posición se obtuvieron dos parámetros claves del sistema de visión: el ángulo de apertura del lente de la cámara y la distancia mínima de visión. El primero se registró con la ayuda de un transportador ubicado en el lente, en el que se aprecia el ángulo de apertura enmarcado por los extremos de la imagen (40°). El segundo se midió con una regla métrica, ubicada al frente del lente de la cámara (Figura 23) y en la cual se registró el primer valor visible en la imagen transmitida a un PC (20 cm). Esto corresponde al primer paso en la obtención del volumen de visión, fundamental para comprender cómo el robot capta el mundo exterior. Igualmente permite fijar los límites preliminares del sistema de Realidad Aumentada dejando claridad sobre la ubicación de los marcadores de RA con respecto a la cámara del robot.

Figura 23. Registro de distancia mínima de visión

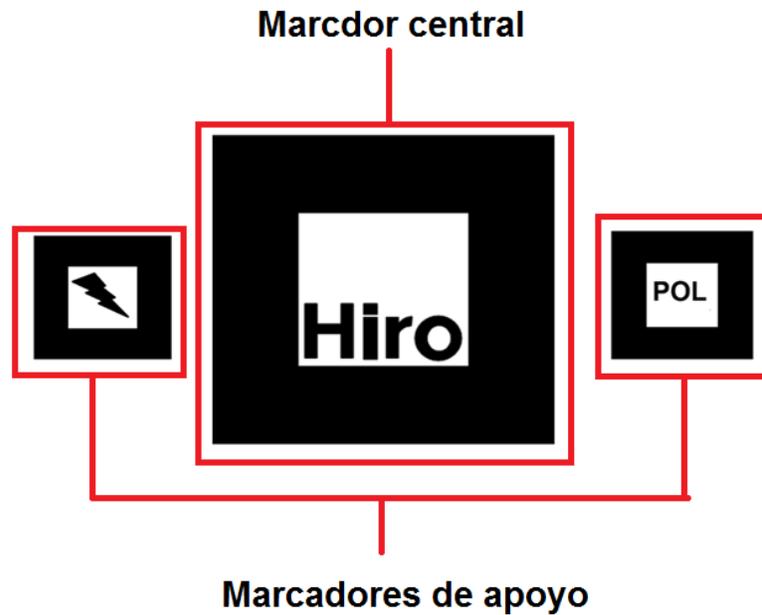


Fuente: Autor

2.2.1. Evaluación de marcadores

Esto se intentó corregir por medio de marcadores de respaldo que apoyan al marcador principal en caso de pérdida parcial o total del marcador principal. Dichos marcadores auxiliares se imprimieron con $\frac{1}{4}$ de escala y se ubicaron a cada lado del marcador principal, asociándoles a todos el mismo objeto (cubo negro), variando solo la escala y la posición, de modo que todos los modelos tuvieran el mismo tamaño y confluyeran en un solo centro fundiéndose en uno solo. El objetivo era lograr que, en el caso de que el marcador principal saliera de forma parcial o total del volumen de visión del robot, los marcadores auxiliares pudieran mantener el objeto virtual presente en la pantalla.

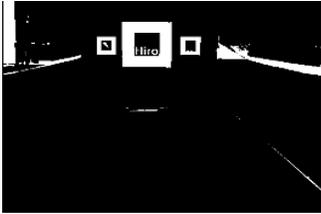
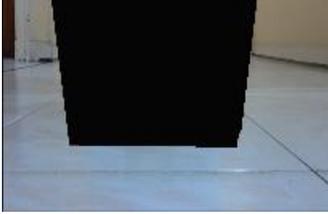
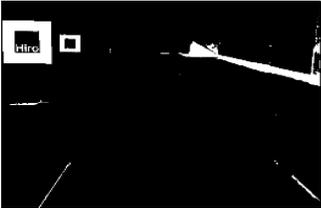
Figura 24. Disposición de marcador central y de apoyo



Fuente: Autor

Al aplicar esta técnica se logró que el objeto virtual se mantuviera presente con la pérdida total y/o parcial del marcador principal, con la ayuda de los marcadores auxiliares. En la siguiente tabla se aprecia lo que registra el software de RA en una imagen binarizada, y cómo a pesar de la pérdida del marcador principal se mantuvo el objeto virtual en la pantalla.

Tabla 4. Resultados obtenidos con los marcadores de apoyo

Marcadores presentes	Identificación de marcadores	Realidad Aumentada
3		
2		
1		

Fuente: Autor

2.2.2. Calibración de parámetros intrínsecos y extrínsecos de la cámara

Después de realizarse esta calibración experimental se obtuvieron los modelos para la transformación de las coordenadas cámara-mundo, tanto para la cámara del robot como para la cámara aérea que lo rastrea por el ambiente de pruebas. Esto se llevó a cabo con la ayuda del toolbox Calib del software MATLAB, que emplea una serie de imágenes de un patrón monocromático bidimensional (patrón de tablero de ajedrez) tomadas desde diferentes ángulos para estimar los parámetros que determinarán los modelos (Fetić & Jurić, 2012; Mejía et al, 2014; Portugal et al., 2009). Para esto se ubicó una fuente de luz blanca de 120 watts, a 1.7 mt de altura

sobre la mesa de pruebas. Asimismo, se dispuso la cámara sobre un soporte metálico a una distancia de 80 cm del tablero de ajedrez, dicha cámara esta conectada a un ordenador que se encarga de registrar las imágenes utilizadas en la calibración; en cada imagen se altera la posición y el ángulo del patrón con respecto la cámara, conservando en todo momento los 80 cm ya establecidos. El montaje para obtener las imágenes se puede apreciar en la Figura 25.

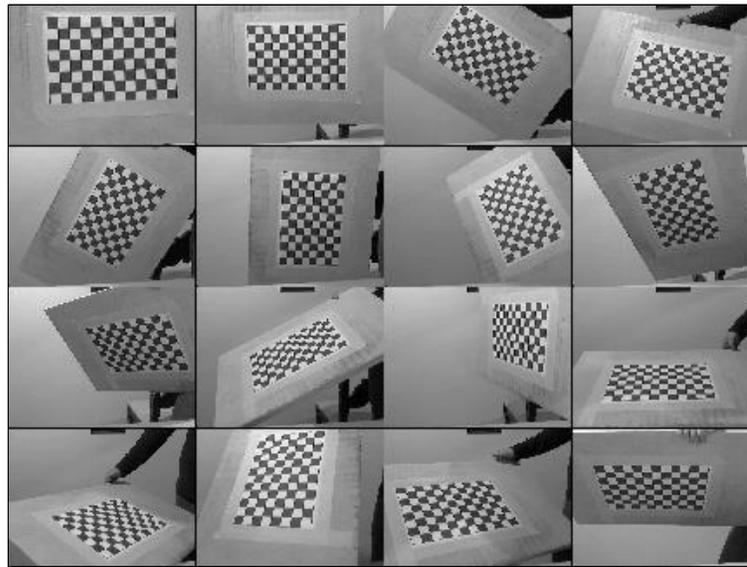
Figura 25. Montaje para la obtención de las imágenes de calibración



Fuente: Autor

Las imágenes para la calibración obtenidas de las dos cámaras (del robot y aérea) son las siguientes:

Figura 26. Imágenes de calibración de la cámara frontal del robot



Fuente: Autor

Aplicando el *Toolbox Calib* de MATLAB a los dos conjuntos de imágenes se obtienen los siguiente parámetros:

- Focal length (f_c): distancia focal
- Principal points (c_c): puntos principales
- Skew (α_c) : coeficiente de asimetría que define el ángulo entre los pixeles del X y del eje Y
- Distortion (K_c): coeficientes de distorsión de la imagen (radial y tangencial)
- Pixel error (err): error de estimación de los pixeles

Tabla 5. Parámetros intrínsecos cámara del robot

Distancia focal	$fc = [2823.09391 \ 2822.27427] \pm [35.21089 \ 36.48657]$
Puntos principales	$cc = [975.84165 \ 786.27499] \pm [61.06188 \ 50.74250]$
Coefficiente de asimetría	$\alpha_c = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
Distorsión	$kc = [0.13908 \ -0.39973 \ 0.01080 \ -0.00793 \ 0.00000] \pm [0.06959 \ 0.68340 \ 0.00778 \ 0.00979 \ 0.00000]$
Error de pixel	$err = [2.26937 \ 1.41767]$

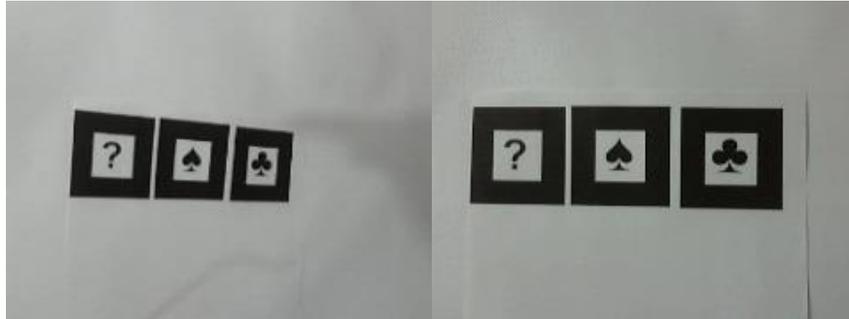
Fuente: Autor

Teniendo los parámetros de la tabla anterior, se calcularon las matrices de los parámetros intrínsecos de cada cámara del robot (KK).

$$KK = \begin{bmatrix} 2823.09391 & 0 & 975.84165 \\ 0 & 2822.27427 & 786.27499 \\ 0 & 0 & 1 \end{bmatrix}$$

La validación de dicha KK se puede hacer re proyectando un objeto 3D simple, como un cubo o un recuadro en una imagen captada por la cámara, y verificando si es fiel a la perspectiva. Esto se realiza partiendo del mundo 2D de la cámara y extrapolando el objeto al mundo 3D real, utilizando como base una imagen real captada por dicha cámara. Las imágenes empleadas para realizar dicha re proyección se pueden apreciar en la figura 27.

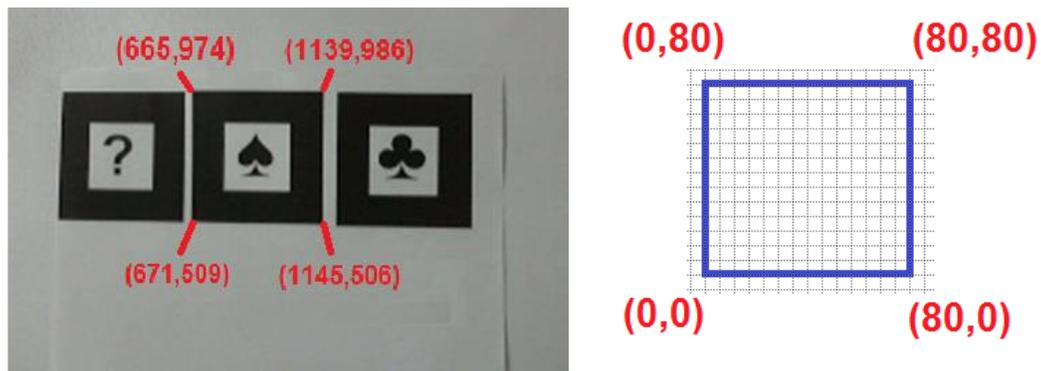
Figura 27. Imágenes de referencia para modelación y validación de modelo de la cámara del robot



Fuente: Autor

Para realizar los cálculos se tomaron las esquinas del marcador Pica en cada imagen, extrayendo sus coordenadas (i, j) en función de los píxeles que lo conforman; posteriormente, basándose en el tamaño del marcador en el mundo real (80x80 mm), se fijaron los vértices de un recuadro (x, y) , que será reproyectado sobre el marcador Pica.

Figura 28. Ejemplo de extracción de coordenadas de vértices del marcador *pica* y del recuadro que se reproyectará en la imagen



Fuente: Autor

Al aplicar la extracción de coordenadas a las dos imágenes de la Figura 28, se obtiene la siguiente tabla.

Tabla 6. Vértices del marcador Pica en las dos imágenes de la Figura 29 (i, j) y los vértices del recuadro que se quiere reproyectar sobre ellas (X, Y.)

Imagen 1	Posición (i, j) (píxeles)	Posición (X,Y) (milímetros)
	(671,509)	(0,0)
	(665,974)	(0,80)
	(1139,986)	(80,80)
	(1145,506)	(80,0)
Imagen 2	(740,560)	(0,0)
	(710,950)	(0,80)
	(1043,956)	(80,80)
	(1070,590)	(80,0)

Fuente: Autor

Basándose en lo anterior, se obtiene la matriz de transformación H:

Matriz H para imagen 1

$$H = \begin{bmatrix} -0.0063 & 0.0001 & -0.7967 \\ 0.0003 & -0.0067 & -0.6043 \\ 0 & 0 & -0.0012 \end{bmatrix}$$

Matriz H para imagen 2

$$H = \begin{bmatrix} 0.0053 & -0.0004 & 0.7974 \\ 0.0009 & 0.0051 & 0.6034 \\ 0 & 0 & 0.0011 \end{bmatrix}$$

Empleando MATLAB se obtiene la matriz de rotación Rt:

Matriz Rt Imagen 1

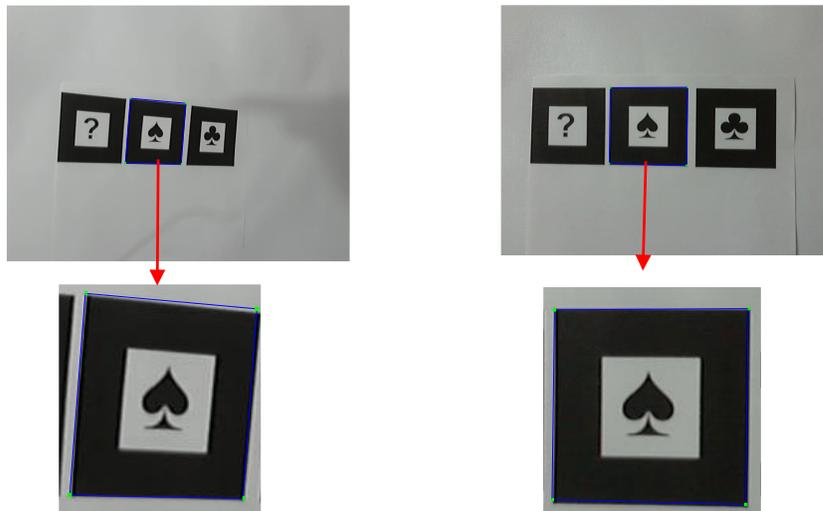
$$Rt = \begin{bmatrix} -0.9685 & 0.0154 & 57.5426 \\ 0.0461 & -1.0303 & 52.1649 \\ 0 & 0 & -520.7926 \end{bmatrix}$$

Matriz Rt Imagen 2

$$Rt = \begin{bmatrix} 1.0102 & -0.0762 & -52.6111 \\ 0.1716 & 0.9724 & -49.8574 \\ 0 & 0 & 591.8969 \end{bmatrix}$$

Para comprobar la validez de la transformación lineal, empleando la matriz H, se recalculan los valores de los vértices de la marca Pica en ambas imágenes y se grafican sobre ellas en color azul.

Figura 29. Imágenes con re proyección de esquinas



Fuente: Autor

Esta reproyección muestra que el recuadro azul es fiel a la perspectiva de la cámara, pues los vértices del recuadro reproyectado coinciden con los vértices del recuadro que rodea al marcador Pica, validando los parámetros obtenidos en KK y por consiguiente su calibración en 2D. Para verificar estos mismos parámetros, en una reproyección de un cubo 3D, se calcula la matriz de proyección P de la cámara del robot en cada una de las dos imágenes del marcador Pica.

La matriz P para la imagen 1 es:

$$P = \begin{bmatrix} -2626.468483 & 37.53620702 & 1480.648186784 & 32976.6392010 \\ 112.7200152 & -2784.601768 & 785.8411288678 & 250150.6610459 \\ 0.187233101 & 0.00237003 & 0.98259786185 & -491.455129698 \end{bmatrix}$$

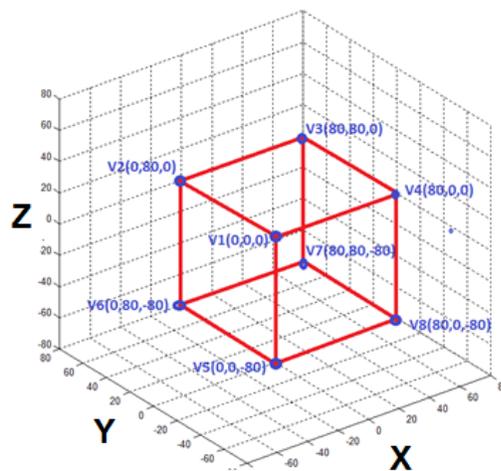
La matriz P para la imagen 2 es:

$$P = \begin{bmatrix} 2915.93530989 & -237.644081973 & -5.06.0907962 & 442094.71420 \\ 500.702055332 & 2815.89160918 & 653.03879321 & 334558.161997 \\ 0.478189536 & -0.026864582 & 0.875434270 & 597425289391 \end{bmatrix}$$

Como se observa, las matrices de reproyección P de la cámara varían de la imagen 1 a la imagen 2, debido al cambio de posición de esta con respecto al marcador Pica. Esto implica, que sí se quisiera insertar este recuadro en una serie de imágenes o un video, debe calcularse una matriz P para cada fotograma o imagen y así asegurar su correcta inserción.

Para validar todo los parámetros obtenidos (K, H, Rt, P) se reproyectará un elemento 3D (cubo) en vez de uno 2D (recuadro). Cada lado de dicho cubo tendrá las mismas dimensiones del recuadro empleado en la prueba inicial (80x80 mm), con el fin de que la cara inferior se alinee con el marcador Pica, y así poder tener un punto de referencia para verificar que la reproyección sea fiel a la perspectiva de la cámara. Para ello se presentarán las coordenadas del mundo para dicho cubo.

Figura 30. Vértices de cubo reproyectado en las coordenadas del mundo



Fuente: Autor

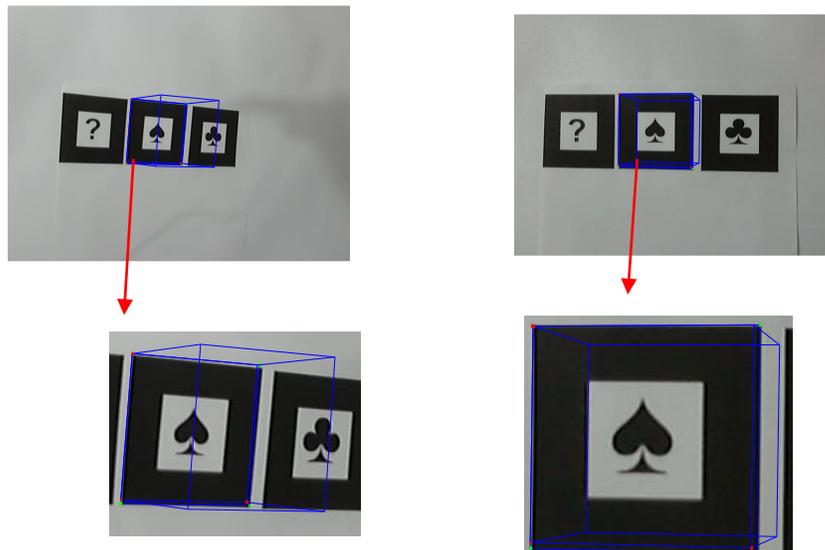
Tabla 7. Puntos de referencia de las imágenes para reproyección de cubo

Vértices del cubo	Coordenadas en el mundo (mm)
Vértice 1	[0 0 0]
Vértice 2	[0 80 0]
Vértice 3	[80 80 0]
Vértice 4	[80 0 0]
Vértice 5	[0 0 -80]
Vértice 6	[0 80 -80]
Vértice 7	[80 80 -80]
Vértice 8	[80 0 -80]

Fuente: Autor

Los resultados de la inserción del cubo se muestran a continuación:

Figura 31. Imágenes con proyección de cubo



Fuente: Autor

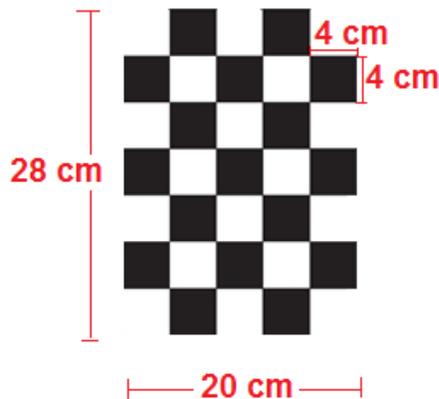
Se puede apreciar que las aristas del cubo que se sobrepuso al marcador son coherentes con la perspectiva, validando los parámetros obtenidos y corroborando la efectividad de la calibración.

2.2.3. Estimación de error de la Realidad Aumentada con respecto al mundo real

Una de las dificultades, al momento de trabajar con Realidad Aumentada, es conocer el error que se presenta al tratar de representar objetos reales con objetos virtuales insertados por la RA. Esto cobra mayor importancia cuando es más de un objeto que se quiere insertar, o cuando se tiene un escenario integrado por objetos reales y virtuales, como es el objetivo de este trabajo.

Para poder establecer dicho error, de forma cuantitativa, se optó por utilizar los parámetros intrínsecos de la cámara del robot, ya calculados, y calcular los de la cámara virtual; esta posiciona los elementos 3D en la RA. Para ello, se empleó un patrón de calibración real de tablero de ajedrez (Figura 32) y se diseñó una copia virtual en un software CAD, conservando sus dimensiones exactas, de modo que cuando sea insertado por el software de RA en las imágenes de calibración conserve la escala precisa para representar el patrón de calibración real.

Figura 32. Patrón de calibración real

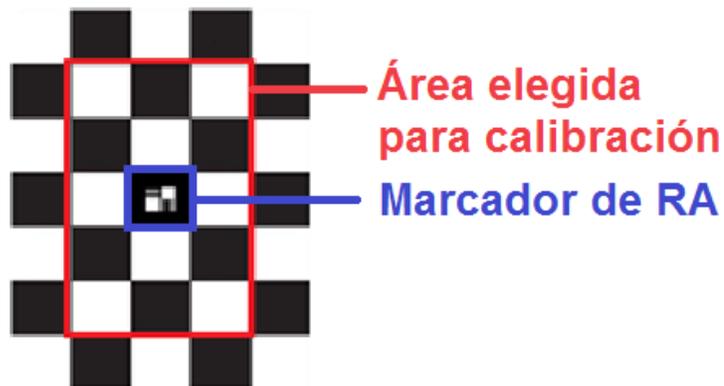


Fuente: Autor

El patrón de calibración virtual se insertó en las imágenes de calibración, utilizando el software de RA e integrando un marcador a la parte central del patrón real, conservando las condiciones de distancia y variación de posición para los dos patrones (Figura 33). Con esto se busca obtener una matriz de errores que refleje el desfase entre los objetos reales y los re proyectados por el software de RA *BuildAR pro*.

Para tomar las imágenes de calibración se conservaron las mismas características del ambiente de calibración de la cámara del robot. También hay que aclarar que para obtener los parámetros de calibración, en este caso no se tomó en cuenta la totalidad del patrón, sino solo un área de 5x3 cuadrados en la sección interna, para evitar el problema de pérdida de las esquinas en la imagen.

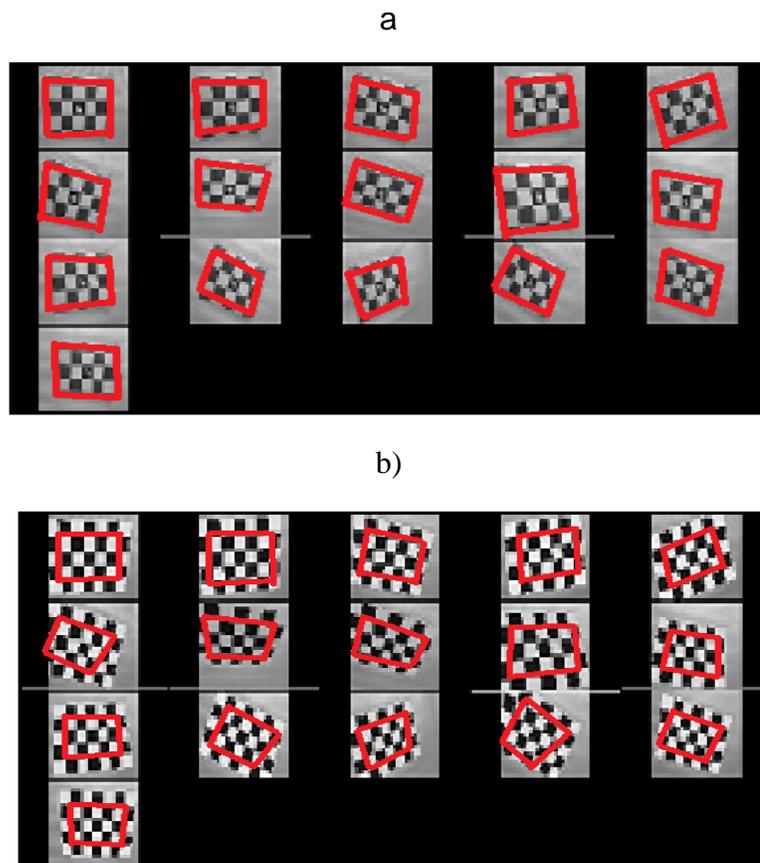
Figura 33. Patrón de calibración con inserción de marcador de RA y delimitación de área elegida para la calibración (encerrada en recuadro rojo)



Fuente: Autor

Las imágenes originales de la prueba de calibración fueron utilizadas para obtener los parámetros de la cámara física del robot y posteriormente se insertó la RA en la imagen y se obtuvieron los parámetros de la cámara virtual.

Figura 34. Imágenes de calibración a) imágenes cámara real b) imágenes con RA



Fuente: Autor

Después de realizar el proceso de calibración se obtuvieron los siguientes parámetros:

Tabla 8. Parámetros intrínsecos cámara del robot

Distancia focal	fc = [1087.05547 1085.37688] ± [12.28062 12.62137]
Puntos principales	cc = [664.74004 285.66282] ± [11.64882 7.39761]
Coefficiente de asimetría	alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distorsión	kc = [-0.00269 -0.09811 -0.00654 -0.00783 0.00000] ± [0.04079 0.29939 0.00247 0.00337 0.00000]
Error de pixel	err = [0.33100 0.53965]

Fuente: Autor

Tabla 9. Parámetros intrínsecos de RA

Distancia focal	fc = [912.01678 944.94379] ± [4.03723 4.37670]
Puntos principales	cc = [681.69049 314.73249] ± [4.09202 2.95802]
Coefficiente de asimetría	alpha_c = [0.00000] ± [0.00000] => angle of pixel axes = 90.00000 ± 0.00000 degrees
Distorsión	kc = [0.02424 -0.05054 -0.00113 0.00527 0.00000] ± [0.01355 0.07326 0.00103 0.00163 0.00000]
Error de pixel	err = [0.16882 0.21787]

Fuente: Autor

Para cada conjunto de parámetros se calcula la matriz KK.

Modelo cámara real

$$\begin{bmatrix} X_p \\ Y_p \\ 1 \end{bmatrix} = KK \begin{bmatrix} X_d(1) \\ X_d(2) \\ 1 \end{bmatrix} \quad KK_R = \begin{bmatrix} 1087.05547 & 0 & 664.74004 \\ 0 & 1085.37688 & 285.66282 \\ 0 & 0 & 1 \end{bmatrix}$$

Modelo cámara virtual

$$\begin{bmatrix} Xp \\ Yp \\ 1 \end{bmatrix} = KK \begin{bmatrix} X_d(1) \\ X_d(2) \\ 1 \end{bmatrix} KK_V = \begin{bmatrix} 912.01678 & 0 & 681.69049 \\ 0 & 944.94379 & 314.73249 \\ 0 & 0 & 1 \end{bmatrix}$$

Para estimar qué tan diferentes son los dos modelos (RA y real) se realizó una matriz de errores, conformada por los errores presentes en la estimación de los componentes de la matriz KK de la cámara de la Realidad Aumentada con la real.

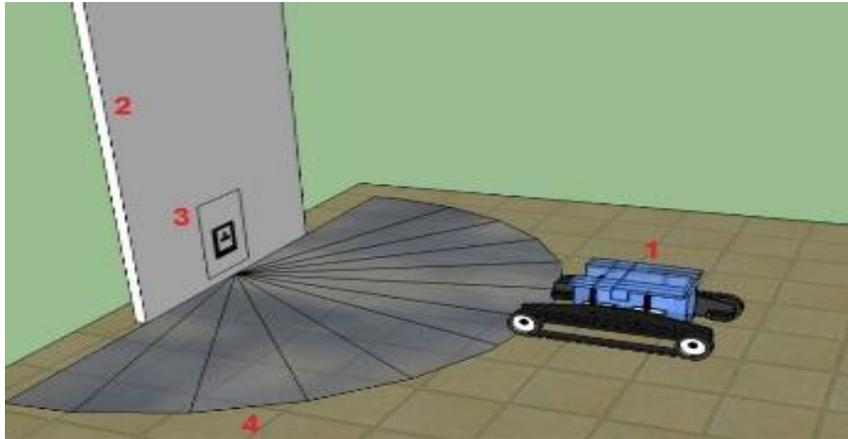
$$ME\% = \begin{vmatrix} 16.10 & 0 & 2.54 \\ 0 & 12.93 & 10.17 \\ 0 & 0 & 0 \end{vmatrix}$$

Basado en el promedio aritmético de los errores presentes en cada elemento de la matriz ME, se puede decir que el error presente en la estimación de los parámetros intrínsecos de la cámara virtual posicionada por la Realidad Aumentada es de 10.43%, esto implica que tanto la escala del objeto insertado por la RA como su ubicación y orientación, con respecto a la cámara, puede presentar este error si se compara con una imagen de un objeto real tomada bajo estas mismas condiciones.

2.2.4. Ángulos y distancias de detección de marcadores.

Para obtener los ángulos mínimos de detección se fijaron los marcadores en un telón blanco de 115x79 cm y se ubicó el robot a una distancia de 90 cm.; se eligió esta distancia pues desde ella se aprecia de forma clara el cubo virtual insertado por la RA, además esta distancia es fácil de medir debido a que el piso está conformado por baldosas de 30x30 cm, facilitando la toma de medidas y la ubicación del robot en esta prueba. Para iniciar, se ubicó el robot de forma perpendicular al marcador y se fue desplazando alrededor, desde 0° a 180°, medidos con respecto a la superficie del telón, conservando la distancia de 90 cm.

Figura 35. Escenario y parámetros de medición de ángulos críticos de trabajo con múltiples marcadores 1) Robot, 2) Telón, 3) Marcador, 4) semicírculo que delimita el posicionamiento del robot



Fuente: Autor

La siguiente tabla muestra los resultados obtenidos en función de los ángulos límites de operación para cada marcador de la prueba.

Tabla 10. Ángulos mínimos de detección de marcadores

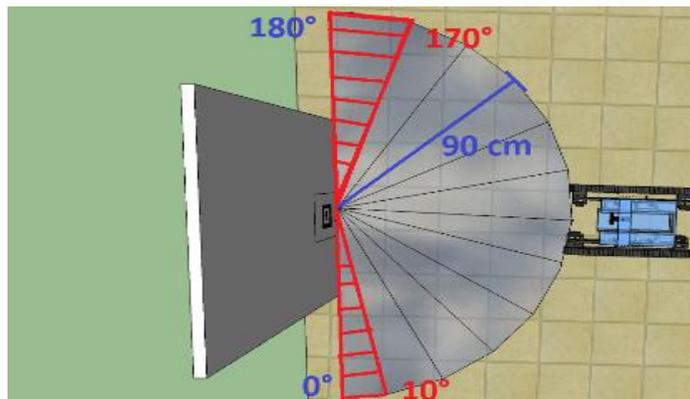
MARCADOR	ÁNGULO LÍMITE DE DETECCIÓN DE MARCADORES MEDIDO ÁNGULO IZQUIERDO (GRADOS)	ÁNGULO LÍMITE DE DETECCIÓN DE MARCADORES MEDIDO ÁNGULO DERECHO
	4	176
	10	170
	8	172
	5	175
	10	170

Fuente: Autor

Esta tabla fija los ángulos mínimos de incidencia que debe tener el robot sobre los marcadores para que el software de RA pueda detectarlos de forma óptima. Es decir, para asegurar el buen funcionamiento del sistema se debe propender porque los marcadores estén en un ángulo de visión con respecto a la cámara del robot de entre 10° y 170°, de otro modo no se garantiza la

identificación de los marcadores por parte del software de RA. Se concluye que los marcadores no tienen el mismo desempeño y esto se debe tener presente en su uso.

Figura 36. Representación de los ángulos críticos de visión (en rojo) entre los cuales no son detectados los marcadores



Fuente: Autor

Por último, en la tercera prueba se buscó establecer las distancias mínimas y máximas de detección de marcadores. Para ello se conservaron las mismas características de los ambientes de pruebas anteriores, ubicando el marcador a la misma altura de la cámara situada en la parte delantera del robot móvil, es decir, a 14 cm de altura del piso, de modo que se pueda alinear de forma perpendicular al robot.

Figura 37. Robot frente a marcador



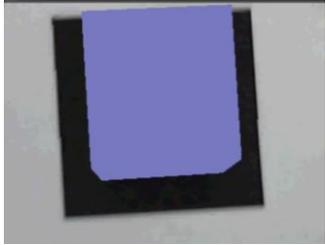
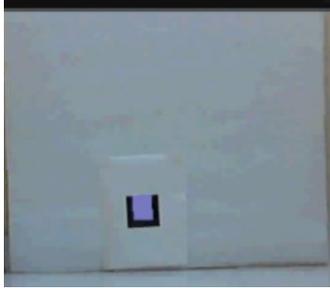
Fuente: Autor

Ya hecho esto, se procede a alejar paulatinamente el robot hasta que no se aprecie el objeto vinculado al marcador.

Para esta prueba se evaluaron dos software de Realidad Aumentada, de los más usados, como son el *BuildAR* y el *BuildAR pro*, para tener una perspectiva más amplia de los rangos de operación de este parámetro.

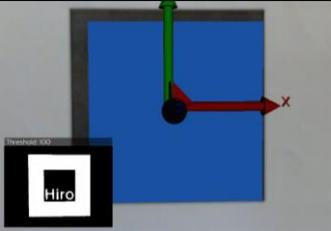
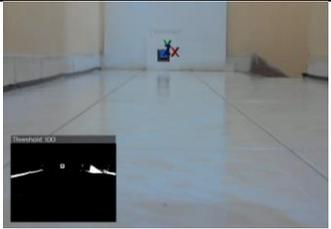
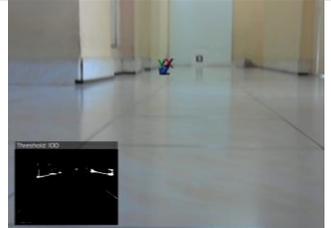
Los resultados obtenidos empleando el marcador Hiro se pueden ver en las siguientes tablas.

Tabla 11. Distancias de operación empleando *BuildAR*

DISTANCIA (cm)	VISUALIZACIÓN	ESTADO DEL OBJETO 3D
20		Presente en la imagen
89		Presente pero oscila por la vibración de la cámara
338		No hay presencia del objeto 3D en la imagen

Fuente: Autor

Tabla 12. Distancias de operación empleando *BuildAR Pro*

DISTANCIA (cm)	VISUALIZACIÓN	ESTADO DEL OBJETO 3D
20		Presente en la imagen
338		Presente con oscilaciones mínimas en el modelo por la vibración de la cámara
650		Pérdida definitiva del objeto 3D y presencia de falsos positivos

Fuente: Autor

En ambos software, se observó que las vibraciones del robot causadas por su parte mecánica y por las irregularidades de la superficie por donde se desplaza afectan la identificación del marcador, pues desestabilizan el auto foco de la cámara y en algunas ocasiones difuminan la imagen.

Al final de la prueba, se obtuvo que la distancia máxima obtenida bajo los anteriores parámetros fue de 338 cm para el *BuildAR* y 650 cm para el *BuildAR Pro*, después de los cuales no fue visible el objeto 3D. Por otro lado, se observó que a una distancia mayor de 89 cm para el *BuildAR* y 127 cm para el *BuildAR Pro* el objeto virtual se desestabiliza levemente, es decir, aparece intermitentemente y presenta una vibración leve con el robot estático.

Cabe anotar que después de la pérdida de los objetos 3D se invirtió el proceso acercando el robot a la marca para medir la distancia de recuperación del objeto 3D, obteniendo 600 cm para *el BuildAR Pro* y 301 cm para el *BuildAR*.

Después de lo anterior, se decidió realizar una segunda prueba con el fin de establecer la distancia mínima de operación del sistema. Para este caso se cambió el cubo que trae el software por defecto por un cubo más grande y así facilitar el análisis cuando la cámara penetra un objeto virtual.

Para esta prueba se mantuvieron los parámetros básicos de la prueba inicial, con la diferencia de que se situó al robot a una distancia de 1 m y se fue acercando al marcador hasta que desapareció el modelo 3D.

Tabla 13. Acercamiento del robot al marcador desde 1mtr hasta 17 cm

DISTANCIA (cm)	VISUALIZACIÓN	ESTADO DEL OBJETO 3D
100		Visualización completo del objeto 3D
20		Visualización interna del cubo
17		Pérdida del objeto 3D

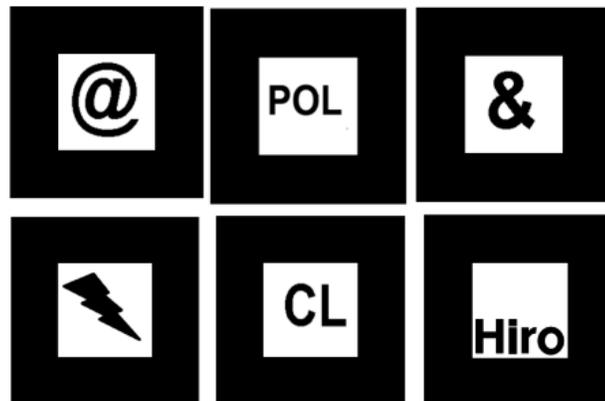
Fuente: Autor

Con este procedimiento se pudo concluir que la distancia mínima a la cual debe estar el robot de un marcador de tamaño estándar (8x8 cm) para que la Realidad Aumentada funcione eficientemente es de 17 cm. Esto debido a que el volumen de visión del marcador es demasiado estrecho para abarcarlo a distancias inferiores, además, teniendo en cuenta que el robot mide 28 cm de largo y que este gira sobre su propio eje necesitando 14 cm para ello, esta distancia es suficiente para que pueda reaccionar y evadir el obstáculo.

Si se requiere que el robot trabaje por debajo de esta distancia, una solución es disminuir la escala del marcador y aumentar la escala del objeto virtual, con lo que se conservaría la integridad del objeto que se quiere representar.

La misma metodología de las pruebas anteriores se aplicó a un conjunto de marcadores simples (Figura 38) con el fin de tener una vista más general del efecto de la forma en los ángulos y distancias de detección.

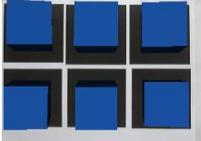
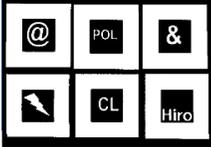
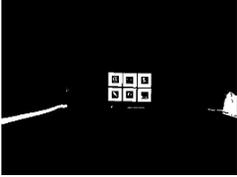
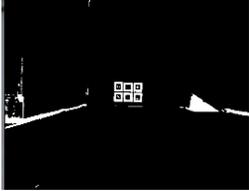
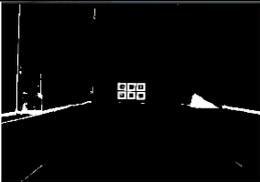
Figura 38. Marcadores empleados en la prueba



Fuente: Autor

Los resultados obtenidos se muestran en la Tabla 14.

Tabla 14. Distancia y estados de objetos 3D

DISTANCIA (cm)	VISUALIZACIÓN	BINERIALIZACIÓN	ESTADO DEL OBJETO 3D
45			Cubos presentes en todos los marcadores
214			Fallas intermitentes de ubicación de los cubos
365			Pérdida intermitente de los cubos 3D en los marcadores POL y CL
387			Pérdidas esporádicas de los cubos 3D, en los marcadores POL, CL y HIRO
430			Pérdidas esporádicas de los objeto 3D, en los marcadores POL, CL, HIRO y @
465			Aparición de falsos positivos y pérdidas esporádicas de los objeto 3D, en los marcadores: POL, CL, HIRO, @ y &

626			Incremento de falsos positivos y pérdida casi total de los objetos 3D en los marcadores POL, CL,HIRO, @ y &
648			Pérdida de todos los elementos virtuales en todos los marcadores

Fuente: Autor

Por otro lado las pruebas basadas en descriptores geométricos estudiaron la influencia de la forma del marcador en su desempeño por medio del análisis de descriptores de forma tales como: porcentaje de área convexa, porcentaje de área simple, número de elementos y solidez.

2.2.5. Volumen de visión

El primer paso para obtener el volumen de visión es obtener los marcos de visión, lo que no es más que un conjunto de recuadros, cuyas dimensiones están definidas por la cantidad de centímetros que puede abarcar la cámara en una imagen a una distancia específica, tanto de forma vertical como horizontal. Estos se obtuvieron utilizando una cinta métrica dispuesta de forma paralela a la parte delantera del robot para apreciar la extensión de la componente en el eje “X” del cuadro de visión; para el eje “Y” se ubicó en una regla escazada en cm de forma perpendicular al patrón anterior. Para todos los planos de visión se siguió el mismo procedimiento.

Figura 39. Proceso de medición de cada marco de visión



Fuente: Autor

Para obtener este volumen se realizaron 5 medidas del marco de visión en intervalos de 30 cm, a partir de la parte delantera del robot.

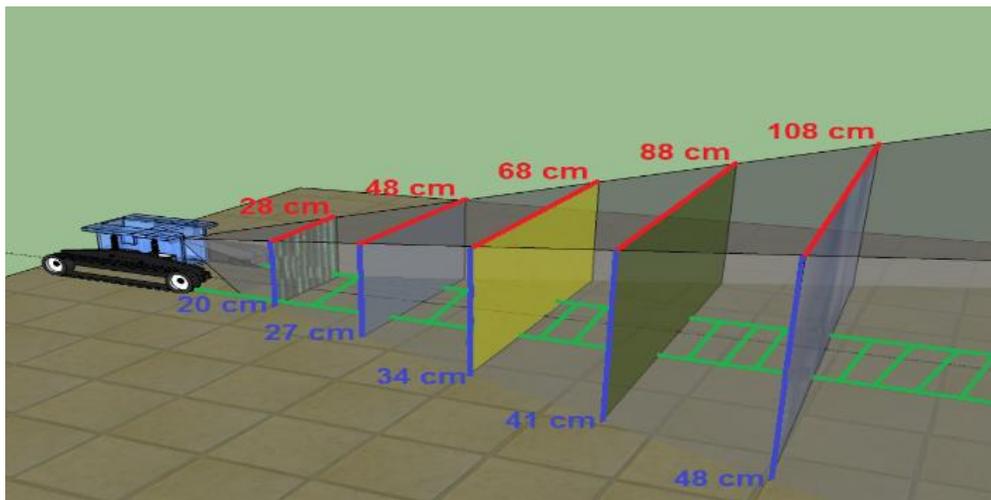
Tabla 15. Dimensiones de los marcos de visión obtenidos

Distancia de la parte frontal del robot a la muestra (cm)	Extensión del marco de visión en el eje X (cm)	Extensión en el marco de visión en el eje Y (cm)
30	28	20
60	48	27
90	68	34
100	88	41
120	108	48

Fuente: Autor

Al agrupar los marcos de visión se obtiene un volumen dentro del cual deben estar ubicados los marcadores para poder ser detectados por el robot. Este volumen define los tamaños de los objetos que se pueden representar según las distancias.

Figura 40. Volumen de visión construido con los marcos de visión obtenidos experimentalmente



Fuente: Autor

Teniendo en cuenta la figura anterior, es necesario considerar los puntos ciegos para fijar los márgenes de seguridad para la navegación. El volumen de visión revela la incapacidad del robot de ver los marcadores de RA cuando se encuentran demasiado cerca y/o fuera de los marcos de visión, lo que causa errores en la representación de los obstáculos.

2.3. Pruebas de Realidad Aumentada y marcadores

En la actualidad existen muchas aplicaciones de RA, pero entre los ambientes de desarrollo más usados se encuentran los software: *ARToolKit* y *BuildAR*. El primero es un conjunto de librerías de C++, las cuales sirven para la implementación de aplicaciones de Realidad Aumentada en equipos de cómputo bajo la licencia GNU *General Public License* (GPL). El segundo es un software de

Realidad Aumentada para el sistema operativo *Windows*, su plataforma es muy amigable y fácil de usar, con una distribución gratuita, la cual permite realizar pruebas de Realidad Aumentada, insertar nuevos marcadores y nuevos objetos 3D. No se requiere conocimientos específicos de programación como en el *ARToolKit*, aunque en su versión gratuita tiene algunas limitaciones.

Para la identificación del software de Realidad Aumentada se evaluaron unos parámetros considerados relevantes y se le dio una calificación de 0 a 5 a cada uno, según el criterio del investigador.

Los criterios escogidos fueron los siguientes:

- Estabilidad: se dice que un sistema es estable cuando su nivel de fallos disminuye por debajo de un determinado umbral, en este caso se consideró el número de fallos obtenidos en las pruebas realizadas de uno y otro software. En este caso, también, se pueden definir un fallo como la identificación errónea de marcadores, oscilaciones y pérdidas del objeto virtual a corta distancia del marcador.
- Multiplataforma: es la capacidad de ser instalado un software en más de un sistema operativo.
- Accesibilidad al código fuente: es la capacidad de acceder o de descomponer un software en el lenguaje de programación que fue escrito, por ejemplo C++, Basic, Assembler, etc.
- Bajo costo: hace alusión al costo de la licencia, en este caso entre más bajo sea el costo mejor.
- Experiencia del programador: el conocimiento que el usuario del software de Realidad Aumentada debe tener en el uso específico de un lenguaje de programación para poder implementar sus aplicaciones.
- Documentación disponible: se refiere a la existencia y fácil acceso de la información referente al software como tutoriales, manuales de usuario y características técnicas.
- Entorno amigable: es la facilidad de uso del entorno de programación o desarrollo del software.

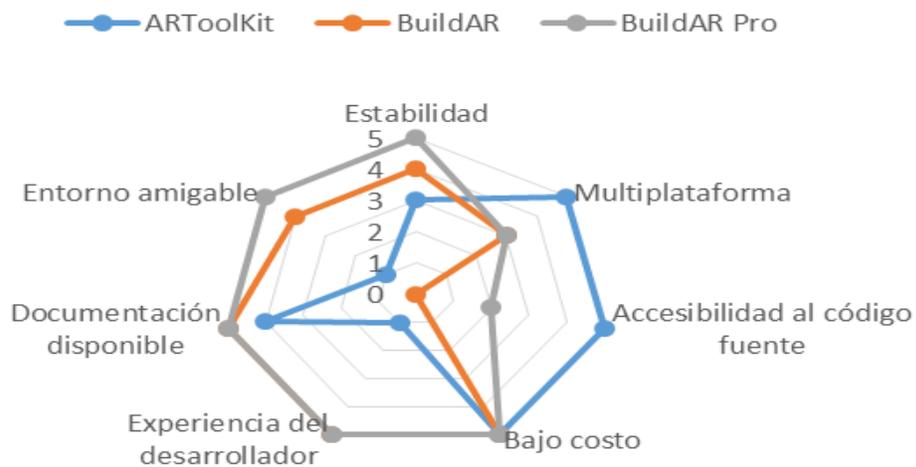
Los resultados obtenidos de la aplicación de los criterios anteriores arrojaron los siguientes resultados:

Tabla 16. Evaluación de software de RA

PARÁMETRO	<i>ARToolKit</i>	<i>BuildAR</i>	<i>BuildAR Pro</i>
Estabilidad	3	4	5
Multiplataforma	5	3	3
Accesibilidad al código fuente	5	0	2
Bajo costo	5	5	5
Experiencia del desarrollador	1	5	5
Documentación disponible	4	5	5
Entorno amigable	1	4	5
TOTAL	24	26	30

Fuente: Autor

Figura 41. Gráfica con la evaluación de software de Realidad Aumentada



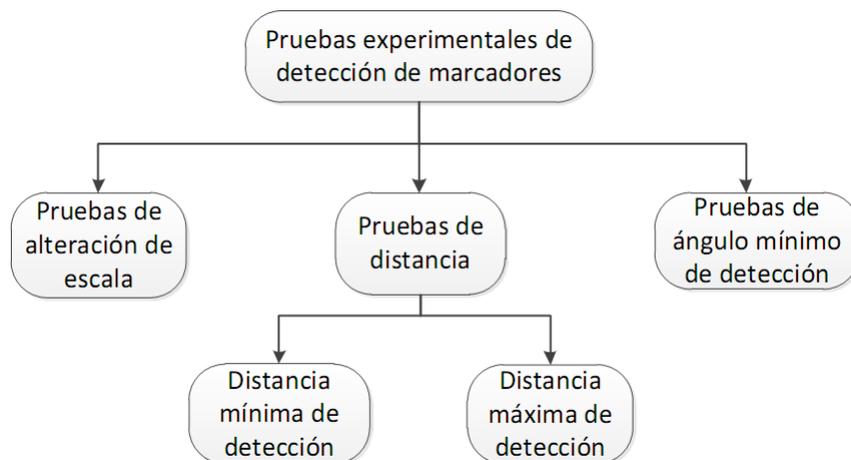
Fuente: Autor

Se puede observar que el software *BuildAR* obtuvo una clasificación alta en documentación, en torno amigable, estabilidad y bajo costo; sin embargo, el *ARToolkit* obtuvo mayores puntajes para multiplataforma y accesibilidad al código fuente. A pesar de todo lo anterior, el *BuidAR Pro* sobrepasa a los dos anteriores, pues obtiene la máxima calificación en 4 de 7 criterios (estabilidad, entorno amigable, documentación disponible, experiencia del desarrollador y bajo costo), ocupando mejor área de acuerdo con los elementos más relevantes este proyecto. Teniendo en cuenta lo anterior, se eligió el software *Build AR Pro* para este proyecto.

Refiriéndose a los marcadores (Figura 38), se les practicaron diferentes pruebas para establecer de forma cuantitativa unos criterios de selección.

Las pruebas se pueden dividir en tres secciones: la primera comprende las pruebas de alteración de la escala del marcador para observar su influencia en el tamaño del objeto insertado por la RA; la segunda agrupa las pruebas para determinar cuáles son los ángulos mínimos de visión de la cámara con respecto al marcador y así asegurar la detección del mismo por el software de RA; y la tercera, agrupa las pruebas para determinar las distancias mínimas y máximas de detección de los marcadores con respecto a la cámara.

Figura 42. Diagrama de pruebas experimentales de marcadores



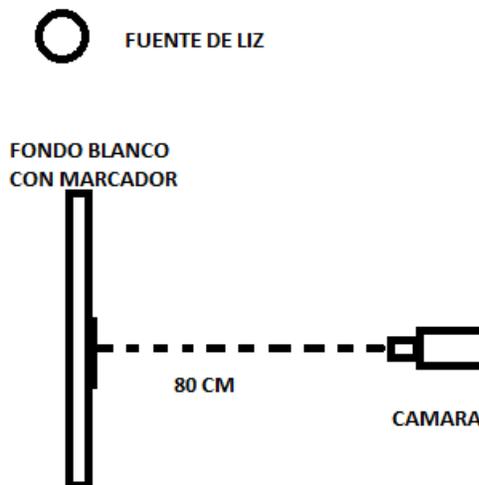
Fuente: Autor

Por otro lado, las pruebas basadas en análisis morfológicos estudiaron la influencia de la forma del marcador en su desempeño, por medio del análisis de descriptores de forma tales como: porcentaje de área convexa, porcentaje de área simple, número de elementos y solidez.

Pruebas de escala. Para esto se imprimió el marcador Hiro en tres tamaños: el primero a su escala original (9x9 cm); el segundo se imprimió al 150% (13.5x13.5 cm); y por último, se imprimió el marcador al 200% (18x18cm).

El ambiente en el que se realizaron las pruebas está constituido por un telón blanco de 115x79 cm, en el cual se adhirió en la parte inferior un marcador. La iluminación consta de un bombillo de luz blanca de 125 watts a 2.5 metros de altura en el centro de la habitación.

Figura 43. Disposición de elementos en prueba de escala



Fuente: Autor

Para esta prueba se dejó el robot estático y solo se alteró la escala del marcador.

Se pudo apreciar en las pruebas que las alteraciones de escala de los marcadores afectan de forma proporcional la escala de los objetos virtuales, por lo cual se puede compensar el tamaño, aumentándolo, para trabajar a distancias mayores a 6 m y disminuir el tamaño del marcador para trabajar a distancias menores a 17cm. La resolución de la cámara es un aspecto a tener en cuenta pues influye directamente en su capacidad de identificación del marcador, tornándose crítica cuando se manejan marcadores a largas distancias (más de 6m) o cuando se manejan marcadores de menor tamaño del estándar (8x8 cm), en este caso todas las pruebas en este trabajo se realizaron con una resolución de 640x480 a 25 FPS. Igualmente se debe considerar que el marcador agrandado debe estar dentro del volumen de visión de la cámara, pues estos sistemas son sensibles a pérdidas o alteraciones mínimas del marcador.

Análisis morfológico de marcadores

Con el objetivo de realizar un análisis la influencia de la forma del marcador en las distancias y ángulos de detección se realizó un análisis morfológico de los marcadores, empleando el software MATLAB.

Los descriptores empleados son los siguientes:

- Convex Hull (Envoltente convexa): teniendo un conjunto de puntos en el plano, su envoltura convexa (o casco convexo) está definida por el polígono convexo de área mínima que cubre todos los puntos (es decir, todos los puntos están dentro del polígono).
- Número de elementos: cantidad de elementos identificados dentro del marcador, definiendo elemento como el conjunto de pixeles conectados o agrupados dentro de la imagen con una cantidad superior a 100.

- Porcentaje de área convexa: porcentaje de la imagen interior del marcador que se encuentra dentro de la envolvente convexa en torno a un elemento identificado.
- Área simple: porcentaje de imagen interior del marcador que se encuentra dentro de un elemento identificado

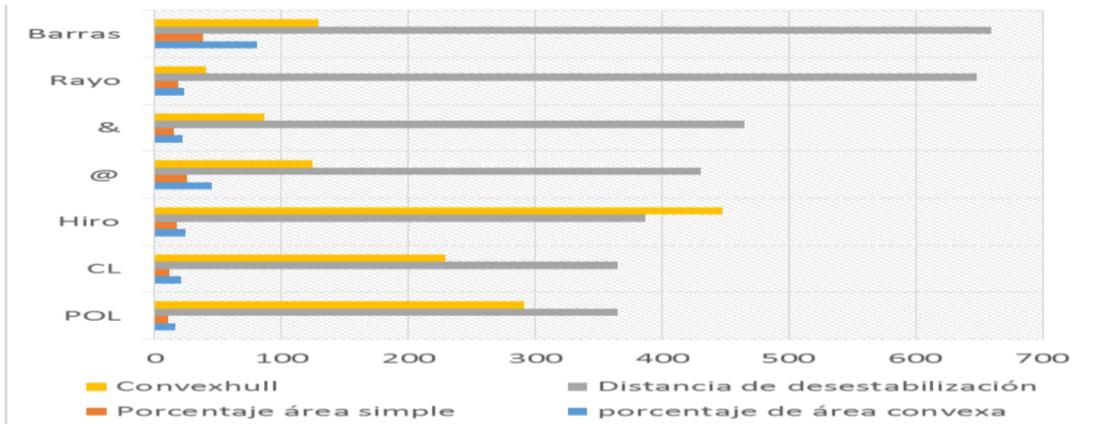
Los resultados de dicho análisis se pueden apreciar en la Tabla 17:

Tabla 17. Descriptores de forma, grados mínimos de detección y distancias máximas de detección discriminada por marcadores

Marcador	Porcentaje área convexa	Porcentaje área simple	Número de elementos	Convexhull	Solidez (0-10)	Grados de visualización	Distancia de desestabilización
POL	16,02	10,13	3	291	6,323	8	365
CL	20,66	11,67	2	229	5,706	10	365
Hiro	24,19	17,61	5	448	8,279	8	387
@	44,63	25,30	1	125	5,667	10	430
&	22,38	15,70	1	86	0,701	5	465
Rayo	22,65	18,86	1	40	8,327	4	648
Barras	80,17	37,69	1	129	4,702	4	660

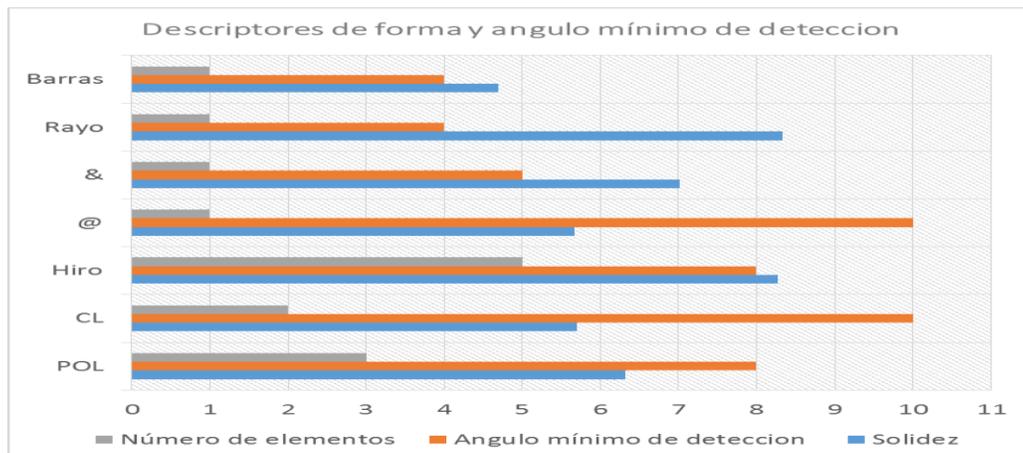
Fuente: Autor

Figura 44. Descriptores de forma y distancias máximas de detección



Fuente: Autor

Figura 45. Gráficas de relación de descriptores de forma y ángulos de detección



Fuente: Autor

Además de lo anterior, también se obtuvo la posición de los centroides de los elementos que componen la parte interna de los marcadores, con el fin de establecer su relación con las distancias de detección.

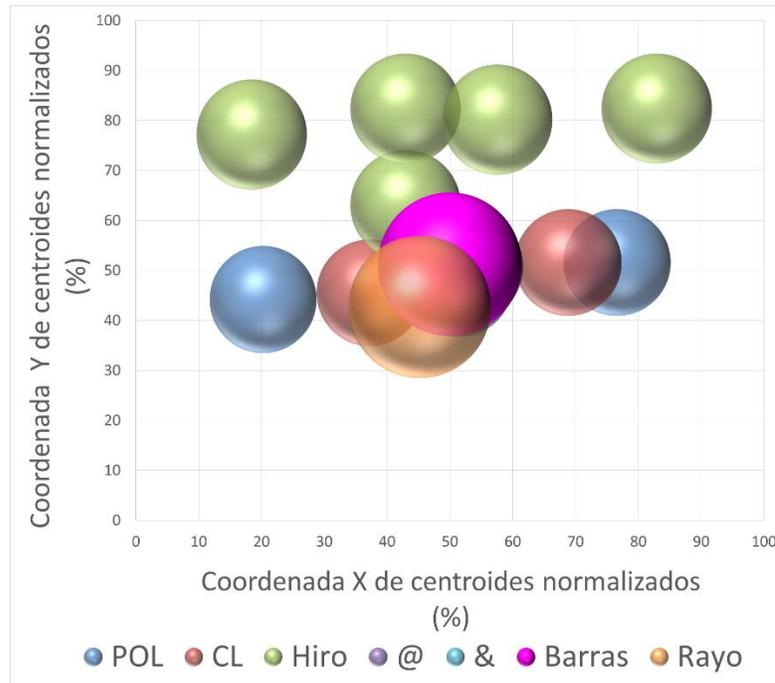
Tabla 18. Centroides de imágenes internas de los marcadores

	Coordenada X	Coordenada y	Distancia de desestabilización
POL	0,20237146	0,44168467	365
	0,51460955	0,47390582	
	0,7661015	0,51590643	
CL	0,37303496	0,45538455	365
	0,68849054	0,51566443	
Hiro	0,18456376	0,77189236	387
	0,4295302	0,63047619	
	0,4295302	0,82333333	
	0,57550886	0,80140984	
	0,82940706	0,82385933	
@	0,52399256	0,50740418	430
&	0,50413433	0,52833896	465
Rayo	0,45204873	0,42658099	648
Barras	0,50012133	0,5125514	660

Fuente: Autor

Para facilitar el análisis de estos resultados se pueden representar en una gráfica en donde se indican los centroides, el número de elementos conectados y las distancias de desestabilización. Estas últimas se representan en una esfera donde el radio es directamente proporcional a la distancia de desestabilización, es decir, a más distancia mayor radio de la esfera.

Figura 46. Gráficas de centroides de imágenes internas de marcadores y distancia de desestabilización representada en el radio de las esferas



Fuente: Autor

Analizando las gráficas anteriores, se puede observar que los marcadores que presentan menos cantidad de centroides, como el marcador Barras o Rayo, fueron identificados a distancias más largas. Adicionalmente, si se analiza este grupo particular de marcadores se encuentra que aquellos centroides más cercanos al centro de la imagen ($x=50$, $Y=50$), como el de Barras, obtuvieron las mejores distancias.

Se pudo observar también, que los marcadores que obtuvieron los ángulos más agudos de identificación, como Rayo y Barras, fueron los mismos que mostraron las distancias de identificación más largas. Con esto podemos inferir que el número de centroides de la imagen contenida en el recuadro del marcador influye no solo en la distancia de identificación, sino también en la magnitud del ángulo mínimo de identificación.

Además de lo ya expuesto, se puede encontrar que el marcador que mejor se comporta (Barras) presenta la menor solidez de todos, al igual que la mayor área convexa. Lo anterior da pie para generar unos criterios básicos de selección y para obtener el mejor rendimiento de los marcadores diseñados. Estos criterios son:

- La imagen contenida dentro del recuadro debe poseer un solo centroide
- La solidez de la imagen contenida dentro del recuadro debe ser menor a 0.5 (Tabla 17)
- El área convexa de la imagen contenida dentro del recuadro debe ser mayor a 35% (Tabla 17)
- El centroide de la imagen contenida dentro del recuadro debe acercarse lo más posible o coincidir con el centroide del marcador

Los criterios anteriores no deben considerarse como una camisa de fuerza en la selección de marcadores para Realidad Aumentada; solo son las recomendaciones que surgen de los análisis experimentales y el análisis de los descriptores de forma.

Para corroborar la relación de la solidez con la distancia de detección se tomaron tres tamaños de la marca Hiro; el primero a una escala estándar (8x8 cm); el segundo con una escala de 1.5 (12x12 cm); y el último con una escala de 2 (16x16 cm), y se fijaron sus máximas distancias de detección.

Tabla 19. Distancias máximas de detección para prueba de solidez

	Escala 1	Escala 1.5	Escala 2
Distancias máximas de detección	6.67	8.41	8.89

Fuente: Autor

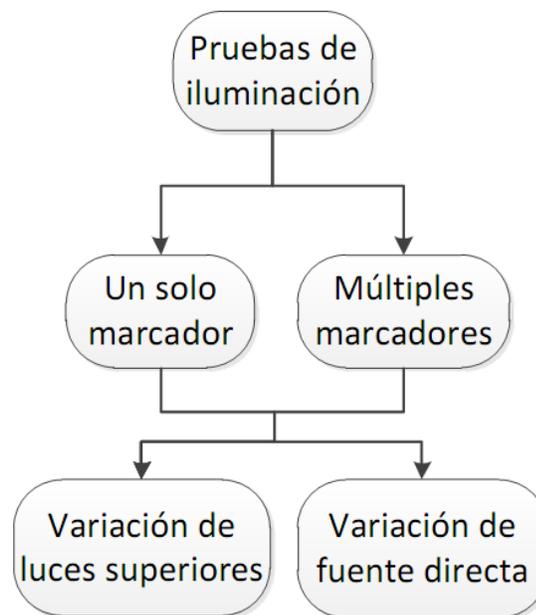
La tabla anterior revela que la distancia máxima de detección no está asociada solo a los criterios mencionados, sino que también está influenciada por el tamaño del marcador, pues aunque se

conservó la solidez en el marcador seleccionado, al alterar la escala, la distancia no se relaciona linealmente con este parámetro, lo cual implica que si bien la solidez es un parámetro importante en la selección de los marcadores debe considerarse en conjunto con la escala.

2.4. Parámetros de iluminación

Uno de los elementos críticos en los sistemas de visión artificial, al igual que en la Realidad Aumentada es la iluminación. En éste caso se realizaron diferentes pruebas para determinar: cómo ésta influye en la segmentación y la umbralización de los marcadores en la RA y en determinar cuáles serían las condiciones ideales para un funcionamiento óptimo del sistema.

Figura 47. Estructura de pruebas de iluminación

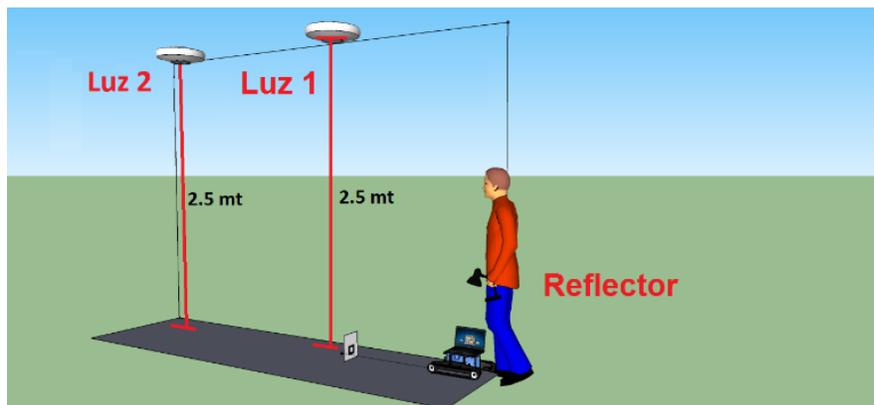


Fuente: Autor

En el ambiente de prueba intervienen: un marcador denominado Hiro, el robot con su cámara, las dos fuentes de luz de 125 watts ubicadas a 2.5 m del suelo, y una fuente de luz directa representada en un reflector con un bombillo de luz blanca de 60 watts.

Para la prueba inicial se encenderán alternadamente las fuentes de luz para observar su efecto sobre el sistema de RA.

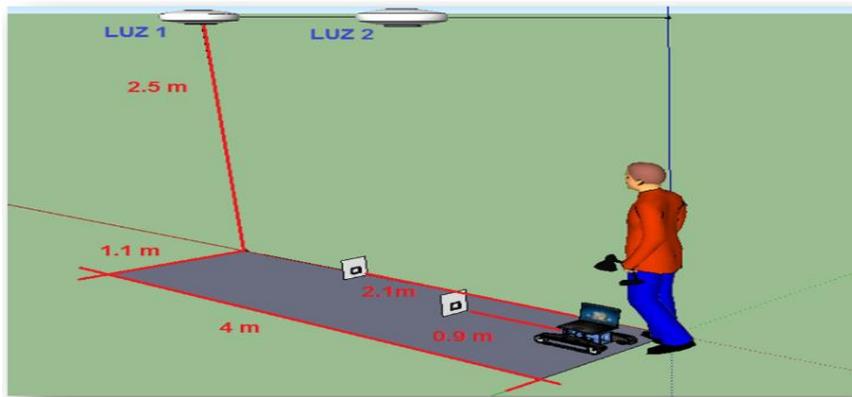
Figura 48. Ambiente de prueba con un solo marcador



Fuente: Autor

Posteriormente se sumó otro marcador y se repitió el ejercicio para verificar el efecto sobre un conjunto de marcadores en el mismo escenario.

Figura 49. Ambiente de prueba con dos marcadores

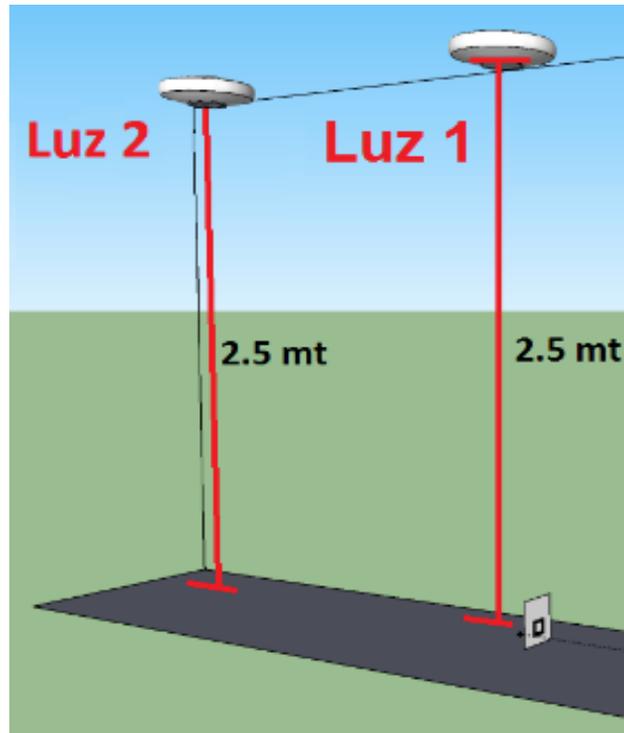


Fuente: Autor

Estas pruebas nos ayudan a establecer la influencia de la posición de las fuentes de luz y el tipo de iluminación (difusa o directa) sobre la detección de los marcadores, ya sean individuales o en conjunto.

Los resultados de estas pruebas se pueden dividir en función de la cantidad de marcadores utilizados, es decir, en pruebas con un marcador y pruebas con múltiples marcadores. Inicialmente se tratarán los resultados obtenidos con un solo marcador (Hiro con un tamaño de 8x8 cm), a una distancia de 90 cm del robot y con variación solo de las fuentes de luz superiores (Luz 1, luz 2).

Figura 50. Disposición de Luz 1 y Luz 2 con respecto al marcador



Fuente: Autor

Aquí se puede ver el efecto de la iluminación sobre el proceso de umbralización y segmentación del software de RA.

Tabla 20. Comportamientos de marcador frente a variaciones de iluminación en fuentes superiores

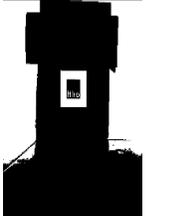
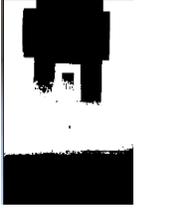
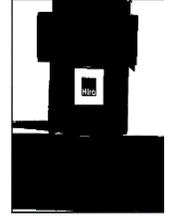
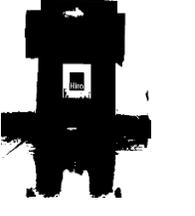
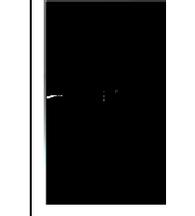
	Solo luz diurna	Solo luz 1	Solo luz 2	luz 1 y 2
Imagen con RA				
Imagen binaria				

Fuente: Autor

Se puede apreciar en las imágenes binarizadas que los mejores comportamientos se obtuvieron con solo la luz 1 presente y con las dos fuentes de luz energizadas (luz 1 y luz 2). Esto es resultado de la carencia de sombras y los niveles de contraste uniformes en la superficie en donde se alojó el marcador.

Este resultado se complementó con una prueba en la cual se adicionó una fuente de luz directa, consistente en un reflector con una bombilla de luz blanca de 60 vatios y conservando las condiciones de la prueba anterior.

Tabla 21. Comportamientos de marcador frente a variaciones de iluminación en fuentes superiores y reflector

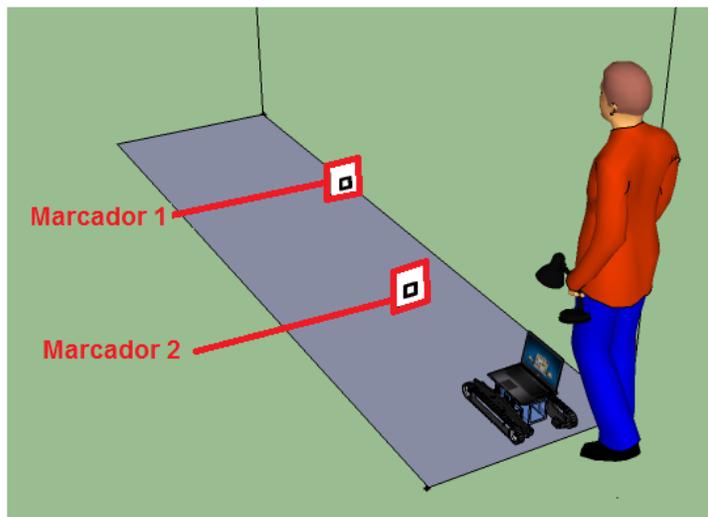
	Luz 1 y 2 con reflector desde el robot	Luz 1 y 2 con reflector desde el robot con sombra	Luz 1 y 2 y reflector encima del marcador	Luz 1 y 2 y reflector detrás del marcador	Luz 1 y 2 y reflector frente a la cámara
Imagen con RA					
Imagen binaria					

Fuente: Autor

Igual que en el caso anterior, los mejores resultados se obtuvieron con las luces 1 y 2 presentes, pues se puede apreciar el recuadro del marcador en la imagen binarizadas sin deformaciones. Sin embargo, en esta prueba la fuente de luz directa (reflector) en uno de los escenarios se encontraba ubicada desde el robot y en el otro sobre el marcador. Un aspecto negativo que se observa es la generación de sombras en las proximidades del marcador debido a la fuente de luz directa, lo que ocasiona problemas de detección reflejándose en falsos positivos.

En el caso de las pruebas con múltiples marcadores, estas se realizaron en un ambiente con las mismas características descritas en la Figura 49 e intercalando la posición de los marcadores entre sí, como se ve a continuación.

Figura 51. Disposición de los marcadores en el ambiente de pruebas de múltiples marcadores



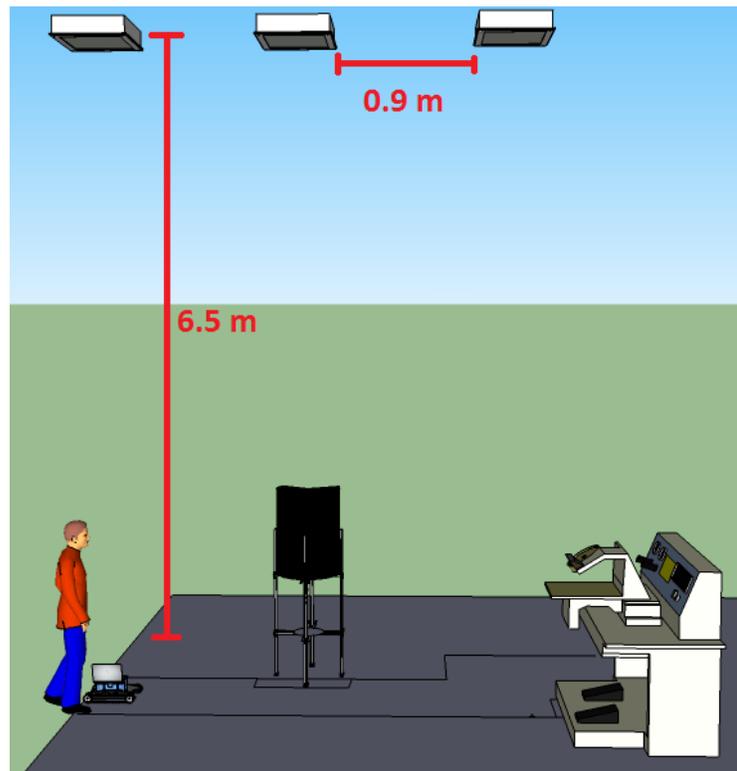
Fuente: Autor

Todo lo anterior ayuda a fijar un marco de referencia para las condiciones de iluminación que debe predominar en el ambiente de trabajo; dicho marco debe estar regido por las siguientes sugerencias:

- La iluminación debe propender por la ausencia de sombras en el marcador, de igual forma, para una mejor interpretación de los marcadores se recomienda que sea uniforme y tenue.
- No se recomienda utilizar fuentes de iluminación directa pues afectan la percepción del ambiente por parte del robot, esto se da al generar puntos de brillo repercutiendo en problemas de identificación de marcadores manifestado en falsos positivos.
- En el caso de tener que emplearse fuentes de luz directas, se recomienda situarlas en la parte superior de los marcadores o en la misma posición de la cámara, apuntando hacia el frente e iluminando su volumen de visión.

En el caso de la prueba industrial, la iluminación constará de 3 lámparas ubicadas encima de la ruta del robot, cada una de estas posee 4 tubos fluorescentes de 20 vatios cada uno. Cada lámpara se encuentra a 6.5 m del suelo y con una separación de 0.90 m entre ellas.

Figura 52. Ubicación de las fuentes de iluminación en el ambiente de prueba industrial



Fuente: Autor

2.5. Ambiente de pruebas

En la industria, entre los ambientes más comunes están los corredores o pasillos, ya sean delimitados por un elemento refractivo en el piso o por barreras físicas, como paredes o mallas de seguridad. Estos se encuentran no solo en plantas de producción y zonas de trabajo sino también en áreas administrativas y de bodegaje, por lo cual se eligió como ambiente de prueba principal un

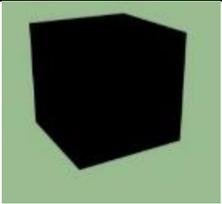
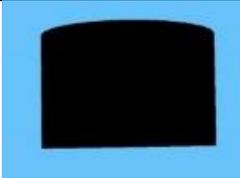
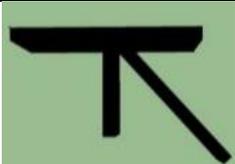
corredor de 420x110 cm, en el cual se ubican diferentes obstáculos, y por medio de su sistema de visión artificial el robot debe llegar de un extremo al otro, en lo posible, sin tocar los obstáculos.

En un principio la prueba se realiza con objetos físicos; posteriormente se cambian estos objetos con marcadores para que el software de RA inserte los elementos virtuales en la imagen.

El escenario se diseñó con el fin de evaluar el comportamiento de los sistemas en ambientes estructurados con obstáculos que posean líneas curvas, ángulos rectos y objetos irregulares con aristas (Poliedro).

Los elementos seleccionados y sus marcadores asociados respectivamente son los siguientes:

Tabla 22. Elementos utilizados en las pruebas de navegación

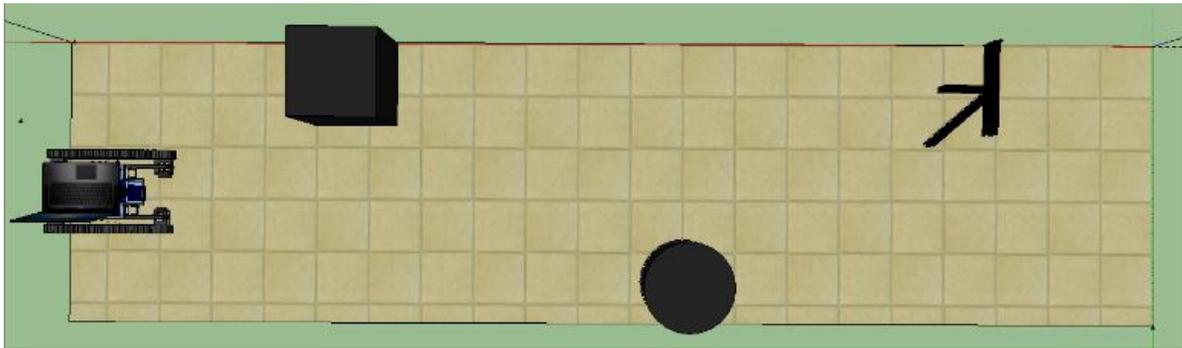
ELEMENTO	FIGURA	DESCRIPCIÓN	MARCADOR
Cubo		Cubo de 30x30x30 cm	
Cilindro		Cilindro de 15 cm de Radio y 16 cm de altura	
Poliedro		Poliedro está basado en la forma de una letra T la cual en su parte superior posee un rectángulo de 35x16x1 cm y en su parte inferior presenta una columna de forma rectangular de	

		15x16x1 cm. Adicionalmente se le insertó otro rectángulo en un ángulo de 45° con respecto a la columna con un tamaño de 22x10x1 cm .	
--	--	--	--

Fuente: Autor

Estos objetos se posicionaron como se ve en la Figura 53.

Figura 53. Vista superior del ambiente de prueba con objetos reales



Fuente: Autor

Adicionalmente, se implementó una prueba en un ambiente industrial, ubicado en la Institución Universitaria Pascual Bravo, consistente en una planta de empaquetado de alimentos conformada por una máquina de empaquetado, una llenadora de líquidos y una cortadora. Todo esto se empleará para verificar el comportamiento del robot en este tipo de escenarios.

Figura 54. Laboratorio de Mecatrónica, Institución Universitaria Pascual Bravo



Fuente: Autor

Se decidió para la prueba tomar un corredor aledaño a las máquinas de llenado, de forma que el robot inicie en un extremo y llegue al otro sin sufrir colisiones o salirse de la senda delimitada por una cinta de color negro. En este caso el robot interactuará con dos obstáculos, el primero es una máquina llenadora de granos y el segundo es una máquina de corte (Figura 55).

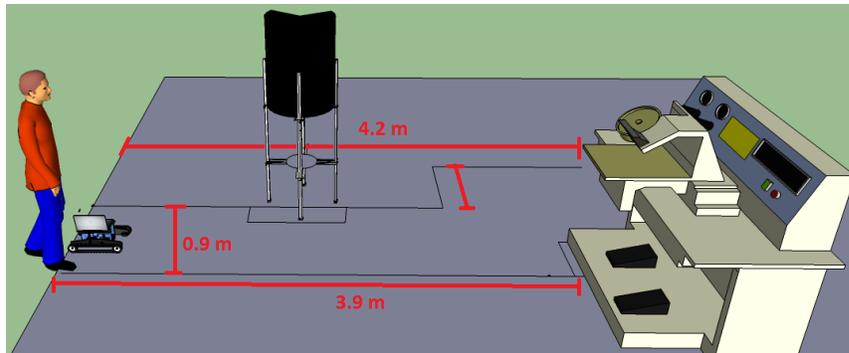
Figura 55. Máquina llenadora de granos y cortadora industrial



Fuente: Autor

El corredor de la prueba consta de 4.2 m de largo y 0.90 m de ancho, con una desviación en la parte final.

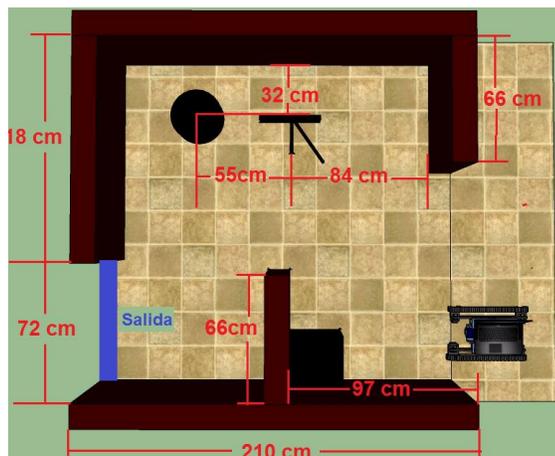
Figura 56. Dimensiones de ambiente de prueba industrial



Fuente: Autor

Por último, se decidió plantear un ambiente distinto a un corredor, en el cual se evaluará el comportamiento del robot en un espacio más amplio. Para tal efecto se utilizaron los obstáculos empleados en las pruebas de corredores (cubo, cilindro y poliedro) dispuestos en un espacio rectangular de 210x190 cm el cual se aprecia en detalle en la Figura 57.

Figura 57. Ambiente de prueba para prueba sin corredores



Fuente: Autor

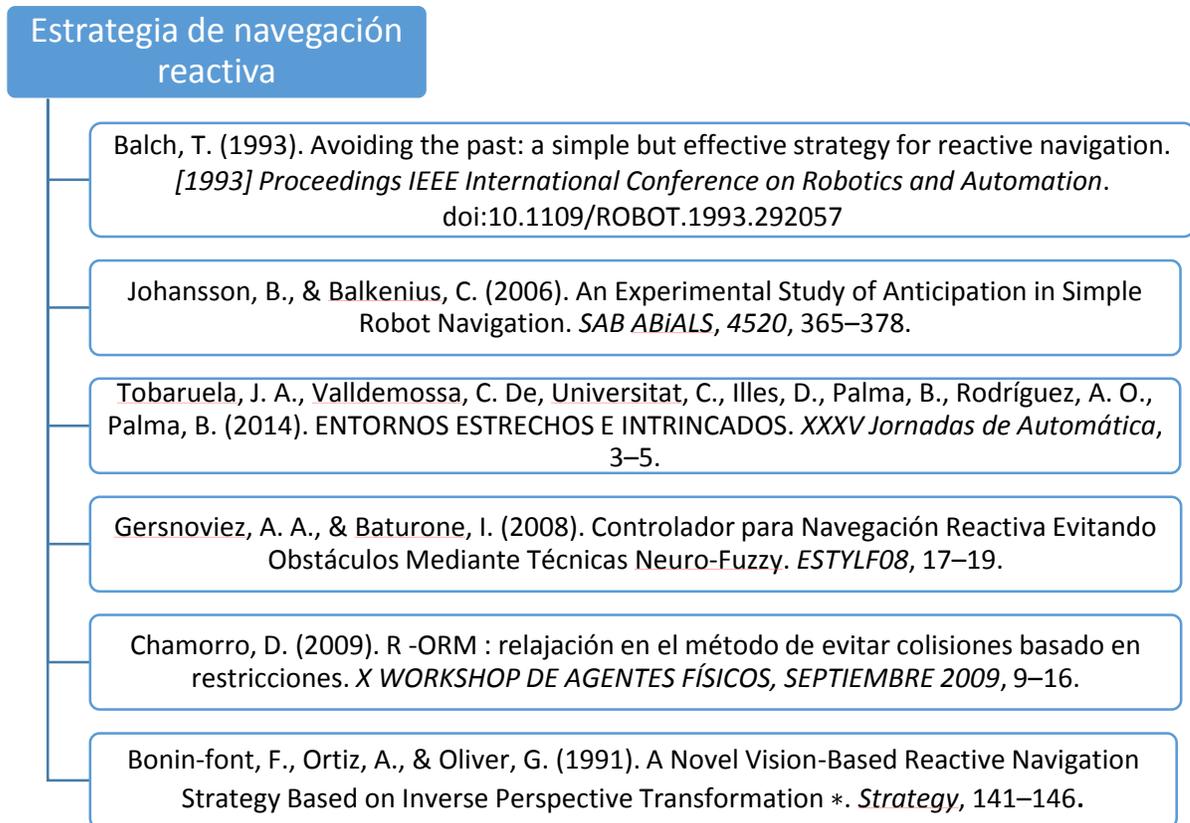
El objetivo, en este caso, es que el robot navegue por el ambiente desde el punto de inicio hasta la salida del laberinto, esquivando los obstáculos, sin colisionar con las paredes que delimitan el ambiente.

2.6. Algoritmos de navegación

El primer algoritmo de navegación implementado para las pruebas se conoce con el nombre de algoritmo reactivo. La parte comportamental en este algoritmo está determinada por la información suministrada por los sensores (en este caso la cámara). Las ventajas de este algoritmo son su fácil implementación, un bajo costo computacional y rapidez en sus decisiones (Arias, 2009). Además, la utilización de este algoritmo como una estrategia de navegación está respaldada ampliamente en la literatura.

A continuación se muestra un conjunto de escritos académicos y científicos que soportan esta afirmación.

Figura 58. Artículos que emplean el algoritmo reactivo como estrategia de navegación

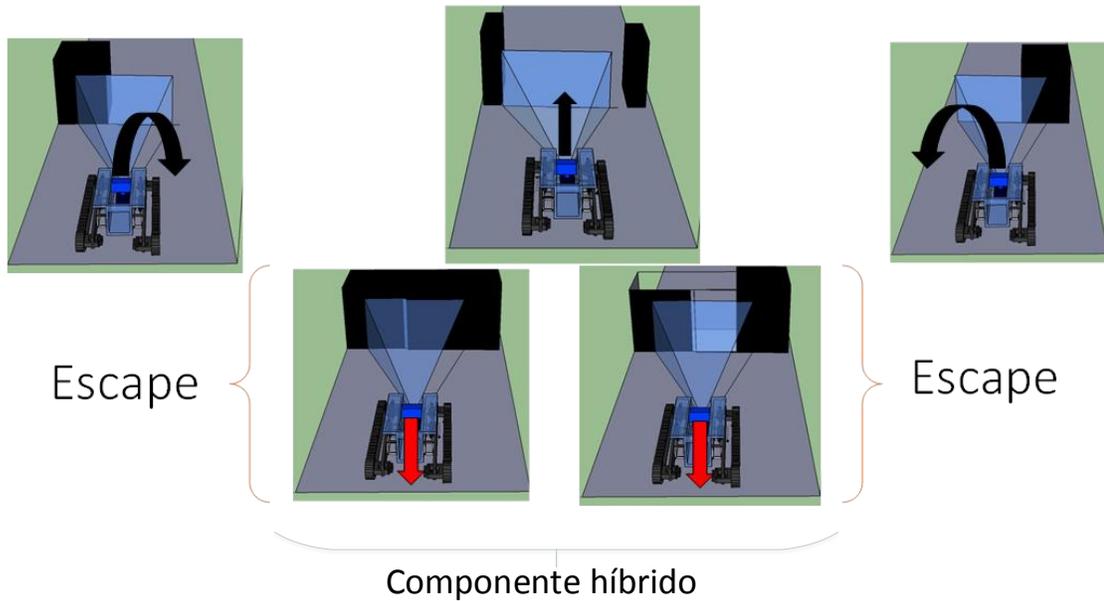


Fuente: Autor

Se debe aclarar que este algoritmo de navegación no es reactivo puro, pues además de los comportamientos clásicos de la reactividad, se le incluyó un comportamiento de escape perpendicular al plano de visión basado en mapeo del espacio libre, convirtiéndolo en un algoritmo híbrido.

A continuación se muestran los casos de navegación contemplados en este algoritmo (Figura 59).

Figuran 59. Casos de navegación

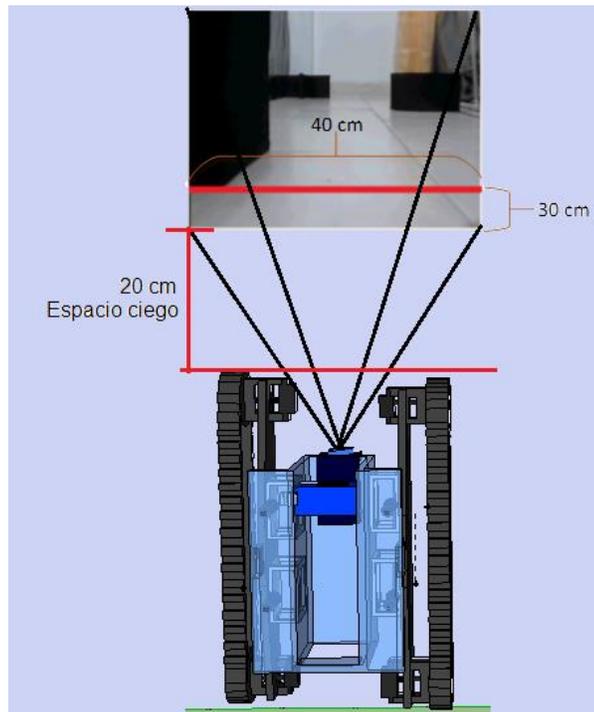


Fuente: Autor

Hablando de la implementación del algoritmo, como tal, este emplea la cámara frontal del robot para obtener una imagen con la que se plantea la navegación. A dicha imagen se le extrae una fila de píxeles (resaltada con rojo en la Figura 60), que representa una distancia de 50 cm, medidos a partir de la parte frontal de robot hasta el punto de detección de los obstáculos (20 cm de punto ciego y 30 cm del plano visible captado por la cámara).

A esta distancia el marco de visión posee 40 cm en su componente horizontal, esta distancia es suficiente para que el robot navegue por el ambiente teniendo en cuenta que el robot tiene un ancho de 30 cm, dejando un margen de seguridad de 5 cm a cada lado (Figura 60).

Figura 60. Distancia de detección de obstáculos y su representación en la imagen tomada por la cámara frontal del robot



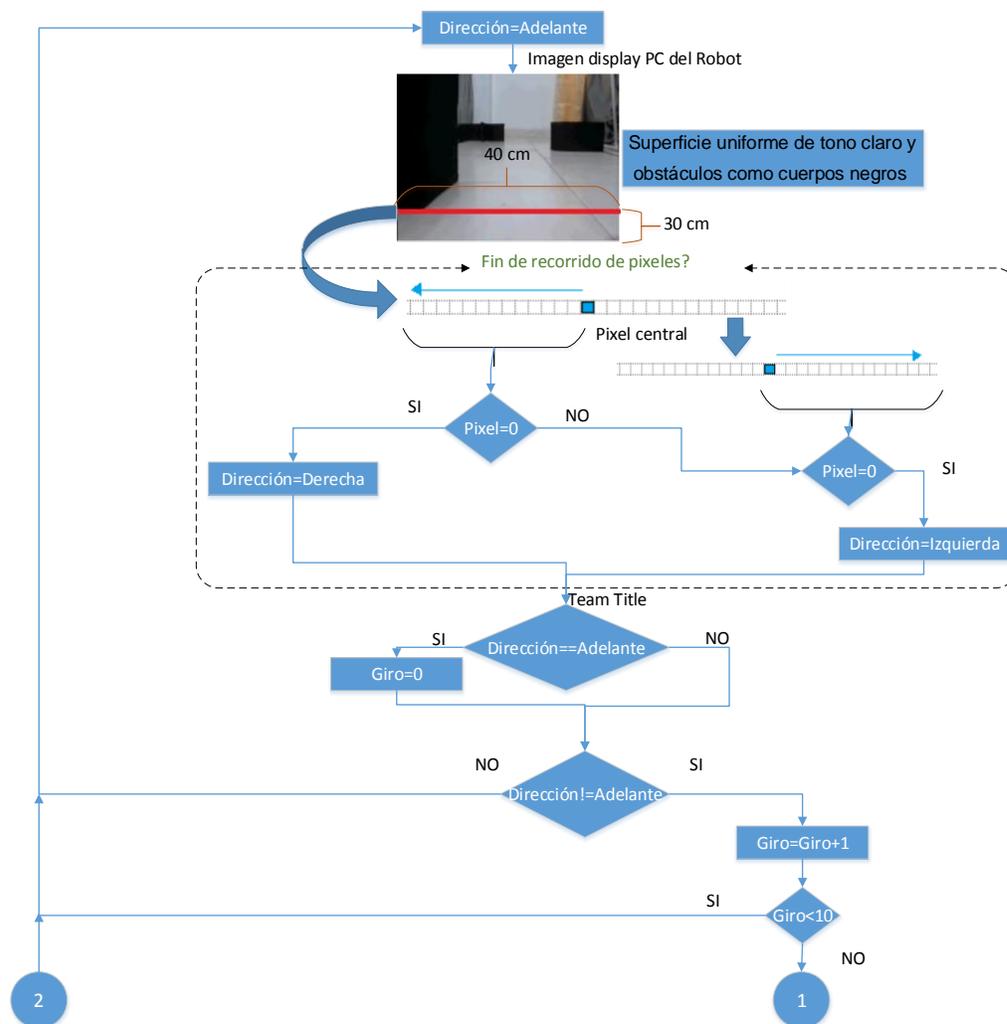
Fuente: Autor

Con el fin de simplificar la detección de los obstáculos, los ambientes de prueba deben tener un piso de textura uniforme y color claro. Esto para garantizar el máximo contraste entre esta superficie y los obstáculos, los cuales son detectados como cuerpos negros en la imagen.

Para la detección de los obstáculos, como cuerpos negros, el algoritmo revisa la franja horizontal de píxeles extraída de la imagen, iniciando del centro hacia afuera en busca de píxeles negros. Si se detecta un obstáculo, el robot reaccionará evadiéndolo, girando hacia un espacio vacío. Si el robot no puede avanzar por no detectar un espacio libre, este activa el comportamiento de escape, muestrea de nuevo la imagen con la línea de píxeles extraída e identifica la sección con mayor cantidad de píxeles blancos para dirigirse en esa dirección.

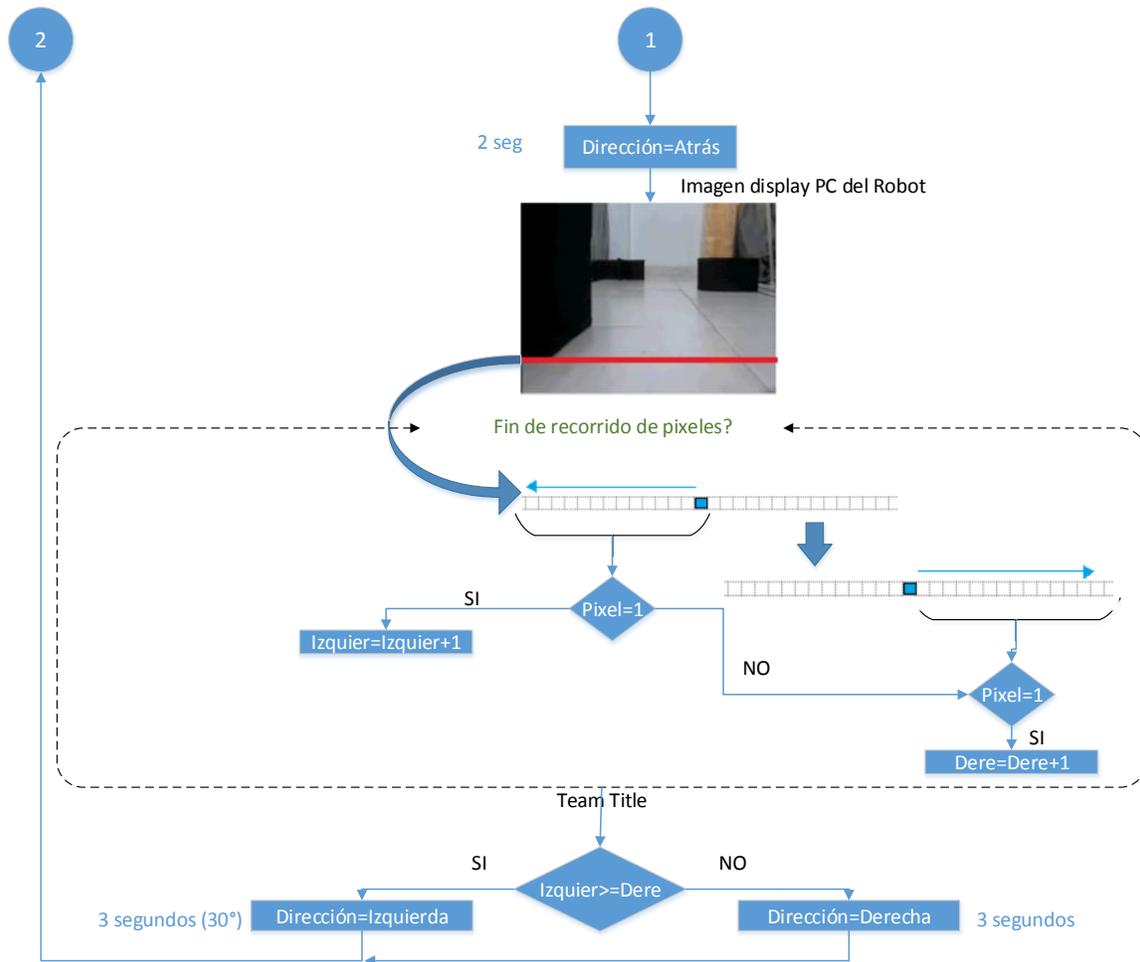
El algoritmo se muestra en detalle en las siguientes figuras:

Figura 61. Sección reactiva del algoritmo de navegación reactivo híbrido



Fuente: Autor

Figura 62. Sección híbrida del algoritmo de navegación reactivo híbrido



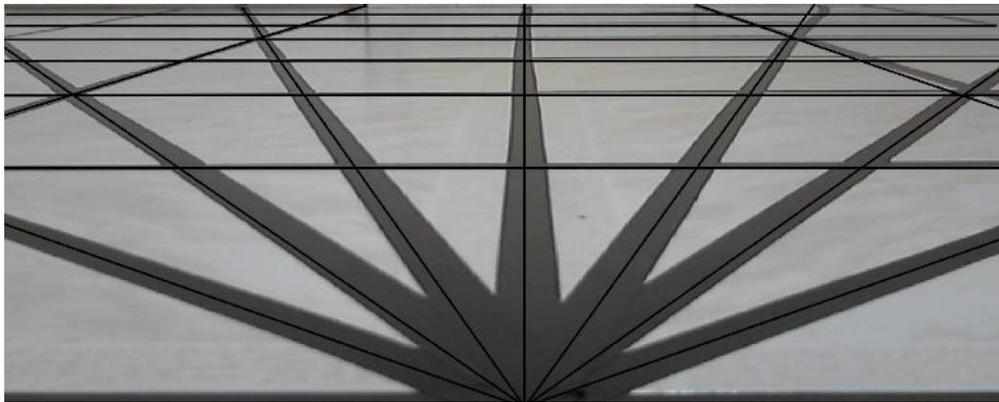
Fuente: Autor

El segundo algoritmo de navegación seleccionado se denomina algoritmo de campos potenciales artificiales. La magnitud del vector resultante de la suma de las fuerzas de repulsión y atracción sobre el robot, define la velocidad de recorrido y el ángulo con respecto al mismo. Esto se traduce en que cuando se obtiene un ángulo positivo el robot gira a la derecha y cuando se obtiene un ángulo negativo el robot gira a la izquierda. Cabe aclarar, que como el robot se mueve a una velocidad constante (15cm/s), en este caso la variable de interés para el desplazamiento es el ángulo de orientación del vehículo en función de su trayectoria hacia la meta. Como en este trabajo el

robot no tiene conocimiento del ambiente de pruebas, la fuerza de atracción de la meta se reemplazó por un vector de atracción que apunta siempre al frente del robot con una magnitud fija para garantizar su desplazamiento.

Para facilitar el proceso de estimación del ángulo de incidencia de los vectores originados desde los obstáculos, se optó por implementar una plantilla para realizar el rastreo de los objetos a través de la ruta. Esta plantilla se obtuvo de forma experimental trazando líneas negras en el piso en intervalos de 22.5 grados, medidos desde el punto focal de la cámara frontal del robot.

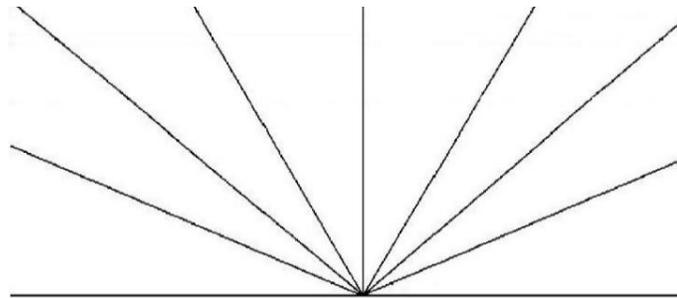
Figura 63. Plantilla experimental para algoritmo de campos potenciales



Fuente: Autor

A la imagen de esta plantilla se le realizó un tratamiento digital para dejar solo el trazado de los ángulos, lo que generó un patrón sobre el cual se hace el rastreo de los objetos y se generan los vectores repulsivos correspondientes.

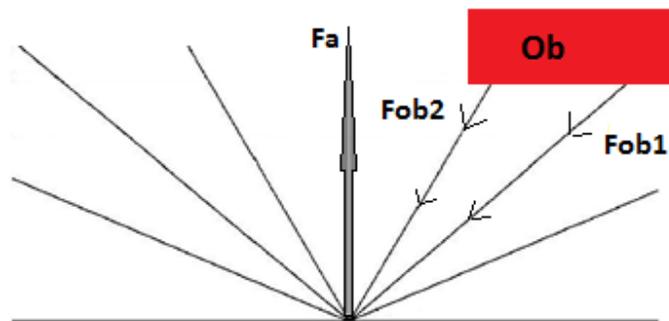
Figura 64. Plantilla con tratamiento digital de imágenes para algoritmo de campos potenciales



Fuente: Autor

A modo de ejemplo, se supone la existencia de un obstáculo “Ob” que se encuentra dentro del rango de visión de la cámara; en el caso de tener el algoritmo sin la plantilla, este generaría un gran número de fuerzas sobre el robot, tantas como el tamaño en píxeles del objeto, pero con la plantilla el sistema solo tomará las fuerzas generadas por los puntos en los que el obstáculo coincide con las líneas de la plantilla, reduciendo significativamente las fuerzas generadas y por consiguiente el costo computacional para definir la trayectoria.

Figura 65. Ejemplo del efecto de un obstáculo sobre el algoritmo de campos potenciales implementando la plantilla obtenida experimentalmente



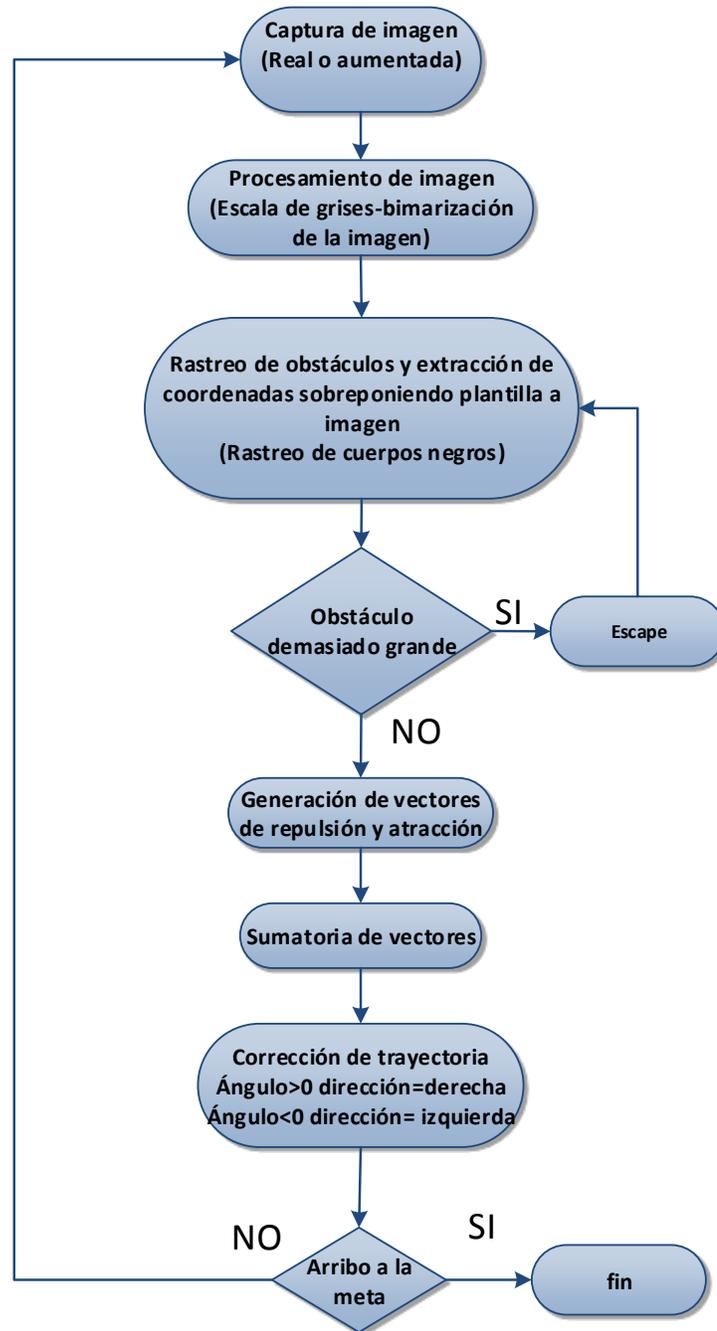
Fuente: Autor

Uno de los contratiempos que se puede presentar es que el obstáculo sea demasiado grande y sature el sistema; en este caso el algoritmo tiene incorporado una instrucción de escape, la cual obliga al robot a retroceder y retomar el escaneo de espacios libres para replantear la ruta. Desde otro punto de vista, si el objeto es muy pequeño o delgado podría escapar a las líneas de detección de la plantilla, pero al acercarse más al robot las líneas de detección se agrupan de forma más estrecha hasta que eventualmente será detectado por el sistema, aunque podría presentarse problemas al detectarse demasiado cerca. Una posible solución sería aumentar la cantidad de líneas de detección en la plantilla de modo que la resolución del sistema se incremente; también se podría complementar la plantilla con líneas horizontales de modo que se muestree también en este sentido.

Esta técnica no solo simplifica el proceso de estimación del ángulo de incidencia de los vectores sino que también ahorra recursos de procesamiento al rastrear solo el segmento de la imagen delimitada con los trazos de la plantilla en vez de analizar la imagen completa. Al realizar pruebas de laboratorio implementando los dos algoritmos, uno sin la plantilla y otro con ella, en el primero se tuvo un tiempo de procesamiento de 20 segundos y en el segundo solo se obtuvo un tiempo de procesamiento de 1 segundo. Esto muestra que el segundo algoritmo es más liviano y presenta mayor velocidad de respuesta;

A continuación se muestra un diagrama general del algoritmo de navegación por campos potenciales artificiales aplicado a la plataforma robótica.

Figura 66. Algoritmo general de campos potenciales aplicado a la plataforma robótica



Fuente: Autor

Hay que aclarar que en este trabajo se plantea un algoritmo de campos potenciales sin meta, por lo cual el robot nunca se detendrá debido a que la fuerza de atracción de la misma es constante y siempre apunta al frente del robot.

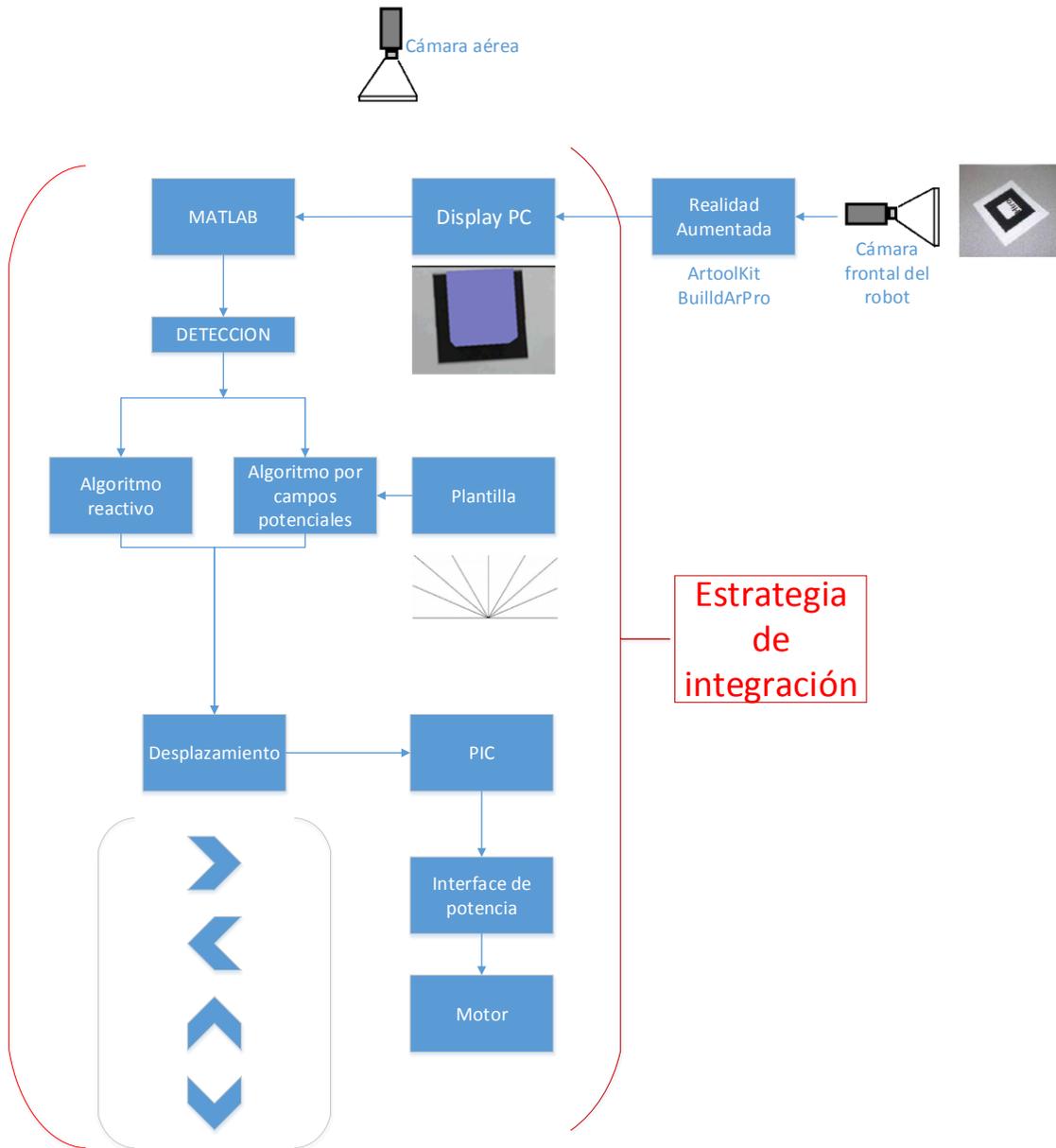
Se realizaron 64 pruebas en corredores por cada algoritmo de navegación; 32 con objetos reales y 32 con objetos virtuales insertados por la RA. Adicionalmente se realizaron 3 pruebas más, también en corredores, mezclando objetos reales y virtuales. Asimismo se hicieron 20 pruebas en un ambiente industrial y otras 20 en un laberinto para robustecer el estudio del comportamiento del robot

2.7. Estrategia de integración de la Realidad Aumentada con marcadores, el sistema de visión artificial y el algoritmo de navegación

La integración de todas las tecnologías que conforman el sistema planteado en este trabajo es fundamental para su buen funcionamiento, por lo cual se propone una estrategia para integrar la Realidad Aumentada con marcadores, el sistema de visión artificial y el algoritmo de navegación. Esta estrategia tiene como elemento fundamental el display del PC que se encuentra en la parte superior del robot. Este display sirve de canal de comunicación entre la Realidad Aumentada, el sistema de visión y los algoritmos de navegación, pues es allí donde se aprecian los elementos aumentados en la imagen generada por el software BuildAR Pro, la cual es tomada por el software de tratamiento digital de imágenes que a su vez transmite dicha imagen tratada a los algoritmos de navegación que identifican los obstáculos y generan la movilidad.

A continuación se muestra el diagrama general del sistema propuesto en este trabajo y en el cual se identifica la estrategia de integración planteado.

Figura 67. Estrategia de integración de la Realidad Aumentada con marcadores, el sistema de visión artificial y los algoritmos de navegación



Fuente: Autor

2.8. Evaluación de trayectorias

Para la evaluación de las trayectorias se utilizaron criterios estándar consignados en la literatura (Muñoz, Ortiz & Londoño, 2007).

1. Longitud de la trayectoria recorrida (PL): es la distancia total recorrida por el vehículo desde el punto de inicio hasta el final del recorrido.
2. Periodos de control (LeM): esta medida es relativa al número de decisiones tomadas por el planificador para alcanzar el objetivo, si el robot se mueve a velocidad lineal (v) constante, da una idea del tiempo empleado en completar la misión (Álvarez, 1998).
3. Distancia mínima a obstáculo: distancia mínima medida desde el centro del robot al obstáculo por el cual se pasó más cerca en toda la trayectoria.

Estos parámetros se obtuvieron con base en las imágenes de la cámara aérea, las cuales fueron analizadas con un algoritmo de visión artificial. Dichos parámetros servirán de base para describir de forma cuantitativa las trayectorias del robot empleando los dos algoritmos de navegación explicados anteriormente.

3. RESULTADOS Y ANÁLISIS

En este espacio se analizarán los resultados obtenidos con las pruebas de navegación descritas en el punto anterior; para ello se utilizaron los datos obtenidos y se relacionaron con la literatura existente, con el fin de establecer los vacíos observados a través del proceso investigativo.

3.1. Pruebas de navegación

En este caso, se buscó representar las diferentes trayectorias trazadas por el robot por medio de dos líneas paralelas, las cuales representan el trazado de cada oruga a través del ambiente de pruebas, azul en el caso de las trayectorias obtenidas con objetos reales y rojo con los objetos insertados por la Realidad Aumentada.

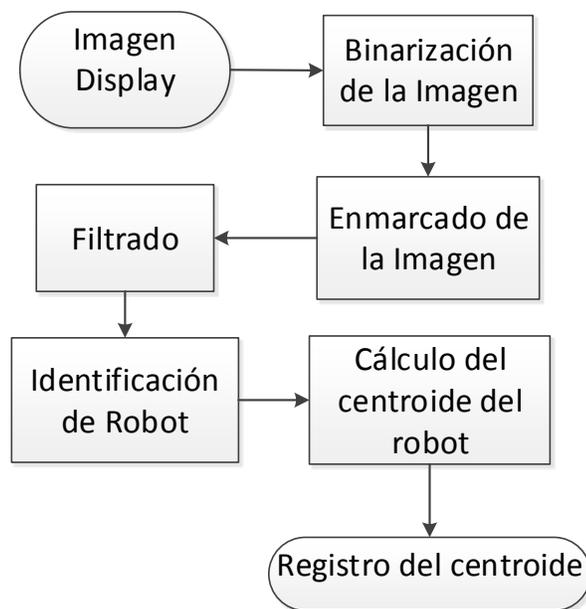
Se debe aclarar que todas las gráficas que se ven en este subcapítulo corresponden a la perspectiva de la cámara aérea que rastrea el robot a través del ambiente de pruebas y no de la cámara frontal del robot; además, las discontinuidades que se perciben en las gráficas de las trayectorias se presentan por el proceso de descomposición de los videos en fotogramas. Este proceso de descomposición se realizó con un periodo de muestreo de 1.5 segundos, basado en el desplazamiento percibido en los videos a través de los ambientes de prueba, de forma que las gráficas de las trayectorias representaran el comportamiento del robot a través del escenario, sin saturar el algoritmo de tratamiento digital de imágenes empleado para rastrear el robot. El proceso general de trazado de las trayectorias comienza con la descomposición del video en fotogramas; a cada fotograma se le recortan los bordes o se enmarca para eliminar objetos ajenos al ambiente de pruebas que pueden estar en los límites de la imagen, como los recuadros del reproductor de *Windows Media* donde se reprodujo el video o el botón de inicio de Windows; luego, se pasa la imagen enmarcada a escala de grises y posteriormente se binariza con un umbral de 0.34, cuidando

que desaparezcan de la imagen elementos indeseados como sombras y cuadrículas generadas por las baldosas del piso, tratando de dejar solo los obstáculos y el robot.

Posteriormente, se realizó un filtrado de los elementos a los que no fue posible eliminar con la umbralización. Dicho filtro se diseñó en base al tamaño de los cuerpos negros presentes en la imagen, eliminándolos si eran demasiado grandes (mayores a 10950 píxeles) o pequeños (menores a 7000 píxeles), para que fueran los obstáculos o el robot.

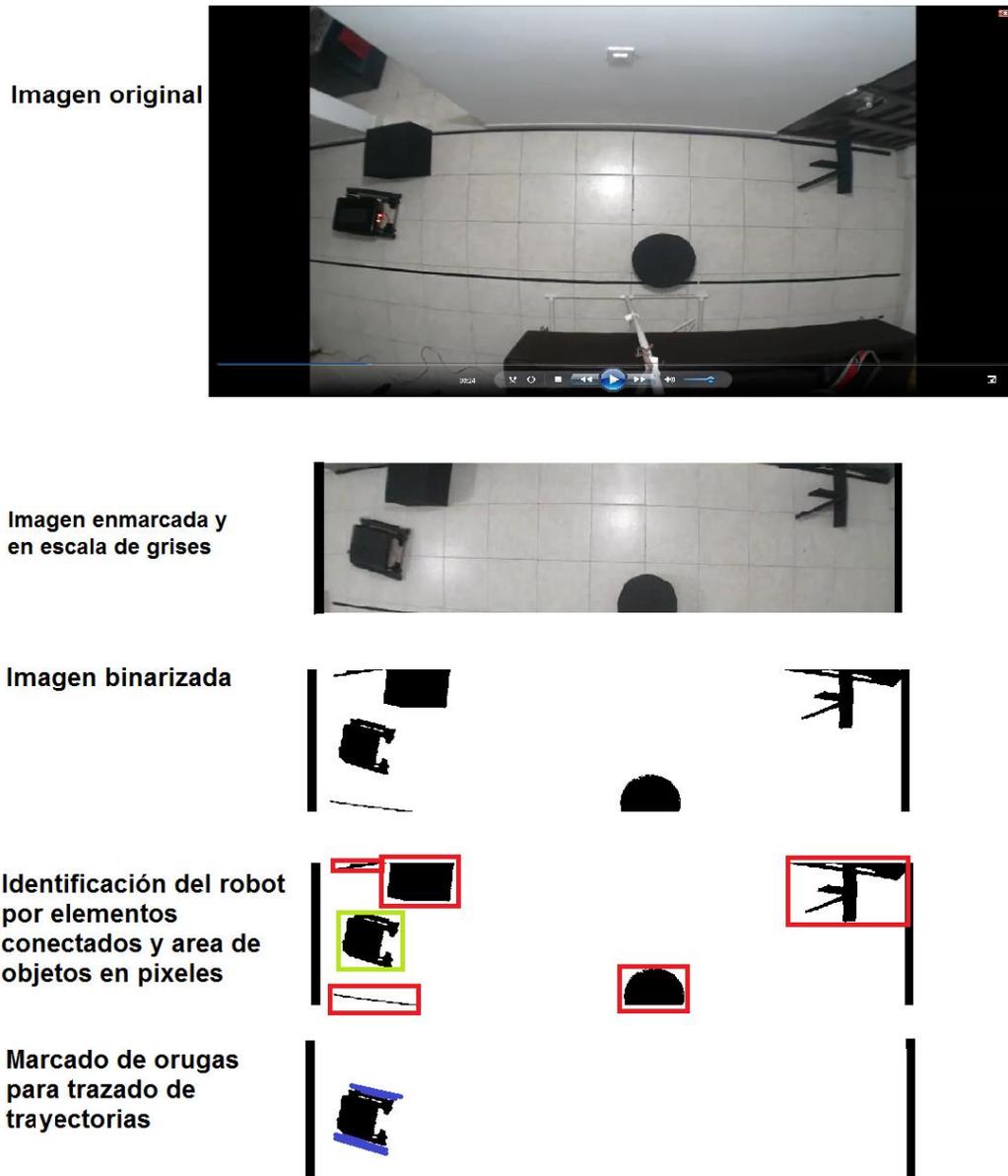
Por último, se identificó el robot basándose en la forma y el tamaño del mismo. El trazado de las trayectorias se consigue muestreando el centro del robot y demarcando los lados del mismo para efectos de registro del trazado de las orugas

Figura 68. Diagrama de flujo de extracción de trayectorias del robot en cámara aérea



Fuente: Autor

Figura 69. Procesos generales que componen el algoritmo de trazado de trayectorias



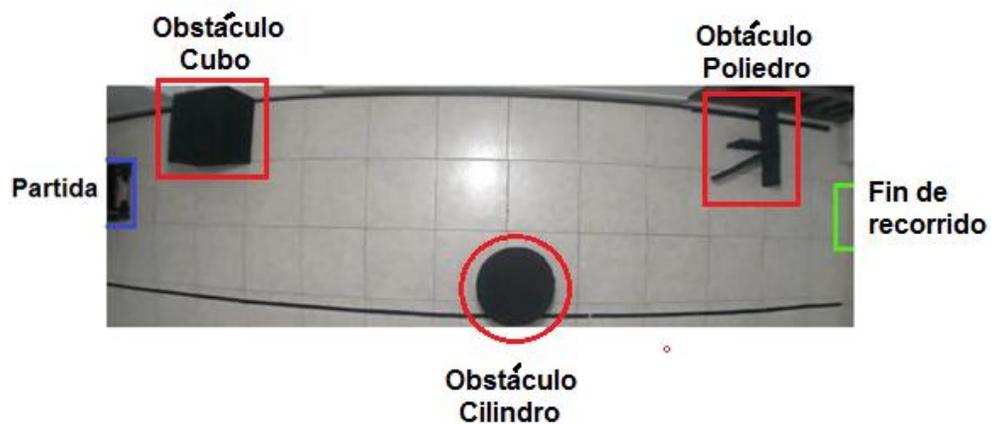
Fuente: Autor

El proceso ilustrado en la figura anterior se repite para cada fotograma, obteniendo al final la trayectoria descrita por el robot en el ambiente de pruebas.

A continuación se muestran los ambientes de prueba y los resultados obtenidos.

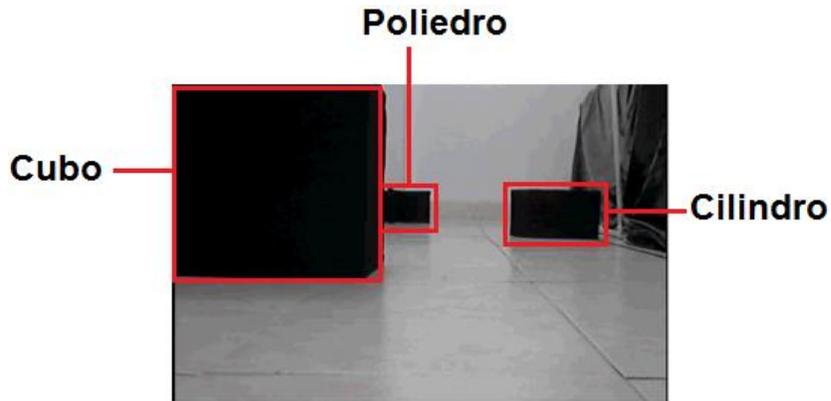
El primer ambiente de pruebas consiste en un corredor de 1.1 m de ancho y 4 m de largo, el cual contiene 3 obstáculos reales (cubo, cilindro y poliedro) distribuidos a través del mismo, adicionalmente, el ambiente está delimitado con un par de cintas negras a los lados para evitar que el robot salga de los límites.

Figura 70. Ambiente real de pruebas objetos reales (cámara aérea)



Fuente: Autor

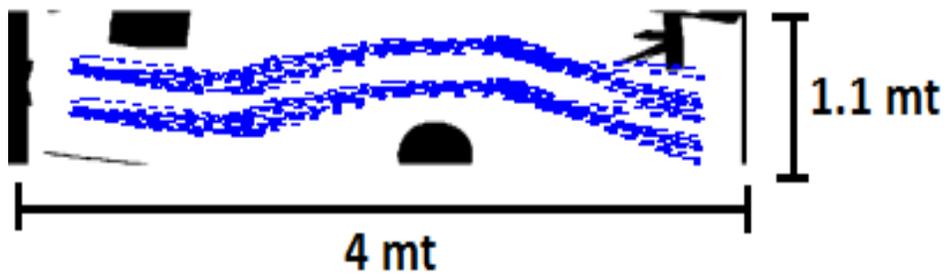
Figura 71. Ambiente real de pruebas objetos reales (cámara frontal del robot)



Fuente: Autor

Las 32 pruebas realizadas con el algoritmo de navegación reactivo y los objetos reales arrojaron las siguientes tendencias:

Figura 72. Gráficas de trayectorias de robot con objetos reales empleando el algoritmo de navegación reactivo

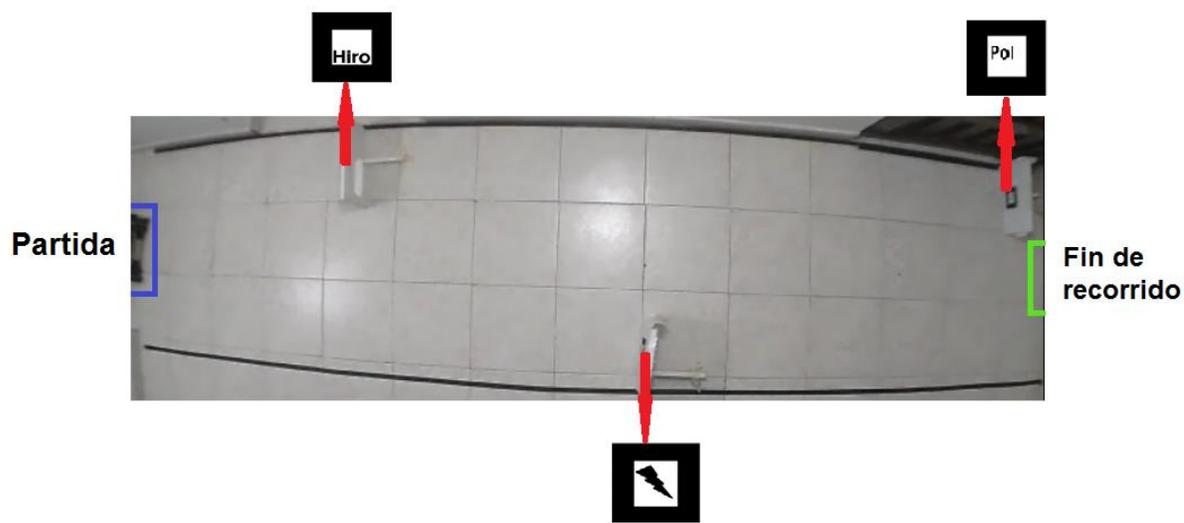


Fuente: Autor

Lo que se puede apreciar es una marcada tendencia en el comportamiento del robot (visto en color azul), en donde el primer y segundo obstáculos son franqueados sin problemas; no obstante en el tercer obstáculo (Poliedro) es visible un acercamiento excesivo en varias trayectorias, al igual que dos, además presentan dos impactos.

Para las pruebas con obstáculos insertados por la RA se mantuvieron las mismas condiciones de las pruebas anteriores, reemplazando los obstáculos reales con marcadores de RA, a los que se les asoció el modelo virtual de cada obstáculo en el BuildAR Pro (Hiro=Cubo, Rayo=Cilindro y Pol=Poliedro). La disposición de los marcadores en el ambiente se puede ver en la Figura 73.

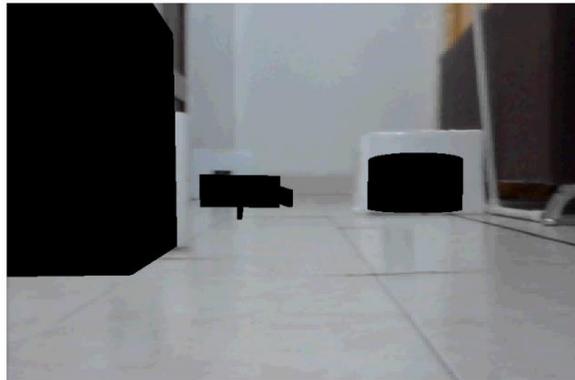
Figura 73. Ambiente de con ubicación de marcadores para la inserción de los objetos virtuales (cámara aérea)



Fuente: Autor

A continuación se puede observar una vista de los objetos insertados por la RA en el ambiente de pruebas, desde el punto de vista del robot.

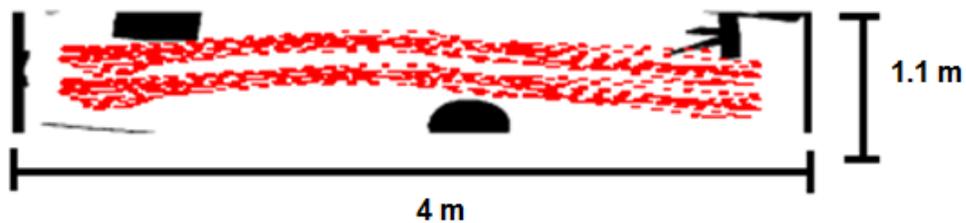
Figura 74. Ambiente de pruebas real con los objetos virtuales insertados por la Realidad Aumentada (cámara frontal del robot)



Fuente: Autor

Las siguientes 32 pruebas (con RA) arrojaron las siguientes tendencias:

Figura 75. Gráficas de trayectorias de robot con objetos insertados por la RA empleando el algoritmo de navegación reactivo



Fuente: Autor

En este caso se presenta una mayor dispersión de las trayectorias, lo que sugiere un comportamiento más inestable y difícil de predecir; asimismo se puede observar que el robot se acercó más a los

obstáculos en las pruebas, en donde se empleó la Realidad Aumentada en reemplazo de los objetos reales, esto debido a la dificultad de mantener los marcadores dentro del volumen de visión del robot, recordando que al perder el marcador también se pierda el objeto virtual afectando la integridad del ambiente simulado.

Esto se puede mejorar de varias formas: implementando marcadores de apoyo para ayudar en la pérdida total o parcial de los marcadores principales; también se pueden aumentar los márgenes de seguridad en los objetos de Realidad Aumentada de modo que el robot se vea obligado a reaccionar desde una distancia más larga, manteniendo durante más tiempo el marcador dentro del volumen de visión de la cámara; esto también se puede lograr aumentando la escala de los objetos virtuales de modo que parezcan más grandes y por consiguiente se aumente la distancia de detección e indirectamente los márgenes de seguridad. Por último, una opción más radical es la integración de una memoria que conserve los objetos virtuales en el caso de la pérdida del marcador.

Por otro lado, el último obstáculo (Poliedro) sigue siendo el más difícil de evadir en los dos casos, generándose en él un mayor número de colisiones, esto sin dejar de lado al cubo, el cual también tuvo (aunque en menor cantidad) colisiones y acercamientos peligrosos, lo que no se observó utilizando los objetos reales. Una opción en el caso del poliedro es complementar su figura con un recuadro negro al nivel del piso o encerrarlo dentro de un cubo para mejorar las probabilidades de detección por parte del sistema de visión, todo esto implementado dentro de la Realidad Aumentada (Carff et al., 2009).

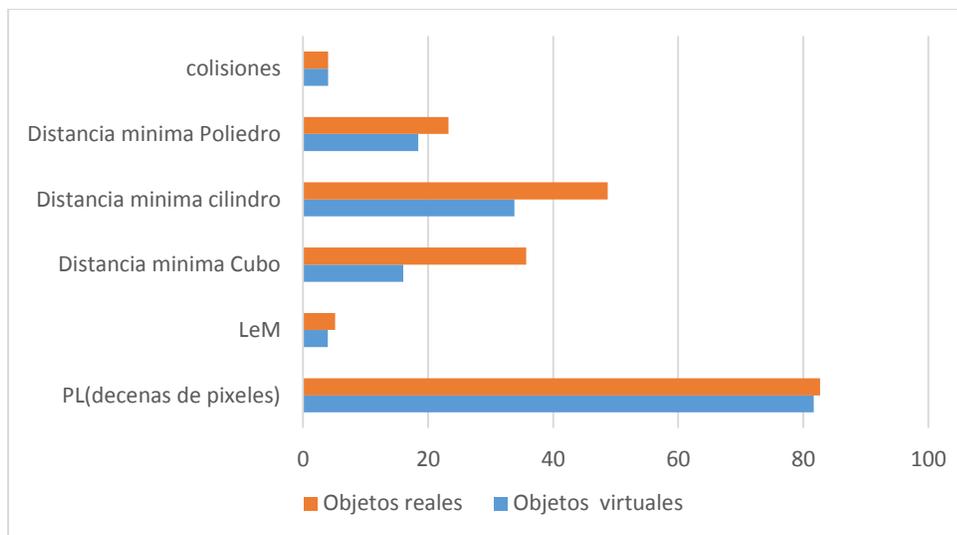
Para realizar un mejor análisis de las trayectorias se aplicaron los criterios de evaluación planteados anteriormente y se consolidaron en una tabla y una gráfica de barras donde se observan sus valores promedio.

Tabla 23. Criterios de evaluación de trayectorias algoritmo reactivo

	PL(decenas de pixeles)	LeM (Periodos de control)	Distancia min Cubo	Distancia min cilindro	Distancia min Poliedro	Colisiones
Objetos virtuales	81,6933	3,9333	16	33,8	18,4	4
Objetos reales	82,7133	5,1333	35,6666	48,7333	23,2666	4

Fuente: Autor

Figura 76. Gráfica de resultados empleando algoritmo de navegación reactivo aplicando métricas de evaluación



Fuente: Autor

Se puede observar que el robot detectó y reaccionó más rápidamente con los objetos reales que con los virtuales, esto sustentado en que las distancias mínimas de acercamiento son más grandes en

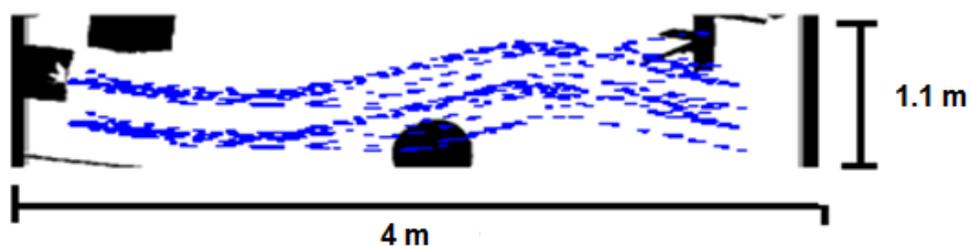
los primeros que en los segundos (Tabla 23). Se debe tener en cuenta que la velocidad de detección puede igualarse aumentando la escala de los objetos virtuales o aumentando el margen de seguridad en estos objetos. Adicionalmente, la cantidad de periodos de decisión (LeM) son mayores empleando objetos reales que virtuales, incrementando a su vez el tiempo de recorrido debido a que la velocidad del robot es constante.

Por otro lado, se refleja en la gráfica una mayor distancia recorrida empleando objetos reales, lo que corrobora el aumento de tiempo de recorrido con respecto a las trayectorias virtuales.

Analizando las distancias mínimas a los obstáculos de forma individual se ve, en el caso del cubo, que el robot se acercó más al virtual que al real, evidenciando que en el caso de los objetos virtuales se debe ampliar la zona de seguridad para prevenir acercamientos excesivos. Se debe aclarar que aunque existen acercamientos peligrosos al cubo, el sistema no falla, pues no presenta colisiones ni con el objeto real ni con el virtual.

Aplicando la misma metodología con el algoritmo de navegación por campos potenciales artificiales se pueden obtener las siguientes tendencias:

Figura 77. Gráficas de trayectorias de robot con objetos reales y algoritmo de campos potenciales

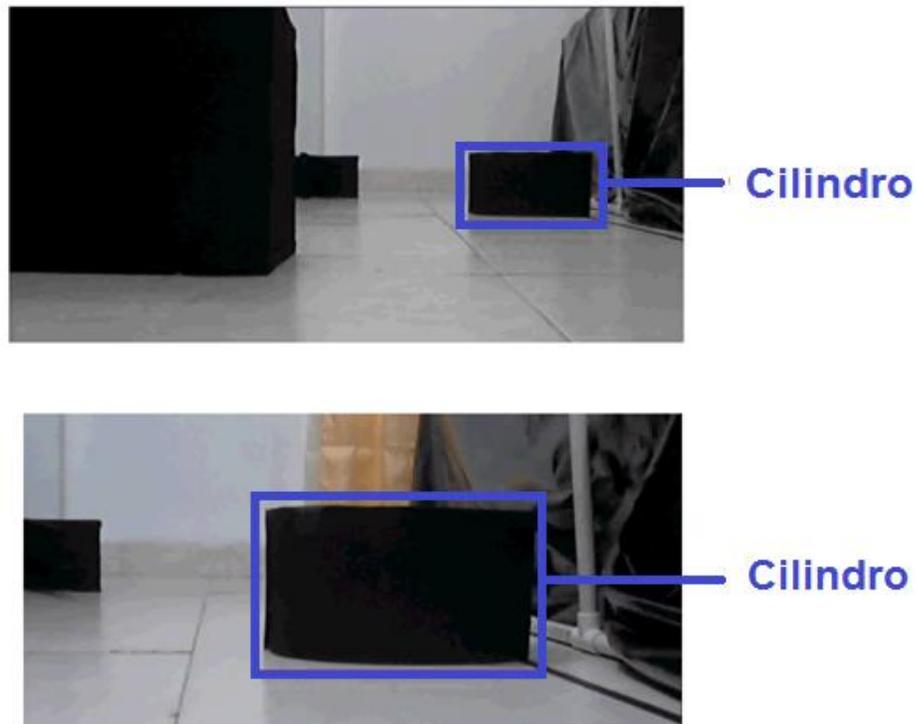


Fuente: Autor

Se aprecia en la figura anterior que algunas trayectorias fueron trazadas sobre los obstáculos *cilindro* y *poliedro*, lo que indica fallos en estos casos concretos, siendo arrastrados los obstáculos por el robot. También se observa que estas colisiones son más frecuentes en el obstáculo *poliedro* que en el *cilindro*. Las dificultades de detección con el poliedro se pueden explicar en parte por la forma del mismo, pues al tener una arista elevada del piso el sistema comete errores de estimación de distancia midiendo una distancia mayor que la real.

Por otro lado los problemas de detección con el obstáculo Cilindro son ocasionados por su curvatura, lo que dificulta la percepción de distancia del robot pues al algoritmo se le dificulta diferenciar un objeto corvo de uno cuadrado.

Figura 78. Imágenes del obstáculo cilindro real captadas por la cámara del robot en el ambiente de pruebas

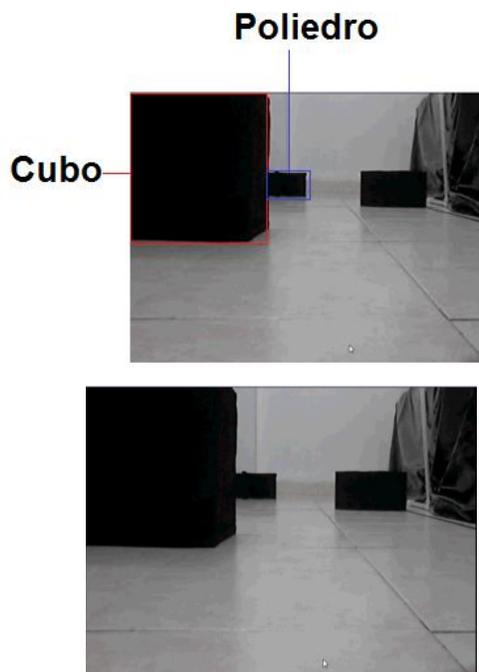


Fuente: Autor

Como se aprecia en las figuras anteriores es fácil confundir el cilindro con un cubo. El algoritmo de tratamiento digital de imágenes es susceptible al mismo problema, lo que origina dificultades en la correcta detección del obstáculo cilindro.

Otra posible causa de esto son las ilusiones ópticas que se presentan cuando se juntan las dos masas de dos obstáculos diferentes, haciéndole creer al sistema que lo que se aprecia es un objeto continuo cuando no lo es.

Figura 79. Ejemplo de efecto óptico de obstáculos independientes unidos en una masa única aparente

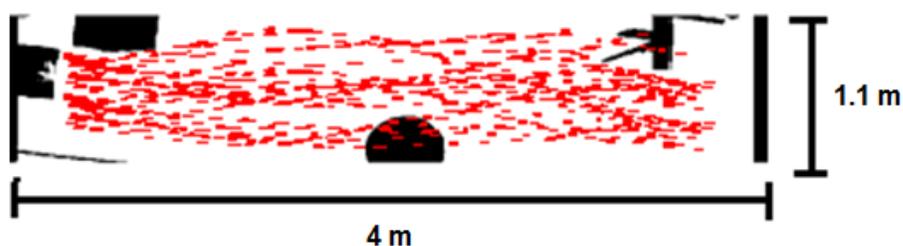


Fuente: Autor

También se presentan irregularidades en la detección, ocasionadas por las vibraciones inherentes al desplazamiento transmitidas a la cámara. Todo esto causó la dispersión de las trayectorias y un acercamiento peligroso a los obstáculos fuera de los márgenes de seguridad.

Para continuar con los análisis de trayectorias, se realizaron 32 pruebas con RA, las cuales arrojaron las siguientes tendencias:

Figura 80. Gráficas de trayectorias de robot con objetos insertados por la RA y algoritmo de campos potenciales



Fuente: Autor

Se puede observar en la Figura 79, que se presentó una dispersión de las trayectorias, esto puede atribuirse, por una parte, a la naturaleza del algoritmo; por otra, a las dificultades en detección de marcadores de RA

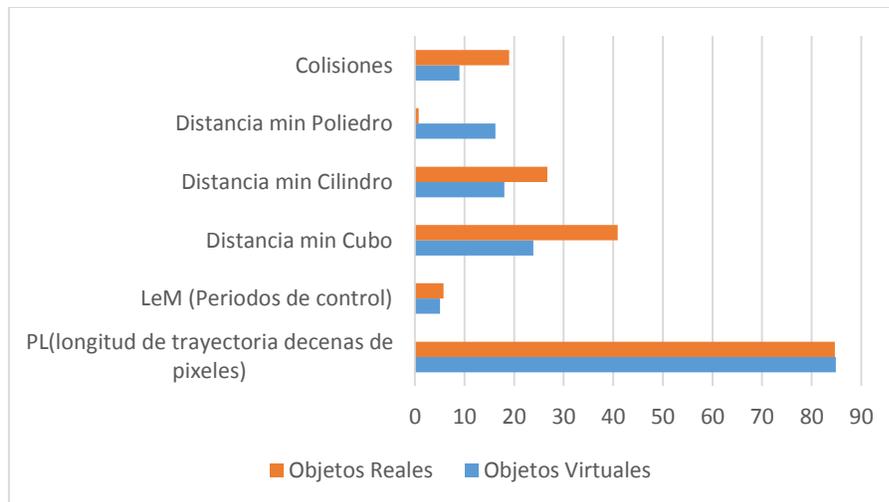
Empleando las mismas métricas de evaluación de trayectorias aplicadas al algoritmo reactivo tenemos:

Tabla 24. Criterios de evaluación de trayectorias algoritmo de navegación por campos potenciales

	PL (longitud de trayectoria decenas de pixeles)	LeM (Periodos de control)	Distancia min Cubo	Distancia min cilindro	Distancia min Poliedro	Colisiones
Objetos Virtuales	84,8461	5,07692	23,8461	18	16,2307	9
Objetos Reales	84,7	5,73333	40,8666	26,6666	0,73333	19

Fuente: Autor

Figura 81. Gráfica de resultados empleando algoritmo de navegación de campos potenciales aplicando métricas de evaluación



Fuente: Autor

Si se comparan las métricas de evaluación de los dos algoritmos de navegación (reactivo y por campos potenciales), se puede observar que en los dos casos el acercamiento mínimo del robot a los obstáculos es mayor en la mayoría de los casos empleando objetos reales que con los virtuales,

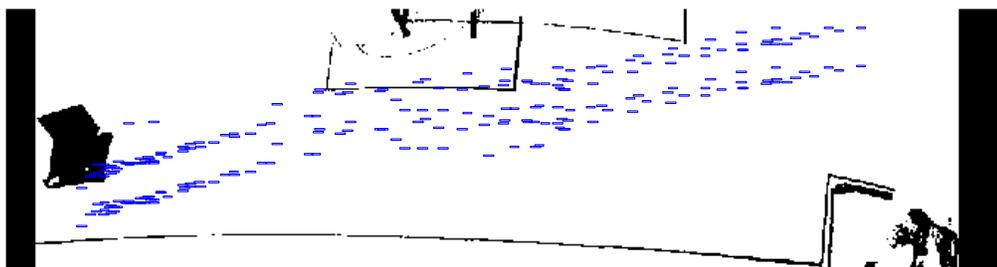
excepto con el obstáculo *poliedro*, en donde la implementación del objeto virtual dio origen a una mayor distancia promedio en el algoritmo de campos potenciales.

Igualmente, se presenta un incremento de los períodos de control empleando el algoritmo de campos potenciales, esto debido a la arquitectura del algoritmo el cual evalúa su trayectoria cada vez que recorre un ciclo completo, a diferencia del algoritmo reactivo que modifica su trayectoria solo cuando detecta un obstáculo.

A continuación tenemos los resultados obtenidos en las pruebas en el ambiente industrial. Se realizaron en total veinte pruebas en total de las cuales diez pertenecen a las pruebas realizadas con los obstáculos reales (llenadora de granos y maquina cortadora); cinco con el algoritmo de navegación de campos potenciales; y cinco con el algoritmo reactivo. Las diez restantes se realizaron de la misma forma, con la diferencia de que la máquina llenadora es reemplazada por un marcador asociado en el software de Realidad Aumentada a una representación digital del mismo.

Las 5 pruebas de navegación en el ambiente industrial, empleando el algoritmo reactivo, arrojaron la siguiente tendencia:

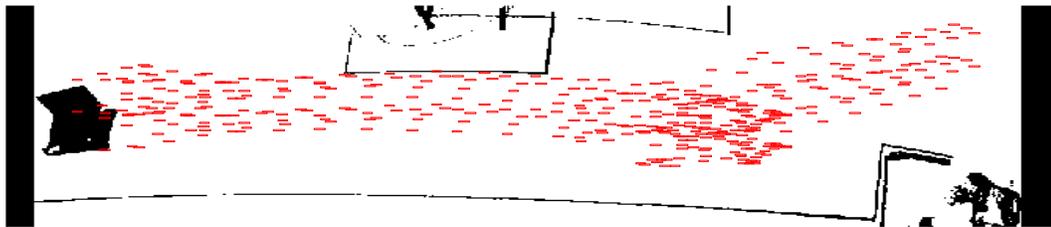
Figura 82. Gráficas de trayectorias de robot en ambiente industrial con objetos reales empleando algoritmo reactivo



Fuente: Autor

Se puede apreciar que el robot tiene problemas para rodear el primer obstáculo (máquina llenadora) el cual se encuentra delimitado por un rectángulo que facilita la detección del obstáculo por parte del robot. Esto es confirmado por el hecho de que al menos 2 de las 5 trayectorias sobrepasaron el área de seguridad que debería separar al robot del objeto. Sin embargo, el robot no presenta problemas a la hora de pasar por el desvío del corredor, aunque se acerca peligrosamente a su esquina superior derecha.

Figura 83. Gráficas de trayectorias de robot en ambiente industrial con objetos virtuales empleando algoritmo reactivo



Fuente: Autor

Analizando lo ocurrido con el algoritmo reactivo, se puede decir que a pesar de que muestra acercamientos a las líneas que delimitan su mundo, no invade la zona fuera del perímetro de seguridad ni colisiona con los obstáculos, lo que muestra la efectividad de dicho algoritmo de navegación en este tipo de ambientes

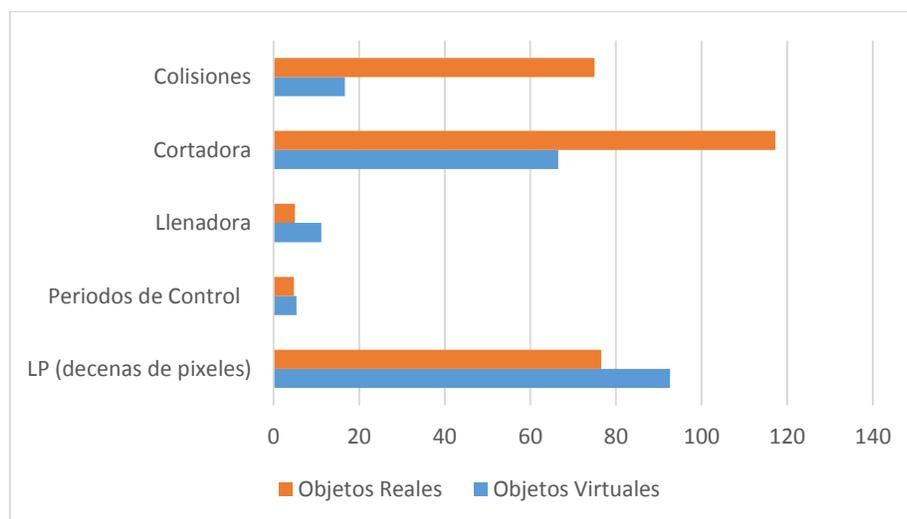
Como en los casos anteriores, se aplicaron a este conjunto de trayectorias criterios de evaluación estándar para realizar un análisis cuantitativo del comportamiento del robot bajo estas condiciones.

Tabla 25. Criterios de evaluación de trayectorias algoritmo reactivo en ambiente industrial

	PL (longitud de trayectoria decenas de pixeles)	LeM (Periodos de control)	Llenadora	Cortadora	Colisiones
Objetos Virtuales	92,65	5,33333333	11,1666667	66,5	0,1666666
Objetos Reales	76,55	4,75	5	117,25	0,75

Fuente: Autor

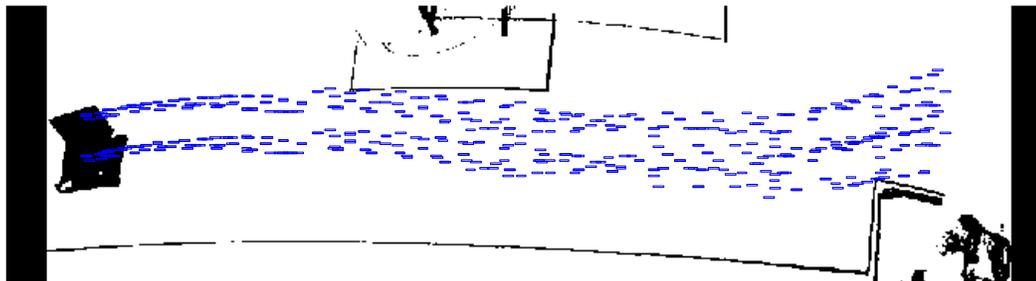
Figura 84. Gráfica de criterios de evaluación de trayectorias algoritmo reactivo en ambiente industrial



Fuente: Autor

Realizando un análisis cuantitativo de las métricas de evaluación, se puede decir que el comportamiento del algoritmo reactivo con los objetos virtuales es mejor que el presentado con los objetos reales, esto a pesar de lo que se observa en el parámetro PL (longitud de trayectoria) pues, en promedio, el robot recorrió más distancia con los objetos virtuales. Con los objetos reales presenta una mayor probabilidad de colisiones (un 75% de probabilidad de colisiones con los objetos reales frente a un 16% de los virtuales). Adicionalmente, se puede observar que en los dos casos (empleando objetos virtuales y reales) las distancias mínimas del robot a la cortadora son mucho mayores que las presentadas con respecto a la llenadora, lo cual se puede explicar en el hecho de que mientras la cortadora presenta una base sólida la llenadora está montada sobre pilares de acero, lo que dificulta la detección del mismo por parte del sistema de visión artificial.

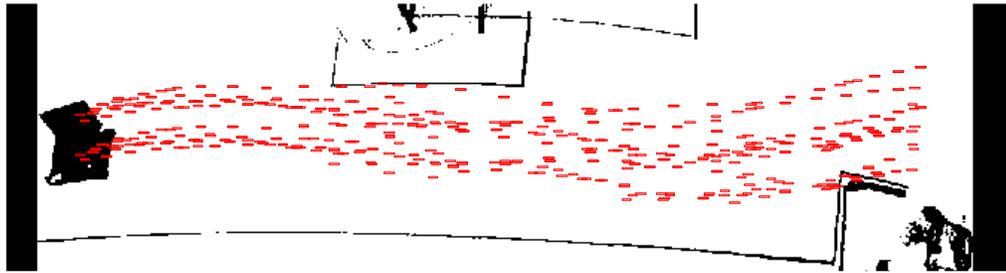
Figura 85. Gráficas de trayectorias de robot en ambiente industrial con objetos virtuales empleando algoritmo reactivo



Fuente: Autor

Al incorporar el objeto virtual reemplazando el primer obstáculo (máquina de llenado), este algoritmo mejoró su comportamiento, pues no sobrepasó la zona de seguridad preestablecida y su comportamiento fue más consistente a través de la prueba; pero persisten los problemas con las esquinas del desvío de la ruta, que aunque en este caso no colisionó con ellas, sí tuvo acercamientos peligrosos es decir acercamientos por debajo de 3 cm.

Figura 86. Gráficas de trayectorias de robot en ambiente industrial con objetos virtuales empleando algoritmo de campos potenciales



Fuente: Autor

El comportamiento del robot con el obstáculo virtual y el algoritmo por campos potenciales mejoró levemente con relación al empleo del obstáculo real, pues si bien hubo un acercamiento peligroso al perímetro del objeto virtual, las otras 4 trayectorias se mantuvieron a una distancia prudencial. Caso contrario ocurrió en el desvío, en donde el comportamiento del robot empeoró, pues se acercó peligrosamente al menos dos veces a la esquina inferior derecha.

Cabe resaltar que el algoritmo reactivo se comportó mucho mejor que el anterior en el desvío empleando objetos virtuales pues no se acercó a ninguna de las esquinas, conservando su distancia en todas las trayectorias.

Figura 87. Comparación de trayectorias del robot en ambiente industrial empleando objetos virtuales a) algoritmo reactivo b) algoritmo por campos potenciales



a) b)

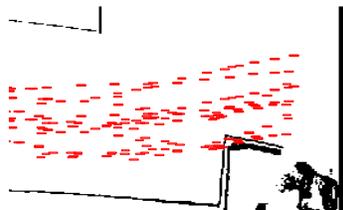
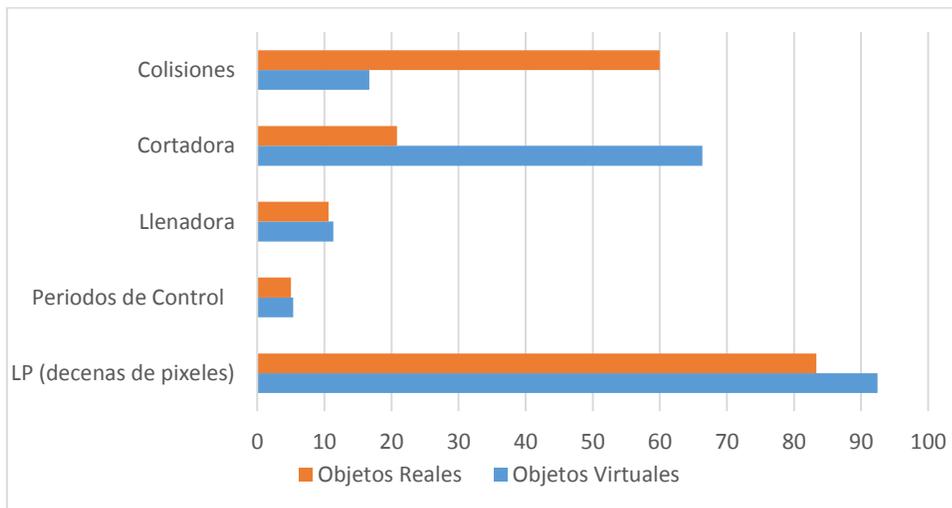


Tabla 26. Criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente industrial

	PL (longitud de trayectoria decenas de pixeles)	LeM (Periodos de control)	Llenadora	Cortadora	Colisiones
Objetos Virtuales	92,45	5,33333333	11,3333333	66,3333333	0,1666666
Objetos Reales	83,34	5	10,6	20,8	0,6

Fuente: Autor

Figura 88. Gráfica de criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente industrial



Fuente: Autor

Analizando el comportamiento del robot según los parámetros de evaluación, empleando el algoritmo de campos potenciales, se puede decir que su comportamiento es muy similar con respecto al observado con el algoritmo reactivo, es decir, con los objetos virtuales se comporta mucho mejor que con los objetos reales, pues aunque en promedio el robot recorrió más distancia

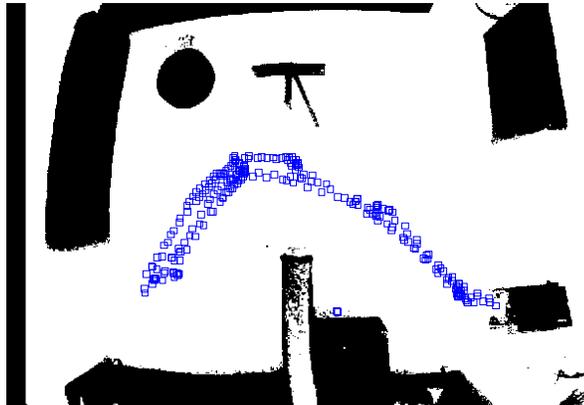
con los primeros que con los segundos, este último presenta una mayor probabilidad de colisiones con respecto a los resultados obtenidos con los objetos virtuales, discrepando solo en la distancia mínima del robot a la cortadora, donde se comportó mucho mejor con los objetos reales. Cabe mencionar que la probabilidad de tener una colisión con el objeto virtual se mantiene estable (16%) y que la probabilidad de tener una colisión con el objeto real disminuyó un 10%, lo que evidencia una mejora con respecto al algoritmo reactivo.

Otro aspecto que se debe mencionar es que el robot logró navegar en ambientes aumentados, es decir, sin los obstáculos físicos presentes, y los problemas en esta área pueden ser subsanados ya sea con una recalibración del sistema o con mejoras en los marcadores.

Debe considerarse la iluminación como un componente determinante en el comportamiento del robot, pues mientras que en las pruebas iniciales en los corredores la iluminación se encontraba a una distancia máxima de 2.5 m del suelo, en la planta en donde se realizaron las pruebas industriales la altura de la iluminación superaba los 4m; adicionalmente, el ambiente industrial presentaba una densidad de luminarias mayor que en los corredores iniciales, generando un ambiente con una iluminación más uniforme y difusa. Esto explica, en parte, las discrepancias encontradas en los resultados entre las pruebas realizadas en los corredores iniciales y el ambiente industrial

Por último se presentan las pruebas hechas en un ambiente distinto a un corredor. Se realizaron en total veinte pruebas, de las cuales diez pertenecen a las pruebas realizadas con los obstáculos reales (cubo, cilindro, poliedro); cinco con el algoritmo de navegación de campos potenciales; y cinco con el algoritmo reactivo. Las diez restantes se distribuyeron de la misma forma, con la diferencia que se realizaron empleando objetos virtuales. Las 5 pruebas de navegación empleando el algoritmo reactivo con objetos reales arrojaron la siguiente tendencia (Figura 89):

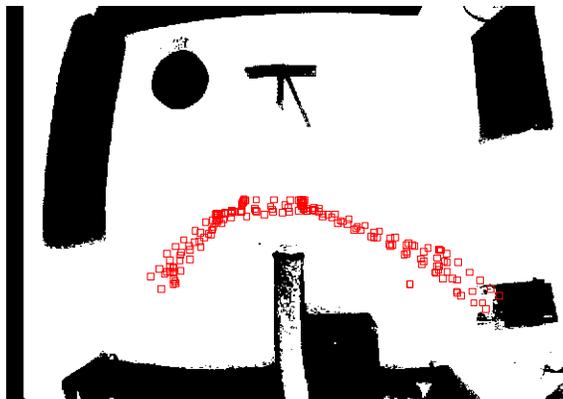
Figura 89. Gráficas de trayectorias de robot en ambiente rectangular con objetos reales empleando algoritmo reactivo



Fuente: Autor

Se puede apreciar que el robot evade los obstáculos sin problemas, adicionalmente las trayectorias siguen una tendencia clara, la cual conduce al punto designado como final del recorrido en todos los casos

Figura 90. Gráficas de trayectorias de robot en ambiente rectangular con objetos virtuales empleando algoritmo reactivo



Fuente: Autor

En este caso se evidencia un mayor acercamiento del robot al paral divisorio; igualmente se aprecia una tendencia clara desde el inicio hasta el final del recorrido, mejorando levemente en este algoritmo al presentarse menos dispersión en los recorridos.

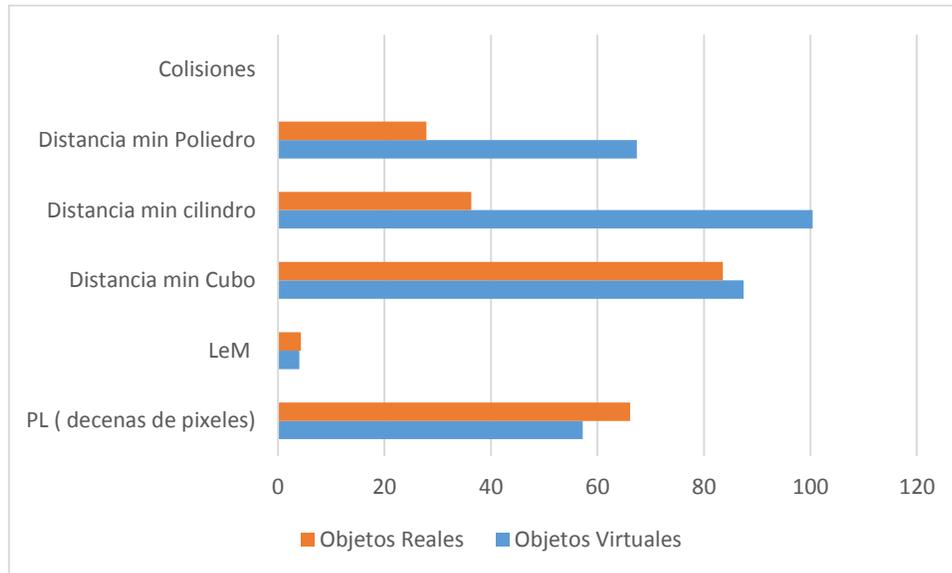
A continuación, se mostrará el resultado de aplicar el conjunto de criterios de evaluación previamente definidos a las trayectorias del robot, empleando los obstáculos reales y los insertados por la RA.

Tabla 27. Criterios de evaluación de trayectorias algoritmo reactivo en ambiente rectangular

	PL (longitud de trayectoria decenas de píxeles)	LeM (Periodos de control)	Distancia min Cubo	Distancia min cilindro	Distancia min Poliedro	Colisiones
Objetos Virtuales	57,2428571	4	87,4285714	100,428571	67,4285714	0
Objetos Reales	66,1571429	4,28571429	83,5714286	36,2857143	27,8571429	0

Fuente: Autor

Figura 91. Gráfica de criterios de evaluación de trayectorias algoritmo reactivo en ambiente rectangular

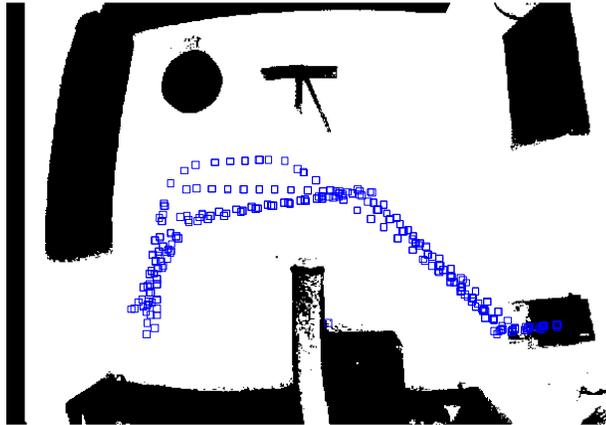


Fuente: Autor

Teniendo en cuenta los parámetros de evaluación aplicados a las trayectorias del robot, generadas con el algoritmo reactivo en el ambiente rectangular, se puede afirmar que el comportamiento presentado con los objetos virtuales es mejor que con los reales, evidenciado en mayores distancias a los objetos representados de forma digital y menor distancia recorrida para llegar al fin del recorrido. Estos efectos se pueden explicar en el hecho de que los marcadores pudieron ser ubicados de mejor forma en este ambiente, pues a diferencia de los escenarios anteriores en donde se simulaba un corredor, en este caso los espacios más amplios dieron la posibilidad de una mejor movilidad del robot y un abordaje de los marcadores por parte de este en ángulos más amplios de ataque.

Después de realizar las pruebas en el ambiente rectángulos con el algoritmo reactivo, se implementó el algoritmo por campos potenciales generando las siguientes trayectorias:

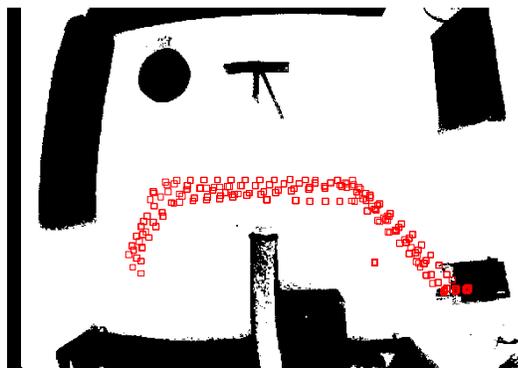
Figura 92. Gráficas de trayectorias de robot en ambiente rectangular con objetos reales empleando algoritmo por campos potenciales



Fuente: Autor

En este caso, el algoritmo de campos potenciales afectó la dispersión de las trayectorias en la segunda mitad del recorrido, pero a pesar de esto todas pudieron llegar hasta la salida de laberinto.

Figura 93. Gráficas de trayectorias de robot en ambiente rectangular con objetos virtuales empleando algoritmo por campos potenciales



Fuente: Autor

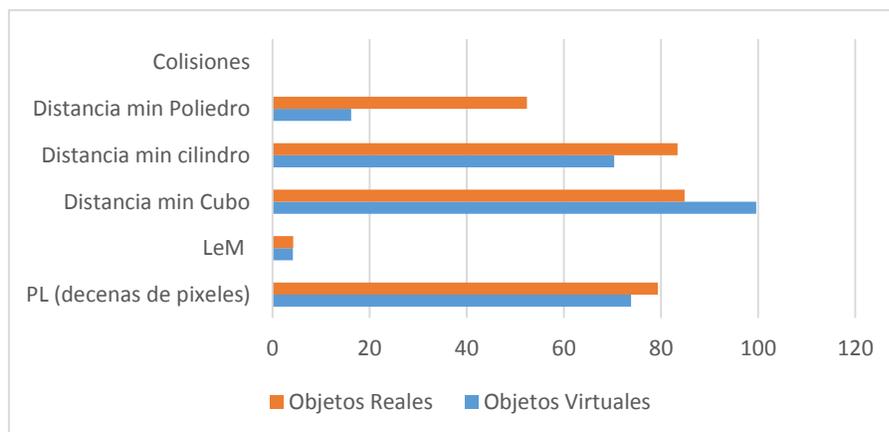
La figura anterior muestra una gran consistencia en el comportamiento del robot en este ambiente empleando objetos insertados por la RA, lo que se puede apreciar en la baja dispersión de las trayectorias, las cuales condujeron al robot satisfactoriamente hasta la salida del laberinto.

Tabla 28. Criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente rectangular

	PL (longitud de trayectoria decenas de píxeles)	LeM (Períodos de control)	Distancia min Cubo	Distancia min cilindro	Distancia min Poliedro	Colisiones
Objetos Virtuales	73,86	4,2	99,6	70,375	16,2307	0
Objetos Reales	79,375	4,25	84,875	83,4	52,4	0

Fuente: Autor

Figura 94. Gráfica de criterios de evaluación de trayectorias algoritmo por campos potenciales en ambiente rectangular



Fuente: Autor

Analizando los criterios de evaluación, en este caso el robot se acercó en mayor medida a los objetos virtuales que a los reales, exceptuando en el cubo, donde se comportó de forma distinta. Igualmente la distancia recorrida hacia la salida del laberinto fue menor con los objetos virtuales, lo que indica un menor tiempo de recorrido empleando estos objetos.

En este ambiente, a diferencia de los dos anteriores (corredor e industrial), se puede ver con más claridad la búsqueda de la salida del laberinto por parte del robot en todos los casos, indistintamente de si se emplean objetos reales o virtuales, algoritmo reactivo o por campos potenciales.

3.1.1. Análisis estadístico de trayectorias

Para probar la similitud estadística que existe entre los dos algoritmos planteados y el uso de obstáculos físicos y virtuales, se realizó un análisis de las métricas de evaluación de trayectorias aplicando una prueba estadística de hipótesis a dichos parámetros. Esta prueba está basada en el valor de la varianza y define si tal valor es suficiente para diferenciar entre tipos de muestras. Para esto se plantean dos hipótesis:

- H_0 (Hipótesis nula) = No existen diferencias significativas entre las muestras, es decir, se consideran pertenecientes a una misma varianza.
- H_1 (Hipótesis alternativa) = Existen diferencias significativas entre las muestras, es decir, se consideran pertenecientes a varianzas diferentes.

Teniendo en cuenta lo anterior, se plantea que H_0 (Hipótesis nula) es Verdadera si $\mathbf{P-value} > \alpha$, y H_1 (Hipótesis alternativa) es Verdadera si $\mathbf{P-value} \leq \alpha$, donde $\mathbf{P-value}$ es calculado a través de la prueba *T student*, y α es el valor de significancia (0,05).

Aplicando la prueba de hipótesis a los parámetros de las trayectorias (treinta y dos trayectorias) del algoritmo reactivo desarrolladas en los corredores (Tabla 29), se puede concluir que tanto la longitud de trayectoria recorrida como la distancia mínima de acercamiento al obstáculo *poliedro* no presentan diferencias significativas entre sus muestras, desde el punto de vista estadístico. Se puede decir con certeza que el comportamiento del robot con respecto a estos dos parámetros es similar, tanto con los objetos reales como con los virtuales.

Los resultados de las pruebas se pueden observar en la tabla siguiente:

Tabla 29. Prueba de hipótesis con algoritmo reactivo

	PL (longitud de trayectoria decenas de píxeles)	LeM (Períodos de control)	Distancia mínima a obstáculo		
			Cubo	Cilindro	Poliedro
P-valué	0,3389	0,00265	0,0000044	0,000287	0,32094
Hipótesis Verdadera	H0	H1	H1	H1	H0

Fuente: Autor

Por otro lado, el resto de los parámetros (distancia mínima cubo, distancia mínima cilindro y periodos de control) presentan diferencias significativas entre sus respectivas muestras, evidenciando un comportamiento diferente entre los datos obtenidos con los objetos reales y con los virtuales.

Tabla 30. Prueba de hipótesis con algoritmo de campos potenciales

	Longitud de la trayectoria recorrida (PL)	Periodos de Control (LeM)	Distancia mínima a obstáculo		
			Cubo	Cilindro	Poliedro
P-valué	0,7253	0,13544	0,002935	0,63883	0,00123
Hipótesis Verdadera	H0	H0	H1	H0	H1

Fuente: Autor

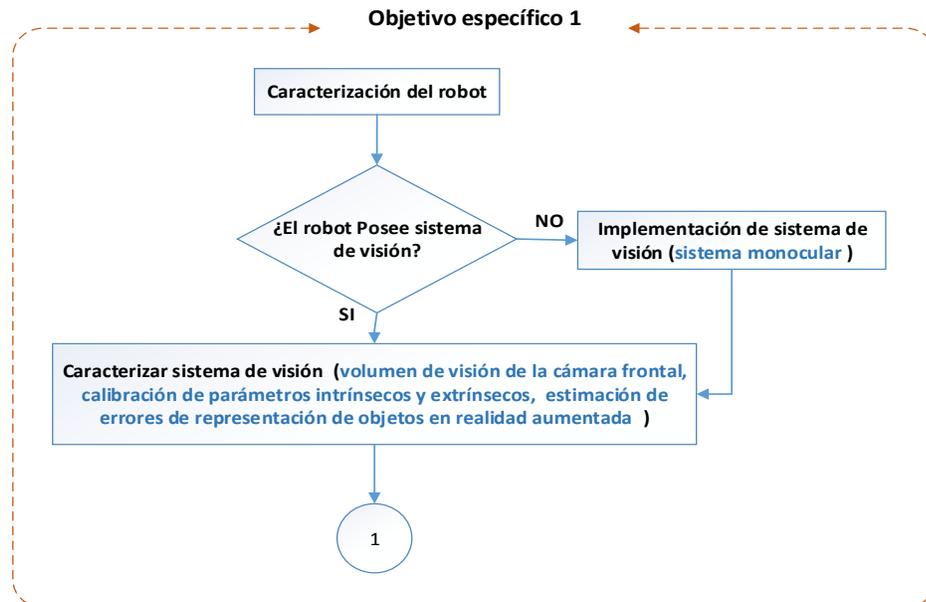
En el caso de la tabla anterior, se puede observar que las métricas de evaluación de la longitud de la trayectoria recorrida, los periodos de control y la distancia mínima al cilindro no presentan diferencias significativas entre las muestras obtenidas con los objetos reales y los aumentados, desde el punto de vista estadístico. La tabla también evidencia que en los dos casos el sistema reacciona a la presencia de los obstáculos *cubo* y *poliedro*, pero que la similitud es mayor en los demás parámetros. Esto se puede mejorar recalibrando el sistema, implementando mejoras en la distribución de la iluminación y de los marcadores, con miras a aumentar la similitud del comportamiento del robot frente a estos dos objetos, ya sean aumentados o reales.

Lo anterior se puede resumir en el hecho de que el algoritmo por campos potenciales presenta similitudes estadísticas en los parámetros PL, LeM y distancia mínima de acercamiento al cilindro, pero que hablando de los otros dos parámetros no presenta comportamientos similares.

3.2. Planteamiento de metodología

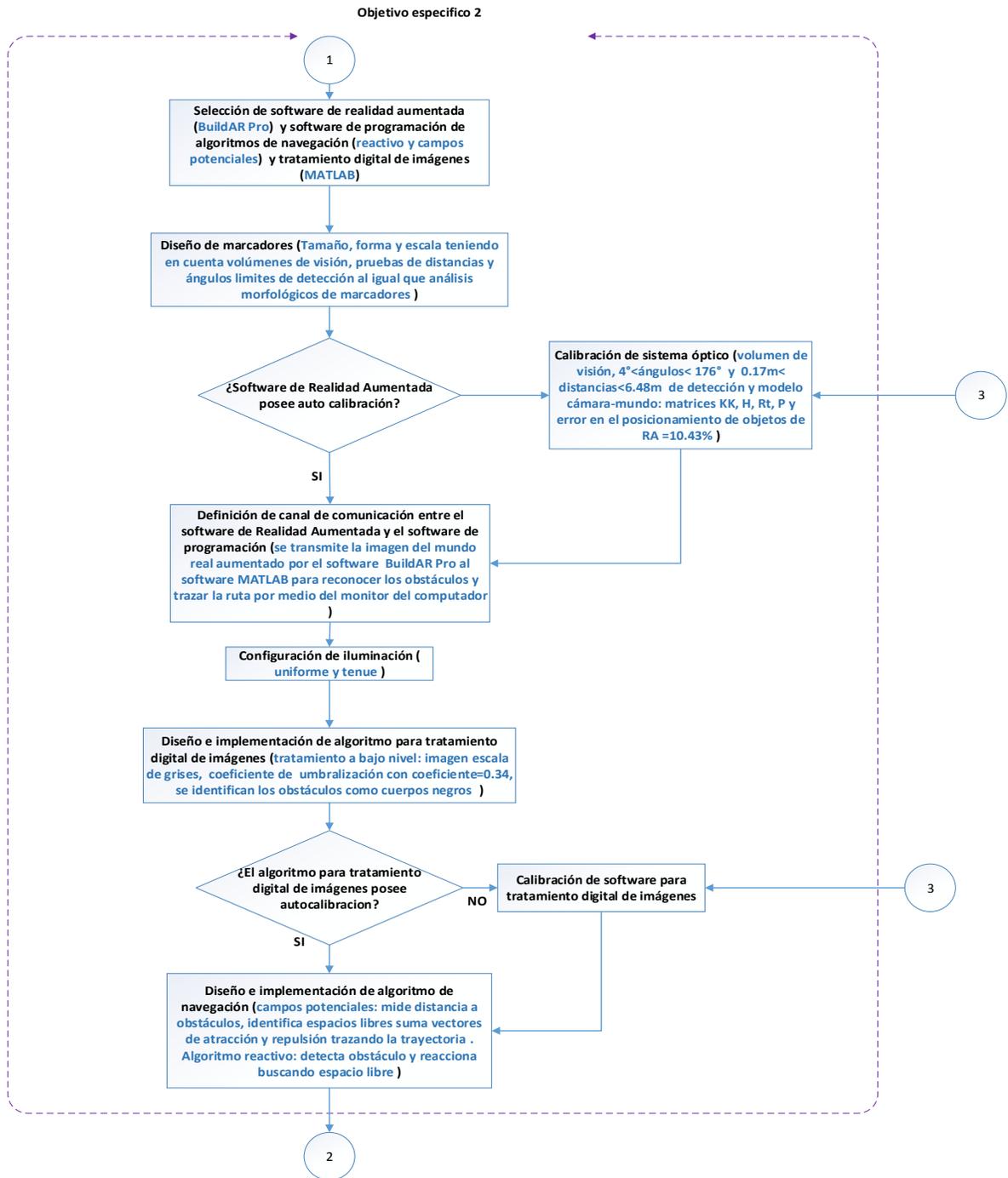
En este subcapítulo se mostrará la metodología propuesta haciendo énfasis en los objetivos que dan origen a cada una de sus tres fases: la primera, se concentra en la caracterización del volumen de visión, en donde se define el tipo de sistema óptico y la representación matemática de los parámetros de dicho sistema; la segunda fase, se centra en determinar una estrategia de integración entre la Realidad Aumentada con marcadores, el sistema de visión artificial y el algoritmo de navegación, en donde se definen los software empleados en la Realidad Aumentada al igual que en la construcción de los algoritmos de tratamiento digital de imágenes y navegación, además se definen las condiciones ideales del ambiente de operación y los parámetros de calibración de los sistemas; la tercera fase, se concentra en la evaluación del sistema en sí (Figura 95), Etapa 1 de la metodología para un sistemas de visión artificial apoyadas con Realidad Aumentada en base a marcadores para un robot móvil (completa).

Figura 95. Etapa 1 de la metodología para sistemas de visión artificial apoyadas con Realidad Aumentada con base en marcadores para un robot móvil



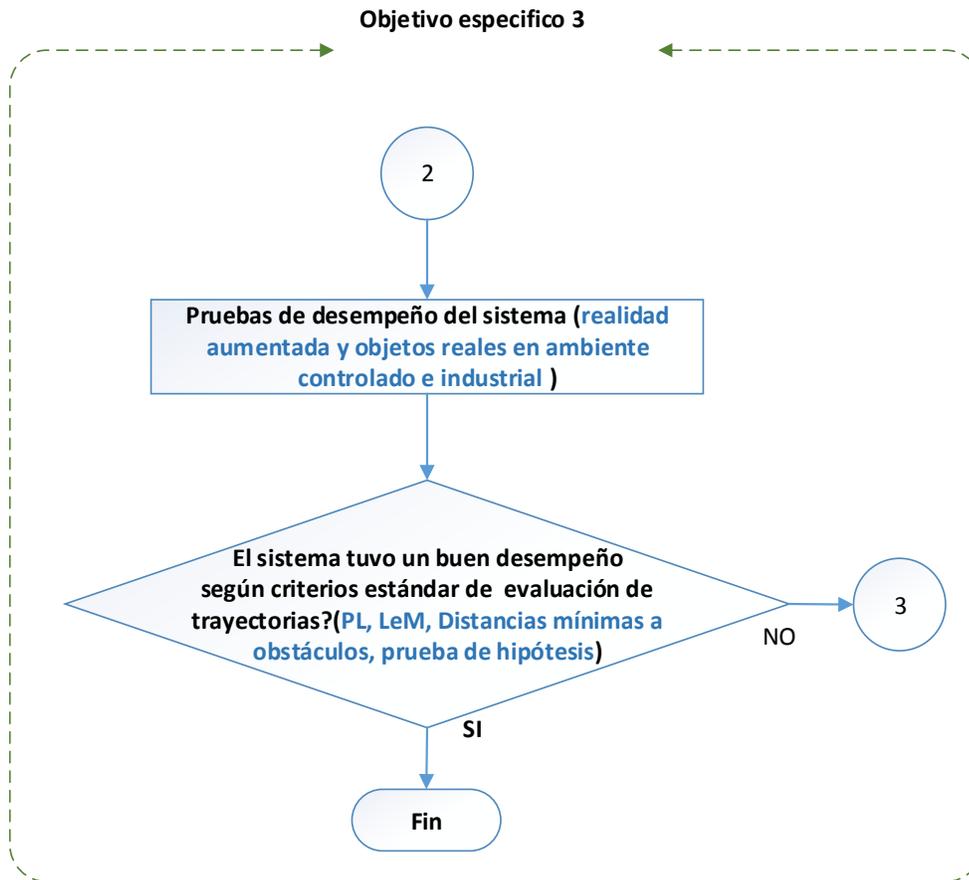
Fuente: Autor

Figura 96. Etapa 2 de la metodología para sistemas de visión artificial apoyadas con Realidad Aumentada en base a marcadores para un robot móvil



Fuente: Autor

Figura 97. Etapa 3 de la metodología para sistemas de visión artificial apoyadas con Realidad Aumentada en base a marcadores para un robot móvil (completa)



Fuente: Autor

En conclusión, se tienen 3 etapas que responden a cada uno de los objetivos del diseño de la metodología. Los elementos a destacar son: la calibración y modelación de la cámara frontal del robot, los algoritmos de navegación, la modelación de la cámara estimada por la Realidad Aumentada, la evaluación de las trayectorias empleando criterios de desempeño plasmados en la literatura (Muñoz et al., 2007) y la implementación de la prueba de hipótesis basada en la *T-Student*. Dicha prueba se realizó tomando dos muestras: la primera, originada por las trayectorias

del robot con los objetos reales; la segunda, originada por el robot con objetos insertados por la RA, estableciendo de forma estadística si son similares entre sí.

4. CONCLUSIÓN

La obtención de los parámetros intrínsecos de la cámara simulada por la Realidad Aumentada es un paso fundamental en la valoración de sistemas que integran la RA con visión artificial en robots móviles, pues con ello se pueden determinar los niveles de error en la representación de objetos aumentados (BuilldAR pro =5%) y realizar las correcciones necesarias para eliminar o mitigar las consecuencias de dicho error

La elección de la pantalla del PC como canal de comunicación entre la Realidad Aumentada, el sistema de visión artificial y el algoritmo de navegación, fue una decisión que permitió integrar los diferentes software.

Las 168 pruebas de navegación realizadas muestran que la integración entre la Realidad Aumentada con marcadores y el sistema de visión es posible; además, como se aprecia en los dos algoritmos de navegación implementados en este trabajo (campos potenciales y reactivo), es adaptable a varios algoritmos en una representación de imagen, lo que brinda una herramienta para la navegación en ambientes estáticos con objetos reales y aumentados.

La gran ventaja de implementar escenarios con Realidad Aumentada es la flexibilidad en los diseños de los ambientes, pues se puede pasar de la simulación de un ambiente compuesto por obstáculos geoméricamente sencillos a uno industrial, sin la necesidad de desplazar objetos de gran talla y peso; solo basta con trasladar un marcador que represente dichos objetos.

Las pruebas de hipótesis practicadas a las trayectorias del robot revelaron que la distancia recorrida en el ambiente de pruebas con los objetos reales y los insertados por la RA, en el caso de los dos algoritmos de navegación, es estadísticamente similar, lo que indica que en los dos casos el robot recorre aproximadamente la misma distancia empleando obstáculos aumentados o reales

La posición, tamaño y forma de los marcadores, el sistema de iluminación, el tamaño de los obstáculos empleados, las distancias mínimas de proximidad a los obstáculos, todo esto afecta la trayectoria del robot encontrándose diferencias entre el empleo de modelos virtuales y reales por lo cual es necesario tener en cuenta estos parámetros, con el fin de realizar las correcciones necesarias para aumentar la similitud de los comportamientos del robot utilizando los objetos reales y los aumentados.

Las características morfológicas de la parte interna de los marcadores (Envolvente convexa, número de elementos, número de centroides, porcentaje de área convexa y área simple) influyen en la distancia y ángulo de identificación de los mismos, por lo cual es fundamental establecer un rango de valores adecuados para la tarea designada (solidez de la imagen <0.5 , el área convexa $>35\%$, número de centroides = 1 para este caso).

Se puede afirmar que la caracterización del sistema de visión del robot, en conjunto con el tratamiento digital de imágenes basado en segmentación y umbralización, se integró de forma eficiente con los algoritmos de navegación reactivo y por campos potenciales artificiales, logrando que el robot navegara en tres ambientes distintos con elementos reales y aumentados, pudiéndose establecer de forma estadística que el comportamiento del robot con elementos virtuales y reales es similar. Todo lo anterior sirve como insumo para la generación de una metodología aplicable a robots móviles con sistemas de visión artificial apoyados por Realidad Aumentada.

5. TRABAJOS FUTUROS

Se debe aclarar que este trabajo no aborda el tema de la mejora de visión en los sistemas robóticos aplicando la Realidad Aumentada, solo se aborda el tema de la integración de la Realidad Aumentada en los sistema de visión artificial de robot móviles, sustentado en una metodología propuesta por el autor, lo que significa que se recomienda que dicho tema (el mejoramiento de la visión artificialmente integrando Realidad Aumentada) sea desarrollado en otros trabajos.

Otro aspecto que no se aborda a profundidad es la solución de problemas de navegación derivados de los sistemas de visión y la Realidad Aumentada, como los falsos positivos, las limitaciones en la representación de los ambientes debido a la ubicación de los marcadores dentro del volumen de visión de la cámara del robot y los comportamientos que el robot debe tomar cuando se encuentra un obstáculo infranqueable. Por esta razón se recomienda ahondar en este tema en trabajos futuros.

REFERENCIAS

- Abdulmuslih Alsirhani, M. (2012). *Análisis de sistemas de Realidad Aumentada y metodología para el desarrollo de aplicaciones educativas*. España.
- Aleksandrovich, Y. D. & Gennadievich, P. G. (2013). Mobile Robot Navigation Based on Artificial Landmarks with Machine. Vision System, 24(11), 1467–1472. doi:10.5829/idosi.wasj.2013.24.11.7010
- Aracil, R. (2002). *Robot Automática. Robots y Sistemas Sensoriales* (1st ed.). España: Prentice Hall.
- Arias, M. (2009). *Navegación de robot móvil en interiores utilizando flujo óptico a partir de imágenes panorámicas*. Universidad Nacional de Colombia.
- Association Robotic Industries (2015). Robotic Industries Association. Retrieved May 31, 2015, from <http://www.robotics.org/company-profile-detail.cfm/Internal/Robotic-Industries-Association/company/319>
- Baker, D. K., Fryberger, C. T., & Ponce, B. A. (2015). The Emergence of Augmented Reality in Orthopaedic Surgery and Education. *The Orthopaedic Journal at Harvard Medical School*, 23, 8–16.
- Barranco, A. I.; Martínez Díaz, S. & Gómez, J. (2010). *Visión estereoscópica por computadora con Matlab y OpenCV*. (A. I. Barranco Gutiérrez & S. Martínez D, Eds.) (1st ed.). Hardcover (dust-jacket). Retrieved from <http://posgrado.itlp.edu.mx/barranco/VisionCurso/libro.pdf>
- BędkowskiMas, J. & Masłowski, A. (2009). Augmented reality approach for mobile multi-robotic system development and integration. *Techniki Komputerowe*, 1, 15–24.

- Bekris, K. E.; Argyros, A. A. & Kavraki, L. E. (2004). Angle-based methods for mobile robot navigation: reaching the entire plane. *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, 3(April), 2373–2378.* doi:10.1109/ROBOT.2004.1307416
- Bermúdez, G. (2002). Robots móviles. Teoría, aplicaciones y experiencias. *TECNURA, 10, 5, 6–17.*
- Bischoff, R. & Kazi, A. (2004). Perspectives on Augmented Reality Based Human-Robot Interaction with Industrial Robots, 3226–3231.
- Bonilla, M. N. I.; Novelo, F. J. Q.; Enríquez, I. J. G. & Cortés, J. M. R. (2009). Desplazamiento de un robot con localización y evasión de obstáculos por visión y ultrasonido. *Congreso Nacional de Ingeniería Electrónica del Golfo -CONAGOLFO, 2009, 3–7.*
- Cano Quinttero, J. B. (2007). *Sistema de control y comando para un robot móvil todo terreno.* Colombia.
- Carff, J.; Johnson, M.; El-Sheikh, E. M. & Pratt, J. E. (2009). Human-robot team navigation in visually complex environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, 3043–3050.*
- Carrizo, J.; García, G.; Secchi, H. & Mut, V. (2006). Sistema de visión 2D1/2. Una aplicación en robótica móvil. In *IV Jornadas argentinas de robótica* (pp. 30–36). Córdoba.
- Castillo, R. A.; Andrés, I.; López, D. & León, M. H. (2006). Localización espacial de un punto XYZ mediante visión artificial. *Periódico Universidad Nacional de Colombia, 16, 15–27.*
- Caudell, T. P. & Mizell, D. W. (1992). Augmented reality: an application of heads-up display technology to manual manufacturing processes. *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, ii.* doi:10.1109/HICSS.1992.183317

- Chen, I. Y.; Macdonald, B. & Burkhard, W. (2008). Markerless Augmented Reality for Robotic Helicopter Applications. *Robot Vision ,Second International Workshop*, 125–138.
- Chen, I. Y.; MacDonald, B. & Wunsche, B. (2009). Mixed Reality Simulation for Mobile Robots. *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, 232 – 237. doi:10.1109/ROBOT.2009.5152325
- Chen, I. Y. H.; MacDonald, B. & Wünsche, B. (2008). Markerless Augmented Reality for Robots in Unprepared Environments. *ACRA 2008: Australasian Conference on Robotics and Automation*. Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Markerless+Augmented+Reality+for+Robots+in+Unprepared+Environments#1>
- Cheng, K.-H. & Tsai, C.-C. (2012). Affordances of Augmented Reality in Science Learning: Suggestions for Future Research. *Journal of Science Education and Technology*, 449–462. doi:10.1007/s10956-012-9405-9
- Collett, T. H. J. & MacDonald, B. A. (2006). Developer Oriented Visualisation of a Robot Program An Augmented Reality Approach. *HRI '06 Proceedings of the 1st ACM SIGCHI/SIGART. Conference on Human-Robot Interaction*, 49–56.
- Dominguez, C.; Essabbah, M.; Cheaib, N.; Otmane, S. & Dinis, A. (2012). Human-Robot-Interfaces based on Mixed Reality for Underwater Robot Teleoperation, *262160(Mcmc)*, 1–4.
- Feiner, S.; Macintyre, B.; Höllerer, T. & Webster, A. (1997). A Touring Machine : Prototyping 3D. *Mobile Augmented Reality Systems for Exploring the Urban Environment*, 97, 74–81.
- Fetić, A. & Jurić, D. (2012). The procedure of a camera calibration using Camera Calibration Toolbox for MATLAB. *MIPRO, 2012 Proceedings of the 35th International Convention*, 1752–1757. doi:978-1-4673-2577-6

- Garrido, R. & García-Alonso, A. (2007). Técnicas de interacción para Sistemas de Realidad Aumentada. In *II Jornadas sobre Realidad Virtual y Entornos Virtuales*. Albacete. Retrieved from <<http://www.sc.ehu.es/ccwgamoa/pub/aper0/AP-RealidadVir->
- González Morcillo, C.; López López, J. C. & Jiménez Linares, L. (2009). *Síntesis de imágenes digitales 3D*.
- González, R.; Rodríguez, F.; Sánchez, J.; López, H. & García, J. (2006). Algoritmo de navegación reactiva de robots móviles para tareas bajo invernadero. *XXVII Jornadas de Automatica*, 888–895.
- Green, S. A.; Chase, J. G.; Chen, X. & Billingham, M. (2008). Evaluating the Augmented Reality Human-Robot collaboration system. *15th International Conference on Mechatronics and Machine Vision in Practice, M2VIP'08*, 521–526.
- Henrysson, A.; Marshall, J. & Billingham, M. (2007). Experiments in 3D interaction for mobile phone AR. *Graphics and Interactive*, 1(212), 187–194. doi:10.1145/1321261.1321295
- Hernández Vázquez, A. (2012). *Herramientas ARToolKit implementadas en el proyecto de Realidad Aumentada TApir para sistemas operativos de código abierto*.
- Hoff, W. A.; Nguyen, K. & Lyon, T. (1996). Computer-vision-based registration techniques for augmented reality, *2904*, 538–548. doi:10.1117/12.256311
- Jara, C.; Candelas, P.; Fernández, M. & Torres, F. (2009). *An augmented reality interface for training robotics through the web. Communication*. Barcelona: 40th International Symposium on Robotics. Retrieved from <http://rua.ua.es/dspace/handle/10045/10163>
- Jaramillo Rojas, G. E. (2011). *Corrección del error en el proceso de registro en los sistemas de Realidad Aumentada utilizando técnicas heurísticas*.

- Jiménez, F. J.; Moreno, J. C.; Gonzalez, R. & Sánchez-Hermosilla, J. (2008). Sistema de visión de apoyo a la navegación de un robot móvil en invernaderos. In *CEA Jornadas de Automática* (pp. 10–17). Tarragona: CEA Jornadas de Automática,.
- Kim, Y. G.; An, J. & Lee, K.-D. (2011). Localization of Mobile Robot Based on Fusion of Artificial Landmark and RF TDOA Distance under Indoor Sensor Network. *International Journal of Advanced Robotic Systems*, 8(4), 1. doi:10.5772/45698
- Koehler, J.; Pagani, A. & Stricker, D. (2010). Robust Detection and Identification of Partially Occluded Circular Markers. *International Conference on Computer Vision Theory and Applications (VISAPP), 1*, 387–392.
- Köhler, J.; Pagani, A. & Stricker, D. (2011). Detection and Identification Techniques for Markers Used in Computer Vision. *Visualization of Large and Unstructured Data Sets - Applications in Geospatial Planning, Modeling and Engineering (IRTG 1131 Workshop)*, 36–44. doi:10.4230/OASlcs.VLUDS.2010.36
- Kozlov, A.; Macdonald, B. & Wünsche, B. (2007). Towards Improving SLAM Algorithm Development using Augmented Reality. *Proceedings of Australasian Conference on Robotics and Automation (ACRA) 2007*, 7–17.
- Krevelen, V. & Poelman, R. (2010). A Survey of Augmented Reality Technologies , Applications and Limitations. *The International Journal of Virtual Reality*, 9(2), 1–20.
- Lee, W. & Woo, W. (2008). Exploiting Context-Awareness in Augmented Reality Applications. *International Symposium on Ubiquitous Virtual Reality*, 51–54. doi:10.1109/ISUVR.2008.14
- López Pombo, H. (2010). *Análisis y desarrollo de sistemas de Realidad Aumentada*. Universidad Complutense de Madrid. Retrieved from <http://eprints.ucm.es/11425/>

- Luo, R. C. & Lai, C. C. (2014). Multisensor Fusion-Based Concurrent Environment Mapping and Moving Object Detection for Intelligent Service Robotics. *IEEE Transactions on Industrial Electronics*, 61(8), 4043–4051. doi:10.1109/TIE.2013.2288199
- Martínez-Gómez; J., Fernández-Caballero, A.; García-Varea, I.; Rodríguez, L. & Romero-González, C. (2014). A taxonomy of vision systems for ground mobile robots. *International Journal of Advanced Robotic Systems*, 11(1). doi:10.5772/58900
- Mejía Bustamante, J. & López Varona, R. (2014). Calibración de Cámara Termográfica Fluke TI-32 Thermographic camera calibration FLUKE TI-32. *Scientia et Technica, Año XIX, 19(1)*, 59–66. doi:ISSN 0122-1701
- Mekni, M. & Lemieux, A. (2014). Augmented Reality : Applications , Challenges and Future Trends. *Applied Computational Science Anywhere*, 205–214.
- Mendoza Marmolejo, L. & Henao Franco, N. I. (2012). *Aplicación de la Realidad Aumentada en Videojuegos*.
- Muñoz, N. D.; Ortiz, L. F. & Londoño, N. de J. (2007). Método de comparación de algoritmos de control de robots móviles. *Politécnica*, 5, 41–47.
- Novak-marcincin, J.; Janak, M.; Barna, J. & Novakova, L. (n.d.). Visualization of Robot Workcell Activity by Augmented Reality Technology Application 2 Development of Augmented Reality, 13–18.
- Ollero Baturone, A. (2005). *Robótica manipuladores y robots móviles* (Boixareu Editore, Ed.). (1st ed.). Barcelona: Marcombo.
- Osuna Altamirano, T. (2010). *Navegación y control de robot móvil*. Instituto Politécnico Nacional Centro.

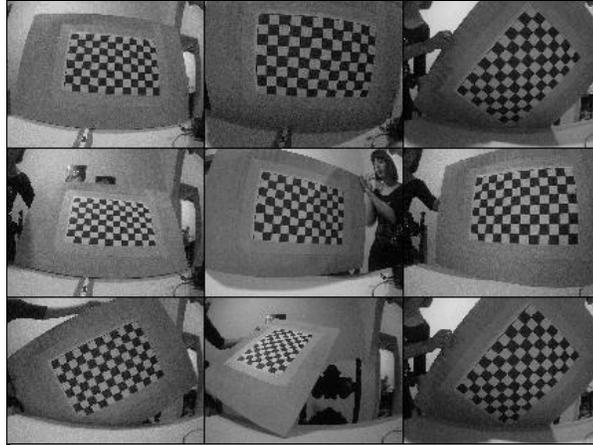
- Pérez, M. A.; Cuevas, E. V. & Navarro, D. (2009). Visión artificial aplicada a vehículos autónomos ciencia. *La Gaceta*, 15.
- Peters, T. M.; Linte, C. A.; Moore, J.; Bainbridge, D.; Jones, D. L. & Guiraudon, G. M. (2008). Towards a Medical Virtual Reality Environment for Minimally Invasive Cardiac Surgery. *Medical Imaging and Augmented Reality*, 5128, 1–11. doi:10.1007/978-3-540-79982-5_1
- Piekarski, W. & Thomas, B. H. (2003). Augmented reality user interfaces and techniques for outdoor modelling. *Proceedings of the 2003 Symposium on Interactive 3D Graphics - SI3D '03*, (C), 225. doi:10.1145/641522.641526
- Portilla, H. & Basañez, L. (2007). AUGMENTED REALITY TOOLS FOR ENHANCED ROBOTICS TELEOPERATION SYSTEMS. *3DTV Conference, 2007*, 1–4. doi:10.1109/3DTV.2007.4379424
- Portugal Zambrano, C. & Mena-Chalco, J. (2009). *Estimación de distancias absolutas utilizando un puntero láser y una cámara web Christian. III Simposio peruano de Computaci Ón Gráfica y Procesamiento de Imágenes*. Universidad Nacional de San Agustín.
- Rodríguez Lomusio, J. P. (2011). *Realidad Aumentada para el aprendizaje de ciencias en niños de educación general básica*. Universidad de Chile.
- Rohs, M. (2007). Marker-Based Embodied Interaction for Handheld Augmented Reality Games. *Virtual Reality*, 4(5).
- Ross, L. & Bradshaw, K. (2012). Fiducial Marker Navigation for Mobile Robots (November).
- Sanahuja, G.; Valera, A.; Sánchez, A. J.; Ricolfe-Viala, C.; Vallés, M. & Marín, L. (2011). Control embebido de robots móviles con recursos limitados basado en flujo óptico. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 8(3), 250–257. doi:10.1016/j.riai.2011.06.012

- Schulze-Ferebee, J. (2008). KUKA Roboter enhances perception of reality in robotics. *Kuka Roboter GmbH, 1*, 1–3. Retrieved from http://www.kuka-robotics.com/germany/en/pressevents/news/NN_060313_KUKA_Roboter_erweitert_Realit%C3%A4tswahrnehmung_in_der_Robotik.htm
- Soria, C. & Carellli, R. (2006). Control de un robot móvil utilizando el flujo óptico obtenido través de un sistema omnidireccional catadióptrico. In *IV Jornadas Argentinas de Robótica* (pp. 25–36).
- State, A.; Hirota, G.; Chen, D. T.; Garrett, W. F. & Livingston, M. A. (1996). Superior augmented reality registration by integrating landmark tracking and magnetic tracking. *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH, 96*, 429–438. doi:10.1145/237170.237282
- Tateno, K.; Kitahara, I. & Ohta, Y. (2007). A Nested Marker for Augmented Reality. *IEEE Virtual Reality Conference*, 259–262. doi:10.1109/VR.2007.352495
- Wang, K.; Liu, Y. & Li, L. (2014). A Simple and Parallel Algorithm for Real-Time Robot Localization by Fusing Monocular. *IEEE/ASME Transactions on Mechatronics, 19*(4), 1447–1457.
- Wang, X. & Kwok, N. (2007). Inspecting Interior Built Environments Using Augmented Reality-Based Autonomous Robot. *24th International Symposium on Automation & Robotics in Construction (ISARC 2007)*, 105–109.
- Wen, R.; Tay, W.-L.; Nguyen, B. P.; Chng, C. B. & Chui, C.-K. (2014). Hand gesture guided robot-assisted surgery based on a direct augmented reality interface. *Computer Methods and Programs in Biomedicine, 116*(2), 68–80. doi:10.1016/j.cmpb.2013.12.018

- White, S.; Feiner, S. & Koplec, J. (2006). Virtual Vouchers: Prototyping a Mobile Augmented Reality User Interface for Botanical Species Identification. *IEEE Symposium on 3D User Interfaces, 2006. 3DUI 2006.*, 119–126. doi:10.1109/TRIDUI.2006.1618281
- Wu, J.R.; Wang, M. L.; Liu, K. C.; Hu, M.H. & Lee, P.-Y. (2014). Real-time advanced spinal surgery via visible patient model and augmented reality system. *Computer Methods and Programs in Biomedicine*, 113(3), 869–81. doi:10.1016/j.cmpb.2013.12.021
- Yu, D.; Jin, J. S.; Luo, S. & Lai, W. (2010). A Useful Visualization Technique: A Literature Review for Augmented Reality and its Application, limitation & future direction Donggang. *Visual Information Communication*, (October 1998), 311–337. doi:10.1007/978-1-4419-0312-9
- Zhang, Y. (2014). *Real-time SLAM for Humanoid Robot Navigation Using Augmented Reality* by. Simon Fraser University.

ANEXOS

Anexo 1. Imágenes de calibración de cámara aérea



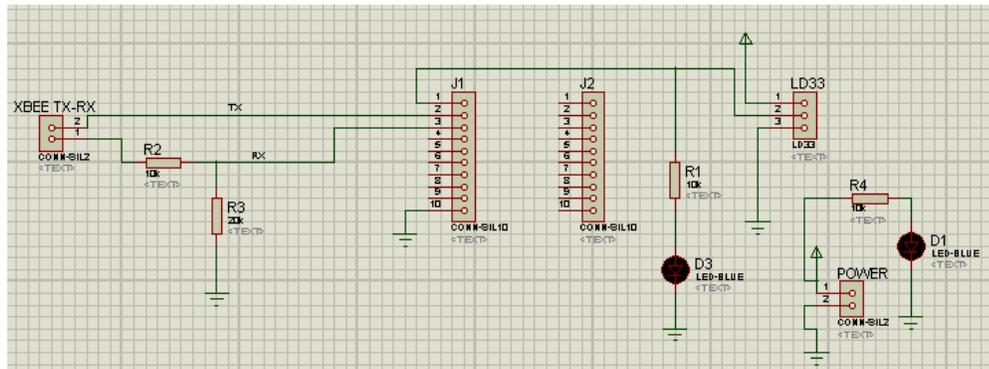
Fuente: autor

Anexo 2. Tabla de parámetros intrínsecos cámara aérea

Distancia focal	$fc = [972.37963 \ 957.39147] \pm [18.31176 \ 18.42006]$
Puntos principales	$cc = [637.00685 \ 451.68720] \pm [18.78591 \ 14.34728]$
Coefficiente de asimetría	$\alpha_c = [0.00000] \pm [0.00000] \Rightarrow$ angle of pixel axes = 90.00000 ± 0.00000 degrees
Distorsión	$kc = [-0.45711 \ 0.23524 \ 0.00017 \ 0.00255 \ 0.00000] \pm [0.03109 \ 0.06682 \ 0.00348 \ 0.00409 \ 0.00000]$
Error de pixel	$err = [1.50316 \ 1.20157]$

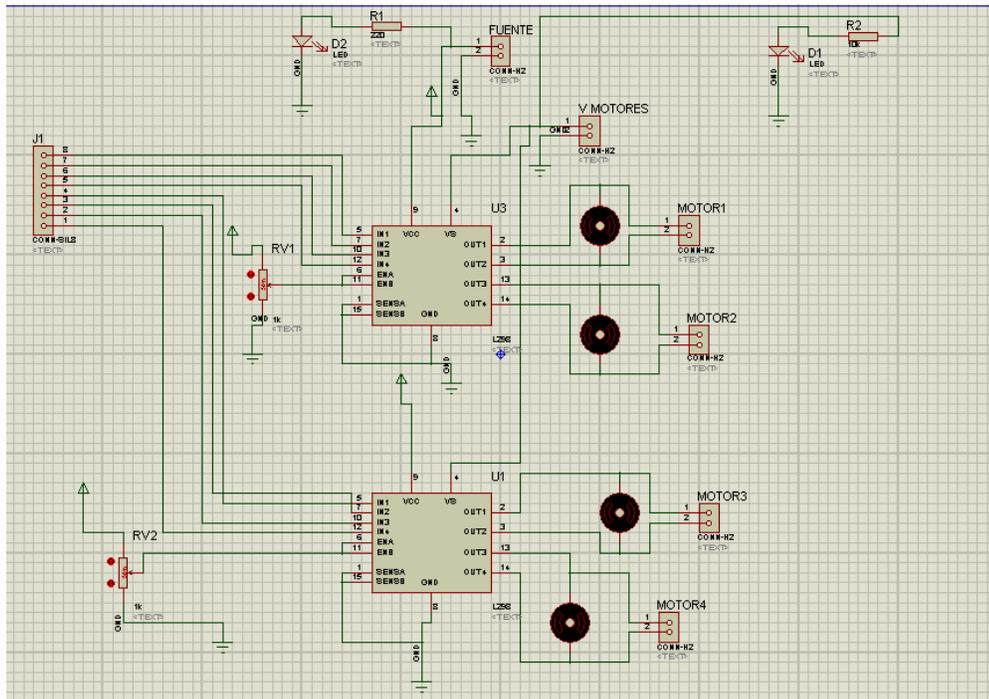
Fuente: Autor

Anexo 3. Plano esquemático de circuito de comunicaciones del robot



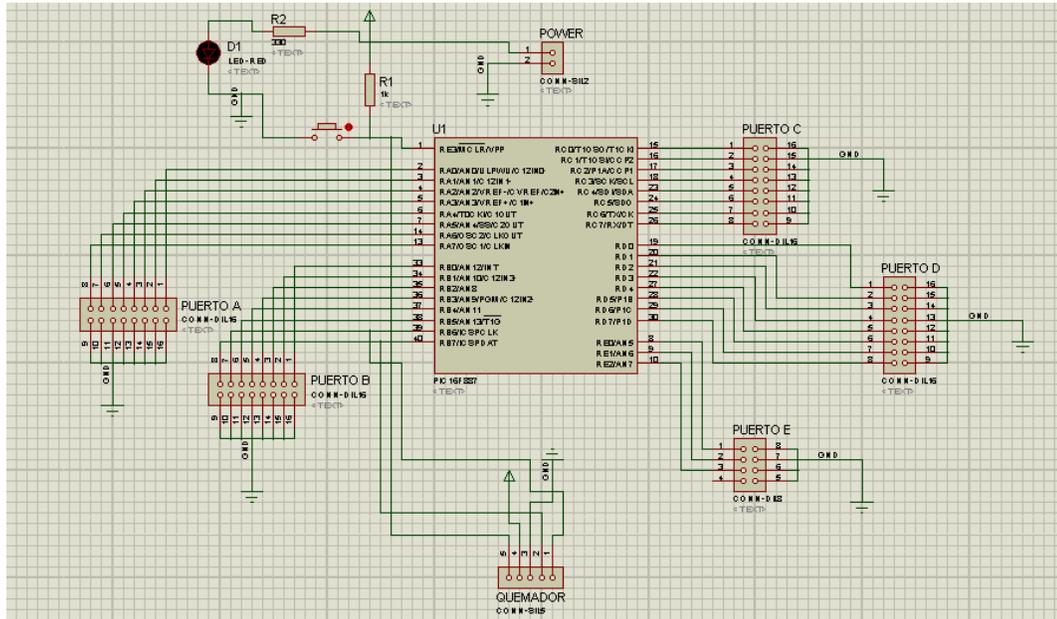
Fuente: Autor

Anexo 4. Plano esquemático de sistema de control de motor del robot



Fuente: Autor

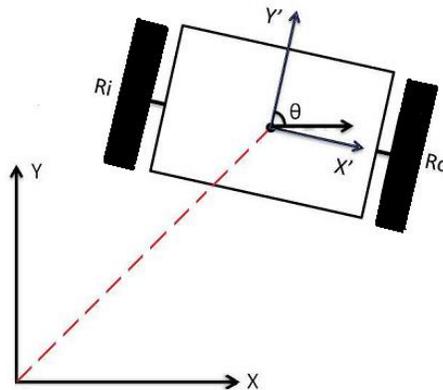
Anexo 5. Plano esquemático de sistema de control del robot



Fuente: Autor

Anexo 6. Modelo cinemático del robot

Un vehículo con orugas a baja velocidad puede ser modelado como un vehículo de configuración diferencial, agregando el deslizamiento de cada oruga (González, Rodríguez, & Guzmán, 2015).



Fuente: Arias, 2009

$$X' = \frac{V_r(t)(1 i_r(t)) + V_l(t)(1 i_l(t))}{2} \cos t$$

$$Y' = \frac{V_r(t)(1 i_r(t)) + V_l(t)(1 i_l(t))}{2} \sin t$$

$$\theta = \frac{V_r(t)(1 i_r(t)) + V_l(t)(1 i_l(t))}{b}$$

(X', Y', θ) : Posición del robot

$V_r V_l$: Velocidad de cada oruga

$i_r i_l$: Deslizamiento de cada oruga

b : Ancho del robot

Anexo 7. Código de algoritmo de extracción de fotogramas de videos

```
% muestreo de video pantalla completa
pause(10)
i=0;
for tt=1:1:3
%xxxxxxxxxxSegmento que capturar el escritorioxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
%pause(5)
robo = java.awt.Robot;
t = java.awt.Toolkit.getDefaultToolkit();
rectangle = java.awt.Rectangle(t.getScreenSize());
image = robo.createScreenCapture(rectangle);
if(tt<10)
filehandle = java.io.File(sprintf('capture0%d.jpg',tt));
javax.imageio.ImageIO.write(image,'jpg',filehandle);
else
filehandle = java.io.File(sprintf('capture%d.jpg',tt));
javax.imageio.ImageIO.write(image,'jpg',filehandle);
end
```

```

%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
pause(0.4)%0.4
end

```

Anexo 8. Código de algoritmo de identificación del robot y trazado de trayectorias

```

% imagen inicial de plantilla
clear
L = imread('capture01.jpg');
grises=rgb2gray(L);% convierte la imagen a escala de grises
humbral=0.34;
bw2=im2bw(grises,humbral);
%XXXXXXXXXXXXXXXXXXXXXXXXXXXX Lectura secuencial de imagenes XXXXXXXXXXXXXXXXXXXXXXX

lee_archivos = dir('C:\Users\LENOVO\Documents\MATLAB\imagenes\*.jpg'); %el formato de imagen
puede ser modificado.
for k = 1:length(lee_archivos) %recorre número de archivos guardados en el directorio
archivo = lee_archivos(k).name; %Obtiene el nombre de los archivos
nombre='C:\Users\LENOVO\Documents\MATLAB\imagenes\'; %Recorre el directorio
I = imread(strcat(nombre,archivo));% lee imagen
%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
grises=rgb2gray(I);% convierte la imagen a escala de grises
humbral=0.18;%38
bw=im2bw(grises,humbral);% convierte la imagen a binaria pura con el humbral calculado
% bw2=bw2&bw;% sobreescritura de imahener
%figure(4)
%imshow(bw2)
%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
% filtro=ones(178,1366);%ENMARCADO INFERIO
for tt=600:768

```



```

L = imread('capture.jpg');
grises=rgb2gray(L);%convierte la imagen a escala de grises
humbral=0.20;
bw=im2bw(grises,humbral);
%lllllllllleer trayectoriasxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
% fffffffffffffFILTRO O ENMARCADOffffff
% filtro=ones(178,1366);%ENMARCADO INFERIO
for tt=600:768
for jj=1:1366
bw(tt,jj)=1;
end
end
for tt=1:80 %ENMARCADO SUPERIOR
for jj=1:1366
bw(tt,jj)=1;
end
end
% fffffffffffff
figure(7)
imshow(bw)
hold on

% if (kk~=12)&(kk~=13)&(kk~=14)%eliminar trallectorias

%pause;
%ccccccccccccccccccimprimir datos reales

for kkk=1:largo
% plot(matbv(kkk,kk*2+1),matbv(kkk,kk*2+2),'r*')
%xxxxxxCrecion de recuadro en torno al centroidexxxxxxxxxxxxxxxxx
xx=(centroides(kkk,kk*2+1)-5);

```

```

yy=(centroides(kkk,kk*2+2)-30);
% rectangle('Position',[xx yy 100 60],'EdgeColor','r')
rectangle('Position',[xx (yy-30) 4 4],'EdgeColor','r')%rectangle('Position',[xx yy 100 60],'EdgeColor','r') x
y coordenadas de recuadro
% pause; % w=100 ancro del recuadro h=60 alto del recuadro
rectangle('Position',[xx (yy+70) 4 4],'EdgeColor','r')%rectangle('Position',[xx yy 100 60],'EdgeColor','r') x
y coordenadas de recuadro
%pause; % w=100 ancro del recuadro h=60 alto del recuadro

%imdistline
%end
end
imdistline
pause
end

```

Anexo 10. Código algoritmo de campos potenciales

```

pause(5)
band='A';
%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%Programa de ensayo de movimiento de los motores servos HS-311
%ABRIR el puerto COM23
%clc;
disp('BEGIN')
SerPIC = serial('COM14');
set(SerPIC,'BaudRate',9600);
set(SerPIC,'DataBits',8);
set(SerPIC,'Parity','none');

```



```

end
angulo= ((atan(yy/xx))*180)/pi; %calculo del ángulo en grados
magnitud=sqrt((xx*xx)+(yy*yy)); %magnitud del vector de obstaculos
tiempo=3-(magnitud/10000); %escalizacion en tiempo de giro
xx=0;
yy=0;
%dddddddddddddddddddddddddddddddddddddddddddddddddddd
%DIRECCON DEL ROBOT

if(angulo>=0)
band='D'
end
if(angulo<0)
band='C'
end

fprintf(SerPIC,band);
pause(tiempo+0.4)
band='A';
fprintf(SerPIC,band);
pause(0.5)
band='I';
fprintf(SerPIC,band);

tt=tt

end
%CERRAR el puerto COM1 al finalizar
band='I' %robot stop
fprintf(SerPIC,band);

```

```
fclose(SerPIC);
delete(SerPIC)
clear SerPIC
```

```
disp('STOP')
```

```
%XXXXXXXXXX
```

Anexo 11. Código para el análisis morfológico de marcadores

```
%% Leer imagen
img=imread('barras.png');
%% Convertir a binaria
bn=~im2bw(img);
%% Mostrar imagen
figure(1)
imshow(bn)
%% Eliminar áreas menores a 100 pixels
bn=bwareaopen(bn,100);
imshow(bn)
%% Contar elementos y etiquetarlos
[L Ne]=bwlabel(bn);
imshow(label2rgb(L));
%% Encontrar propiedades de la imagen
prop=regionprops(L);
s = regionprops(L,
'area','BoundingBox','Centroid','ConvexArea','ConvexHull','ConvexImage','Eccentricity','EquivDiameter','
EulerNumber','Extent','Extrema','FilledArea','FilledImage','Image','MajorAxisLength','MinorAxisLength','
Orientation','Perimeter','PixelIdxList','PixelList','Solidity','SubarrayIdx');
```

```

% sustraigo el vector de areas de las regiones
areatotal=0;
areas=[prop.Area];
%% Gráficar rectángulos en las formas Y los centroides de cada elemento
hold on
for n=1:length(prop)
    rectangle('Position',prop(n).BoundingBox,'EdgeColor','g','LineWidth',2)
    x=prop(n).Centroid(1);
    y=prop(n).Centroid(2);
    % obtengo el area total de los elementos encontrados
    areatotal=areatotal+area(n)
    plot(x,y,'*')
    % text(x,y,'O')
end
hold off

%% Gráficar perímetro
imshow(bn)
%%
hold on
B=bwboundaries(bn);
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'r', 'LineWidth', 2)
end
hold off
%% Seleccionar área más grande
cla
areas=[prop.Area];
[area_max pam]=max(areas);

```

```

am=ismember(L,pam);
imshow(am)

%FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
centroids = cat(1, s.Centroid);
figure(2)
imshow(L)
hold on
imshow(label2rgb(L));
hold on
plot(centroids(:,1), centroids(:,2), 'b*')

areas=cat(1, s.Area)
%'ConvexArea' Escalar que especifica el número de píxeles en 'ConvexImage'.
%'ConvexHull'-matriz que especifica el polígono convexo más pequeño que puede
%contener la región. Cada fila de la matriz contiene el x-e y-coordenadas de un vértice del polígono.
areaconvexa=cat(1, s.ConvexArea);
convexohull=cat(1, s.ConvexHull);
%numero de puntos de areaminima poligonal convexa
numconvexhull=length(convexohull);
figure(3)
imshow(L)
hold on
plot( convexohull(:,1), convexohull(:,2), 'b.')
%'Solidity' - Escalar especificando el porcentaje de los píxeles en el casco
%convexo que también están en la región. Calculado como Area / ConvexArea.
solides=cat(1, s.Solidity);

```

Anexo 12. Código algoritmo reactivo

```

disp('BEGIN')
SerPIC = serial('COM14');
set(SerPIC,'BaudRate',9600);
set(SerPIC,'DataBits',8);
set(SerPIC,'Parity','none');
set(SerPIC,'StopBits',1);
set(SerPIC,'FlowControl','none');
fopen(SerPIC);
% while true
giro=0 % variable que cuenta los instantes sin recorrido a adelante
pause(5)
for tt=0:1:300
%xxxxxxxxxxxsegmento que captur el escritorioxxxxxxxxxxxxxxxxxxxxxxxxxxxx
%pause(5)
robo = java.awt.Robot;
t = java.awt.Toolkit.getDefaultToolkit(); %# Set the capture area as the size for the screen
rectangle = java.awt.Rectangle(t.getScreenSize()); %# Get the capture
image = robo.createScreenCapture(rectangle); %# Save it to file
filehandle = java.io.File(sprintf('capture%d.jpg', 0));
javax.imageio.ImageIO.write(image,'jpg',filehandle);
%xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
lado=2;
img=imread('capture0.jpg');
%figure(1)
% imshow(img)
%title('IMAGEN ORIGINAL')
grises=rgb2gray(img);%convierte la imagen a escala de grises
%figure(2)
% imshow(grises)
% humbral=graythresh(grises); %calcula humbral para convertir la imagen abinaria pura
% for n=0:0.1:1

```



```

band='D';
end
fprintf(SerPIC,band);pause(2.5)
giro=0;
end

end

fprintf(SerPIC,band);pause(0.2)
%*-*-*FIN Posición final

pause(0.5)
tt=tt
end
%CERRAR el puerto COM1 al finalizar
band='I'
fprintf(SerPIC,band);pause(0.2)
fclose(SerPIC);
delete(SerPIC)
clear SerPIC

disp('STOP')
%XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
%VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
%pause(300) %# Wait for 5 min
%i = i + 1; end

```