



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

Facultad de Ingeniería

SISTEMA DE VISIÓN ICONOGRÁFICO
PARA LOCALIZACIÓN DE OBJETOS
EN AMBIENTES CERRADOS

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

P R E S E N T A N :

BARRAGÁN CORTÉS SERGIO
SÁNCHEZ MARTÍNEZ NAZARETH



DIRECTOR: DR. MARIO PEÑA CABRERA

Ciudad Universitaria

México, D.F. 2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México que a través de su Facultad de Ingeniería nos proporcionó una excelente formación profesional.

Al Departamento de Ingeniería en Sistemas Computacionales y Automatización, sección Electrónica y Automatización del Instituto de Investigación en Matemáticas Aplicadas y en Sistemas, por proporcionarnos los medios necesarios para la realización de este proyecto.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT), por el apoyo recibido para la realización de esta tesis.

A los profesores de la facultad que nos compartieron sus conocimientos, anécdotas y experiencias.

A nuestro asesor Mario Peña por el apoyo y consejos que nos dio durante la elaboración de esta tesis y adentrarnos en el mundo de la investigación.

A Román Osorio por darnos la oportunidad de ingresar al IIMAS y apoyarnos durante nuestra estancia en el instituto.

*Sergio
Nazareth*

A mis padres, Jesús Barragán y Ma. de Jesús Cortés

Por que gracias a su amor, ejemplo, apoyo y sacrificio hoy comparto con ustedes la alegría de terminar este proyecto.

A mis hermanos Jesús, Armando y Lourdes

Por el cariño que me han dado siempre, por el apoyo que me dieron durante mi trayectoria escolar.

A Maritza

Por tu amor y paciencia infinitos, por estar a mi lado y apoyarme durante la conclusión de este proyecto. Te amo

A familiares y amigos

Por ser parte de mi vida y compartir incontables momentos donde hemos reído y llorado juntos.

A “los de la piedra”

Por los increíbles momentos que pasamos juntos, por su apoyo y por enseñarme que la facultad de ingeniería no es solo un sitio donde estudiar.

Sergio

A MI FAMILIA

Por su apoyo, paciencia y esperanzas puestas en mi, pero sobre todo porque de una u otra forma han colaborado en la culminación de una etapa muy importante en mi vida.

A MI NOVIA MARY

Por llegar a mi vida, brindarme tu amor incondicional, tu comprensión, tu apoyo y por permanecer a mi lado todo este tiempo, te amo.

A MIS AMIGOS Y AMIGAS

Porque gracias a su amistad, su compañía y sus consejos lograron convertir el duro trabajo de comenzar y terminar la carrera en una simple tarea.

A CHARLY

Por tenerme la confianza y permitirme entrar al selecto grupo de Infraestructura de la DGSCA, logrando con ello aprender de ti muchas cosas que me serán de gran utilidad en mi carrera profesional.

A LAURA

Por recibirme siempre con los brazos abiertos, por todo su apoyo y por todas las facilidades que me brindo cuando estuve en la DGSCA.

Nazareth

No basta con alcanzar la sabiduría, es necesario saber utilizarla.
Marco Tulio Cicerón

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. ANTECEDENTES	5
Introducción.....	7
1.1 Visión Humana	7
1.1.1 El ojo	7
1.2 Visión Artificial.....	10
1.2.1 Sistema de Visión Artificial.....	10
1.2.2 Comparación de la visión artificial y la visión humana.....	11
1.3 Procesamiento digital de imágenes.....	12
1.3.1 Representación de una Imagen digital	12
1.3.2 Histograma de una imagen	15
1.3.3 Adquisición de imágenes	16
1.3.4 Mejoramiento.....	20
1.3.5 Restauración	23
1.4 Reconocimiento de Patrones.....	37
1.5 Software.....	41
1.5.1 Visual Basic.....	41
1.5.2 Matlab.....	41
1.6 Hardware	42
1.6.1 Cámara.....	42
1.6.2 Tarjeta de red	43
1.6.3 Computadora	44
CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA.....	45
Introducción.....	47
2.1 Objetivo del proyecto	47
2.2 Definición del problema	47
2.3 Análisis del problema	49
2.4 Propuesta de solución.....	50
2.4.1 Selección de la cámara.....	51
2.4.2 Diseño de la base iconográfica.	51
2.4.3 Especificaciones para el diseño de la maqueta.	52
2.4.4 Especificaciones para la programación.....	54

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN	57
Introducción.....	59
3.1 Software.....	59
3.1.1 Modelado del sistema	59
3.1.2 Detección del icono	62
3.1.3 Identificación del icono	64
3.1.4 Cálculo de la distancia al icono	65
3.1.5 Cálculo del ángulo de visión.....	68
3.1.6 Obtención de la posición de la cámara	76
3.2 Implementación	78
3.2.1 Descripción gráfica del sistema	78
3.2.2 Interfaz de inicio	79
3.2.3 Interfaz de acceso	80
3.2.4 Interfaz principal.....	80
3.2.5 Interfaz Calibración Límites	82
3.2.6 Interfaz Captura Parámetros	83
3.2.7 Interfaz Coordenadas Iconos	83
3.2.8 Interfaces Entrenamiento	84
3.2.9 Interfaz del Tutorial	85
3.2.10 Archivo de Ayuda.....	85
3.2.11 Pantalla Propiedades	86
3.2.12 Pantalla Acerca de	86
3.3 Hardware	87
3.3.1 Diseño del controlador.....	87
3.3.2 Implementación del controlador	89
CAPÍTULO 4. RESULTADOS EXPERIMENTALES	91
Introducción.....	93
4.1 Pruebas de precisión.....	93
4.2 Funcionamiento	100
CAPÍTULO 5. CONCLUSIONES.....	101
Introducción.....	103
5.1 Con respecto a las características de construcción del sistema.	103
5.2 Con respecto al software desarrollado.....	103
5.3 Con respecto al funcionamiento del sistema.....	104
5.4 Trabajo futuro	105

Apéndice A. MATLAB Automation Server.....	107
Apéndice B. Curva de ajuste y regresión.....	115
Apéndice C. Cálculo de errores.....	121
Apéndice D. Circuito Temporizador 555 Astable.....	125
Apéndice E. Artículos consultados.....	129
Apéndice F. Hojas de especificaciones.....	148
Referencias.....	159

Introducción

Durante años el ser humano se ha preocupado por preservar la raza humana, como prueba de esto, se encuentran diseños y desarrollos de un gran número de sistemas y dispositivos que de alguna manera intentan sustituir o imitar el comportamiento de algún miembro o modo sensorial del ser humano.

De los sistemas desarrollados son de nuestro interés los relacionados con la visión humana, debido a que este sentido aunque no indispensable para sobrevivir es el mas utilizado para tener un mejor desenvolvimiento dentro de la sociedad; tal es el caso de las personas invidentes, a quienes de alguna manera se les discrimina, y es aquí, donde los sistemas de ayuda a minusvalidos realizan una aportación importante. Es nuestro deseo, que el sistema que se presenta en esta tesis sirva de ayuda a técnicos especializados para elaborar algún sistema que con base en el que desarrollamos, pudiesen llegar a una aplicación industrial y comercial mas robusta, que sea de gran ayuda para las personas invidentes.

El sistema desarrollado, pretende obtener en tiempo real el posicionamiento de un usuario móvil dentro de un espacio de trabajo cerrado, por medio de la interpretación de iconos presentes en imágenes adquiridas con cámaras de video.

Sin embargo, el objetivo principal del trabajo no solo es el mencionado anteriormente, sino que tenga una aplicación directa en la automatización de procesos industriales, en donde es de vital importancia el uso de máquinas de control digital y robots manipuladores, los que en un proceso, en algún momento necesitan conocer su posición dentro del área de trabajo para desarrollar su labor eficientemente. Para un mejor entendimiento, se presentan y a manera de resumen, algunos trabajos realizados anteriormente y relacionados con el tema de desarrollo, que sirvieron de apoyo e inspiración para la realización de esta tesis.

1. Determinación de la posición de un objetivo por análisis de perspectiva, aplicada a un robot basketbolista. (Talledo J, Mayorga E, 2002) [b]

Esté trabajo, muestra un método para poder determinar y proporcionar la posición relativa de un tablero de baloncesto, a un robot logra realizar encestes desde una región valida bien definida. El método utiliza el análisis de la deformación por perspectiva que sufre la imagen de un objeto al ser proyectada sobre un plano.

Para poder cumplir con este objetivo, el robot a través de una cámara obtiene imágenes de una figura que se encuentra en la parte inferior del tablero, dicha figura consta de tres franjas, el procesamiento consiste en tomar las medidas de las alturas de las franjas y la distancia entre las franjas laterales (todo en píxeles), y basándose en leyes de la perspectiva se obtiene una ecuación que relaciona la altura en píxeles de la franja central con la distancia del robot a dicha franja.

De manera similar, se hace un análisis parecido para obtener el ángulo que se forma entre la perpendicular al tablero y el eje óptico de la cámara. Una vez obtenidos estos dos parámetros (distancia del robot al tablero y ángulo) se realizo un pequeño entrenamiento sobre la región, con el cual el robot logra encestar el balón desde cualquier posición que se encuentre dentro de esta.

2. Sistema de Localización y Posicionamiento de Piezas utilizando Visión Artificial. (de Lope J, Serradilla F, 1997) [d]

En este artículo los autores describen el diseño y la implementación de un sistema automatizado que por medio de un programa de cómputo que utiliza técnicas de visión artificial logran hacer el reconocimiento, localización y posicionamiento de piezas de fabricación para la industria aeronáutica dentro de un ambiente de trabajo específico.

Para la realización de este sistema se ocupan dos cámaras CCD conectadas a una tarjeta digitalizadora, un robot de tipo pórtico montado sobre la mesa de trabajo y una computadora para poder procesar las imágenes y realizar el control el robot.

El funcionamiento del sistema a grandes rasgos es el siguiente: una de las cámaras obtiene una toma lejana para obtener un panorama general de la zona de trabajo y la segunda se mueve en conjunto con el efector del robot. La cámara que realiza la toma cercana toma una imagen en 256 niveles de gris, la binariza y toma puntos pertenecientes a la pieza y al fondo, los cuales son etiquetados con diferentes valores. Mediante un procesamiento se identifica la pieza y se toma un vértice de referencia que es referenciada a la imagen panorámica y mediante un modelo geométrico se calcula la posición de la pieza y el movimiento necesario, información que se comunica al robot para depositar la pieza en su destino.

3. Sistema de Visión Espacial (SVS). (Alberta Online Encyclopedia, 2003) [c]

Al trabajar en el espacio, no sólo la falta de gravedad es un factor a considerar, sino que también lo son la oscuridad del espacio y la falta de puntos de referencia.

El sistema de visión espacial proporciona puntos de referencia de igual tamaño, colocados en lugares previamente establecidos de la nave y demás objetos a analizar; calcula la localización, orientación y velocidad a la cual se mueven los objetos con respecto a las referencias: Canadarm¹, Space Shuttle² y la estación espacial.

El sistema funciona mediante una cámara de televisión que adquiere y transmite imágenes de los objetos a 30 cuadros por segundo, se realiza el procesamiento y análisis de ellas antes de que la próxima imagen sea adquirida. Conforme el objeto se mueve, el sistema de visión calcula el cambio de posición de los puntos de referencia y nos da una representación gráfica en tiempo real del objeto, así como su localización y orientación. Esta información permite al operador un manejo más exacto de los objetos mediante el brazo Canadarm, permitiendo así construir o hacer reparaciones a satélites, estaciones espaciales y otros objetos.

De manera muy concreta, se presenta a continuación una descripción del contenido de cada capítulo, con la intención de dar al lector un panorama general de los temas tratados a lo largo de esta tesis.

¹ Brazo manipulador diseñado por los canadienses para trabajar en naves espaciales.

² Nave espacial lanzada por primera vez el 12 de abril de 1981.

En el capítulo 1 se muestran antecedentes de los sistemas de visión, conceptos básicos de procesamiento digital de imágenes y reconocimiento de patrones, así como algunas ideas que sirvieron de base para el diseño del sistema tratado a lo largo de este trabajo.

En el capítulo 2 se establece el objetivo principal del presente trabajo, así como los alcances. Posteriormente se realiza, de manera general, un análisis del problema planteado para plantear, por último, la propuesta de solución.

El capítulo 3 presenta una descripción detallada de las etapas de diseño e implementación del sistema, tanto para la parte de *software*, como para la parte de *hardware*.

El capítulo 4 muestra la etapa de realización de pruebas y análisis de resultados, con el fin de evaluar y entender el comportamiento del sistema desarrollado.

Por último se presentan en el capítulo 5, las conclusiones a las que se llegó al concluir este trabajo, acompañadas de una sección de opciones de trabajo futuro, que pueden ser tomadas en cuenta para futuras tesis.

Capítulo 1

Antecedentes

Introducción

El presente capítulo muestra un breviario acerca de la visión humana y de la visión artificial, por otra parte describe las técnicas más utilizadas de reconocimiento de patrones y procesamiento digital de imágenes, así como también, de manera muy concisa algunos conceptos utilizados para el desarrollo e implementación del sistema en el cual se centra este trabajo.

1.1 Visión Humana

En esas áreas del procesamiento digital de imágenes y el reconocimiento de patrones, es necesario entender el sistema visual humano ya que al comprender las características y las limitaciones de este, se logran dimensionar los diseños y mejorar la efectividad de los distintos algoritmos desarrollados para el tratamiento de imágenes.

El sistema de visión humano está compuesto por el ojo y una parte del cerebro, que procesa las señales neurológicas que provienen de este. El ojo y el cerebro en conjunto convierten la información óptica en una percepción de una escena visual. El ojo funciona como la cámara del sistema visual humano y su función es traducir las ondas electromagnéticas de la luz en un determinado tipo de impulsos nerviosos que son enviados al cerebro a través del nervio óptico para realizar un análisis de lo que se está viendo.

1.1.1 El ojo

El ojo en su conjunto, llamado globo ocular, es una estructura casi esférica de aproximadamente 25 [mm] de diámetro. Está rodeado por tres membranas: la córnea y la esclerótica, que constituyen la cubierta exterior, la coroides y la retina. Los rayos de luz generados o reflejados por un objeto primero golpean el ojo en la córnea. La córnea actúa como un lente convexo, refractando los rayos. Esta refracción forma el enfoque inicial de la luz que entra al ojo. La córnea forma una protección transparente que cubre la superficie anterior del ojo. Después de la córnea, los rayos pasan a través de un líquido claro y húmedo llamado el humor acuoso, y después pasan a través del iris y el cristalino. El iris actúa como una apertura variable que controla la cantidad de luz que puede pasar a través del cristalino. El iris es controlado por músculos que lo abren y lo cierran basados en la intensidad promedio del objeto que es observado.

El ojo (figura 1.1) es capaz de adaptarse a distintos niveles de iluminación gracias a que el diafragma formado por el iris puede cambiar de diámetro, proporcionando un agujero central (la pupila) que varía entre 2 [mm] (para iluminación intensa) y 8 [mm] (para situaciones de poca iluminación).

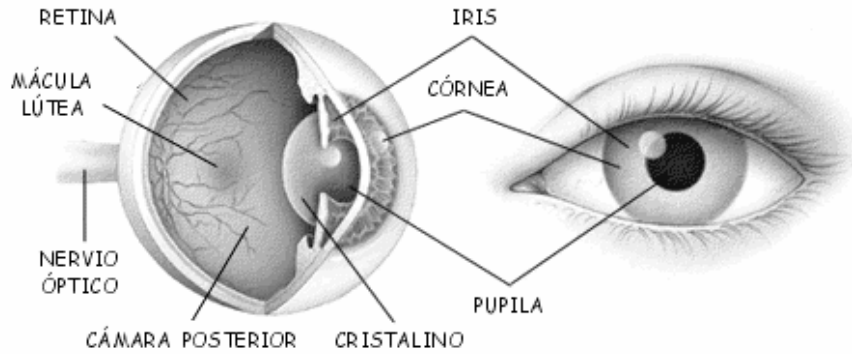


Figura 1.1. Esquema del ojo

El cristalino lleva a cabo el segundo enfoque de la luz, proyectando a esta en la retina. El cristalino es controlado por músculos, los cuales permiten variar la distancia focal del sistema óptico total dependiendo de la distancia del objeto observado. Tal como una cámara, el ojo debe ser enfocado basado en qué tan lejos está este del objeto. Los rayos de luz salen del cristalino pasando a través de una sustancia transparente y gelatinosa, llamada humor vítreo, y son finalmente enfocados en la retina. El humor vítreo mantiene la estructura del ojo mientras que ópticamente une el cristalino a la retina.



(a) Formación de la imagen en el ojo humano (b) Formación de la imagen en una cámara
Figura 1.2.

La membrana más interna del ojo es la retina, que cubre la totalidad de la pared posterior. Cuando el ojo está correctamente enfocado, la luz de un objeto exterior al ojo forma su imagen en la retina. La retina está compuesta por fotorreceptores que convierten la intensidad y el color de la luz en señales nerviosas. Existen dos tipos de fotorreceptores, bastones y conos. Los bastones son los más abundantes: entre 75 y 150 millones están distribuidos sobre la superficie retiniana y son los que más responden a la luz. Su gran área

de distribución, junto con el hecho de que grupos de varios bastones comparten una misma terminación nerviosa, reduce la cantidad de detalle discernible por estos receptores. Los bastones sirven para dar una visión general del campo de visión, no están implicados en la visión del color y son sensibles a niveles de iluminación bajos, tal como en la noche.

Los conos son mucho menos abundantes que los bastones (alrededor de 6 a 7 millones) y están localizados principalmente en la región central de la retina, denominada fovea. Los conos son muy sensibles al color y son algo menos sensitivos a la luz. Son usados para la visión de luz brillante, tal como en un día soleado. Los seres humanos pueden apreciar detalles relativamente finos gracias a esos conos porque cada uno está conectado a su propia terminación nerviosa. Los músculos que controlan el ojo giran el globo ocular hasta que la imagen del objeto visto queda en la fovea. Existen tres diferentes tipos de conos; cada uno responde a una banda distinta del espectro de la luz. Básicamente, cada cono responde de forma diferente a un color arbitrario, así genera un conjunto único de respuestas para cada color de la luz. Con estas señales de los tres tipos de conos, el cerebro tiene la información con la cual forma una percepción distinta de un gran número de colores diferentes.

Las diferencias entre bastones y conos, y sus distribuciones a través de la retina, son responsables de diversos aspectos de la visión. Ya que los conos que sensan el color están concentrados en la fovea, la percepción del color es mejor para los objetos que se ven directamente al frente. Recíprocamente, se tiene una mínima percepción del color para objetos en la visión periférica. Ya que los altamente sensitivos bastones son abundantes por todas partes menos en la fovea, la percepción de luz de bajo nivel es mejor en la visión periférica. Así, durante la noche, los objetos confusos se pueden ver por la parte periférica de la retina cuando son invisibles para la fovea. La relativa insensibilidad de los conos cuenta además para la incapacidad de percibir el color bajo condiciones de poca luz, tal como en la noche.

Como la luz incide en los bastones y los conos, esta causa una reacción electroquímica que genera impulsos nerviosos. Estos impulsos se pasan al cerebro por el nervio óptico. El nervio óptico es una extensión de la retina que lo conecta al cerebro. Un pequeño punto ciego se crea en la retina donde el nervio óptico se une. Los impulsos neuronales son recibidos por el cerebro y procesados por la corteza visual. La percepción de la visión es creada dentro del proceso de la corteza visual.

1.2 Visión Artificial

Para una máquina el proceso de la visión representa un conjunto de acciones complejas, en las cuales deben existir los elementos adecuados para percibir el medio escénico, compararlo con información preestablecida y realizar acciones según las indicaciones programadas, por lo tanto el término de visión artificial se puede entender como la integración de dispositivos para implementar máquinas con capacidad de percibir el medio ambiente escénico que los rodea, con la intención de que vean solo lo necesario para procesarlo (región de interés), analizarlo y tomar una decisión como salida.

Con el propósito de complementar lo anterior, se presenta a continuación una serie de definiciones dadas por reconocidos investigadores en el área:

- Desde una perspectiva general, la visión artificial por computador es la capacidad de la máquina para ver el mundo que lo rodea, más precisamente para deducir la estructura y propiedades del mundo tridimensional a partir de una o más imágenes bidimensionales (Pajares, 2002) [1].
- Ciencia que desarrolla las bases teóricas y algorítmicas para obtener información sobre el mundo real a partir de una o varias imágenes (Haralick, 1992) [2].
- El término “Visión por Computadora” dentro del campo de la Inteligencia Artificial puede considerarse como el conjunto de todas aquellas técnicas y modelos que nos permitan el procesamiento, análisis y explicación de cualquier tipo de información espacial obtenida a través de imágenes digitales (Posgrado, Ingeniería UNAM) [a].

1.2.1 Sistema de Visión Artificial

Un sistema de visión artificial, como se puede ver en la figura 1.3, esta compuesto por los siguientes elementos:

- Adquisición de la imagen: se obtiene la imagen adecuada del objeto de estudio. Dependiendo de la aplicación la imagen puede ser una fotografía, radiografía, termografía, etc.
- Preprocesamiento: con el fin de mejorar la calidad de la imagen obtenida se emplean ciertos filtros digitales o métodos de procesamiento que eliminan el ruido y mejoran la imagen como aumentar el contraste o mejorar los bordes.

- Segmentación: se identifica el objeto de estudio en la imagen.
- Medición (extracción de características): se realiza una medición paramétrica objetiva de ciertos atributos de interés del objeto de estudio.
- Interpretación (clasificación): de acuerdo a los valores obtenidos en las mediciones se lleva a cabo una interpretación del objeto.

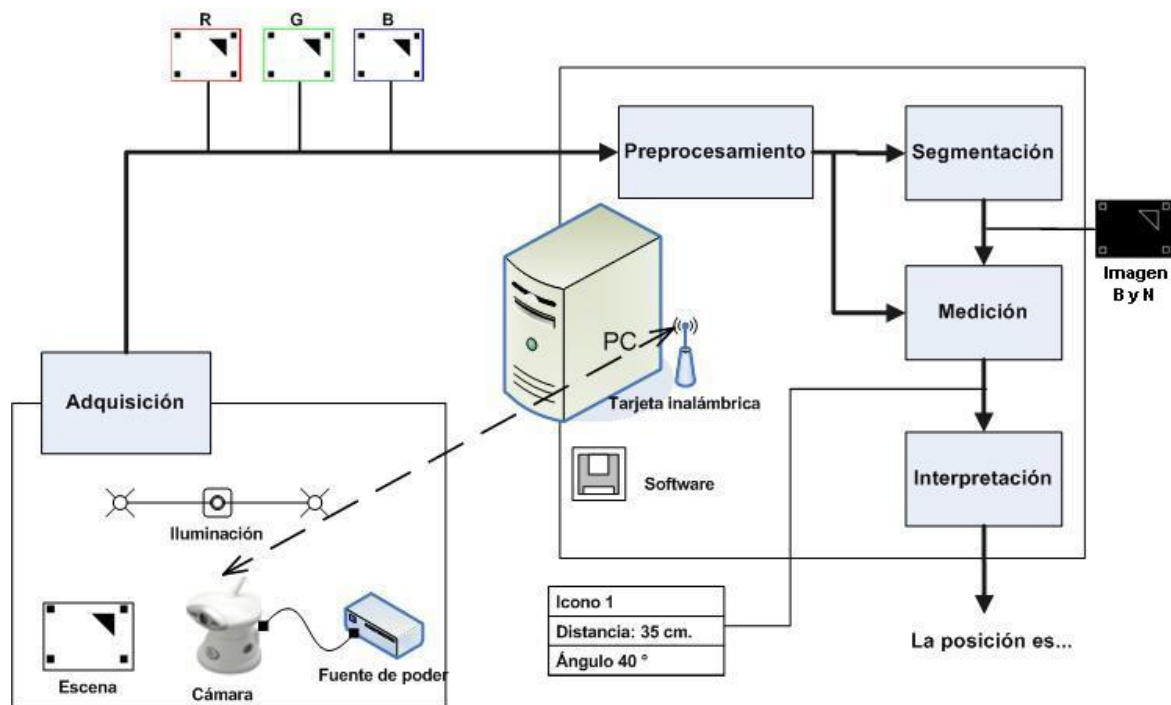


Figura 1.3. Sistema de visión artificial

1.2.2 Comparación de la visión artificial y la visión humana

Para darnos una idea de lo que implica el desarrollo de sistemas de visión artificial es conveniente hacer una comparación entre el humano y el artificial, para tener bien claras las capacidades y limitaciones de ambos.

Sistema Humano

- Mejor reconocimiento de objetos.
- Mejor adaptación a situaciones imprevistas.
- Utilización de conocimiento previo.
- Mejor en tareas de alto nivel de proceso.
- Presenta cansancio y enfermedades que influyen en el desempeño.

Sistema Artificial

- Mejor midiendo magnitudes físicas.
- Mejor para la realización de tareas rutinarias.
- Mejor en tareas de bajo nivel de proceso.
- Mejor en el análisis de objetos

1.3 Procesamiento digital de imágenes

El procesamiento digital de imágenes puede ser entendido como la transformación de una imagen en otra imagen, es decir a partir de una imagen (entrada) se obtiene otra imagen modificada (salida).

En el contexto de la visión artificial el único propósito del procesamiento de imágenes es hacer el análisis posterior más simple y más fiable, con lo que se puede decir que el procesamiento de imágenes debe facilitar la extracción de información para un posterior análisis, de tal manera que la escena pueda ser interpretada de alguna manera.

Para llevar a cabo el procesamiento de imágenes es necesario definir las siguientes etapas:

1. Adquisición de la imagen: se realiza a través de un captor (cámara), el cual lleva a cabo los procesos de muestreo y cuantización.
2. Mejoramiento: en esta etapa se tratan los aspectos relacionados con el realce, la restauración y las transformaciones geométricas.

1.3.1 Representación de una Imagen digital

Una imagen puede ser definida como una función bidimensional $f(x,y)$, donde x e y son las coordenadas espaciales, y la amplitud de f en cualquier par de coordenadas (x,y) es llamada la intensidad de la imagen en ese punto.

Las imágenes digitales están compuestas de un número finito de elementos referidos como "*picture elements*", "*image elements*", "*pels*" o "*pixels*" y cada uno de ellos tiene una posición y valor particular en estructura bidimensional.

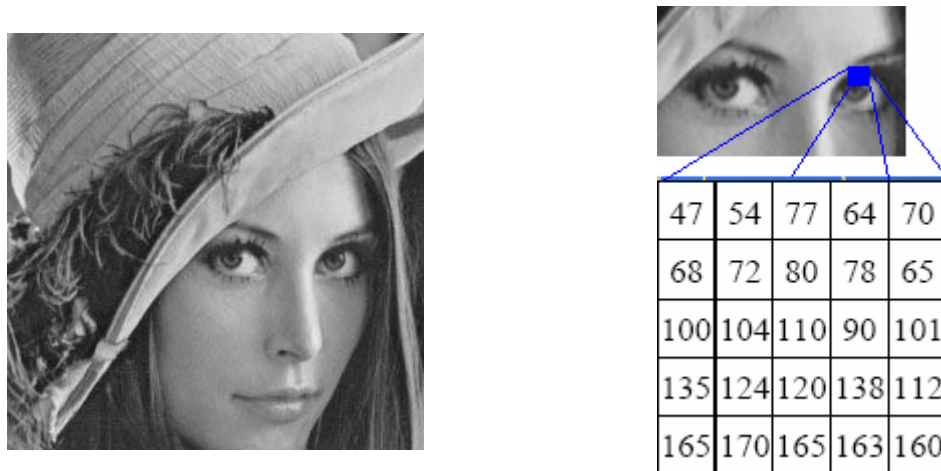


Figura 1.4. Imagen digital

Convertir una imagen a su forma digital requiere que las coordenadas, así como la amplitud sean digitalizadas. El proceso de digitalizar los valores de las coordenadas es llamado “muestreo” y digitalizar los valores de amplitud es llamado “cuantización”, entonces cuando x e y , y los valores de f son todos finitos (valores discretos) la imagen es una imagen digital de $N \times M$ con niveles de grises.

Según el número de píxeles que tenga el dispositivo sensor, la imagen tendrá mayor o menor resolución espacial. En la figura 1.5¹ se muestran varias representaciones de la misma imagen con variación en el número de píxeles y 128 niveles de grises. Puede observarse la pérdida de información cuando la imagen es representada con menor número de píxeles y se se dice que tiene menor resolución por lo tanto menor definición.



¹ Imágenes tomadas de (Pajares, 2002) [1].



Figura 1.5. Imagen con variación en el número de píxeles (muestreo): (a) 200x320, (b) 100x160, (c) 50x80 y (d)25x40 manteniendo el rango de niveles de intensidad entre 0 y 128

En la figura 1.6², mostramos la misma imagen con resolución 200 x 320 píxeles, pero ahora utilizando diferentes niveles de gris (cuantización):1 (a) 64; (b) 32; (c) 8 y (d) 2.

La tecnología actual permite utilizar hasta 10 bits de información para codificar la intensidad de cada píxel, aunque lo más común es tener 8 bits, lo equivalente a 256 niveles de gris con lo que se pueden hacer aplicaciones industriales reales.



² Imágenes tomadas de (Pajares, 2002) [1].

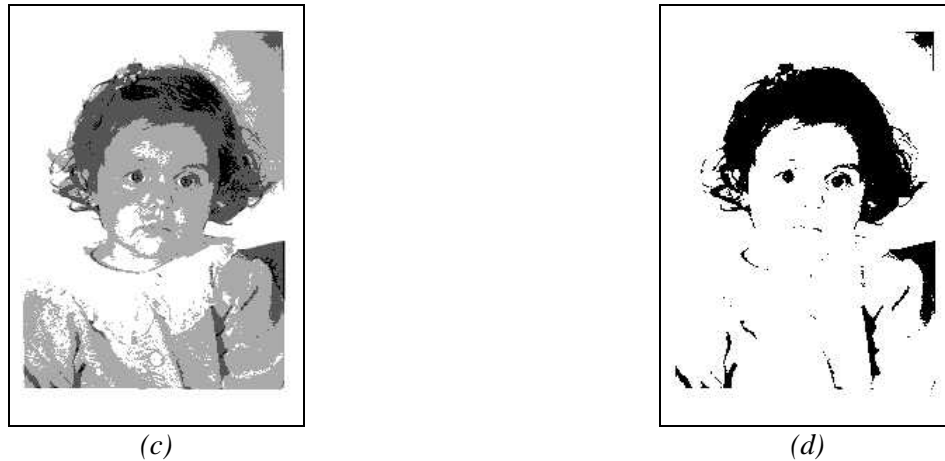


Figura 1.6. Imagen con resolución espacial de 200x320 píxeles, pero ahora utilizando diferentes niveles de gris en (a) 64; (b) 32; (c) 8 y (d) 2.

1.3.2 Histograma de una imagen

El histograma de una imagen es una función discreta que muestra la distribución de niveles de gris en una imagen dando lugar a la función de densidad de probabilidad, es decir, la probabilidad de que un determinado nivel de gris aparezca en la imagen y se define como,

$$P(g) = \frac{N(g)}{M}$$

donde M es el número de píxeles en la imagen y N(g) es el número de píxeles en el nivel de intensidad g.

En la figura 1.7 se muestra una imagen de grises (a), su representación matricial de la fila y columna 1 a la 7 (b) y su histograma (c).

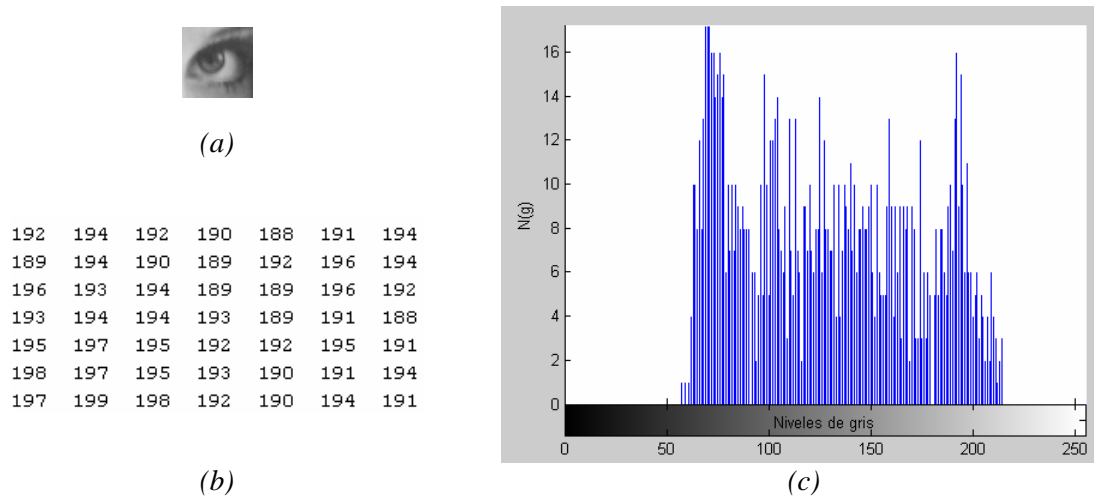


Figura 1.7. Representaciones de una imagen, (a) niveles de gris, (b) matricial, (c) histograma.

Un histograma con los niveles de gris concentrados en la parte baja del rango corresponden a una imagen oscura y un histograma con los valores concentrados en la parte alta corresponden a una imagen brillante.

1.3.3 Adquisición de imágenes

En el campo de la visión artificial, actualmente los medios que se utilizan para la adquisición de una imagen digital son las cámaras de televisión con sensor CCD (figura 1.8), el dispositivo llamado CCD (Coupled Charge Device) es un sensor que utiliza elementos semiconductores fotosensibles, es decir, transforma la energía de la luz incidente en energía eléctrica, la cual es almacenada en una celda.

En las cámaras CCD estos elementos se encuentran distribuidos en forma de un arreglo matricial de $M \times N$ elementos donde M es el número de renglones, N el número de columnas y el valor de cada elemento de esta matriz representa el nivel de intensidad de un píxel de la imagen digital.

Las cámaras CCD son consideradas como dispositivos que poseen baja deformación geométrica de la imagen, muy buena sensibilidad a la luz y una muy buena resolución.

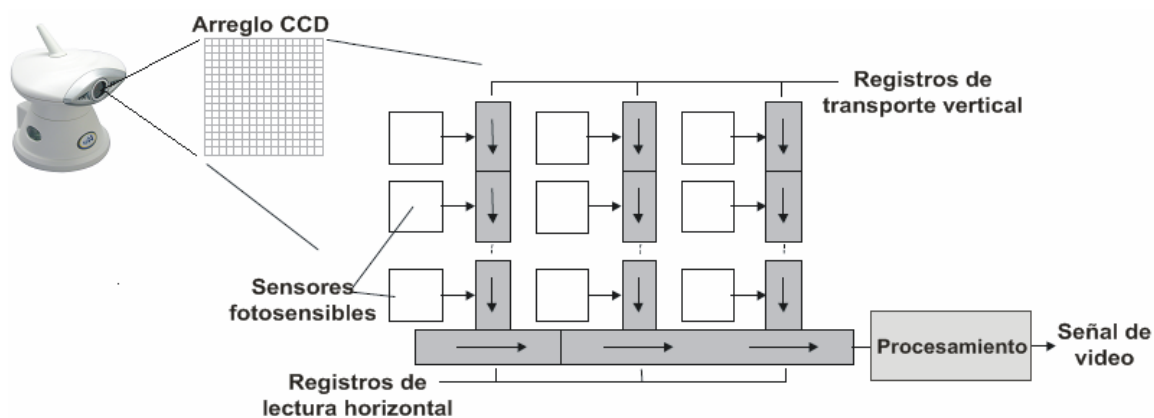


Figura 1.8. La cámara CCD

1.3.3.1 Tipos de imágenes digitales

La forma más general de clasificar las imágenes digitales de mapas de bits es según la cantidad y tipo de información que se asigne a cada píxel.

Dentro de esta clasificación se tienen varios tipos de imágenes digitales, siendo los más comunes los siguientes: *Imágenes de 1 bit por píxel (imágenes binarias)*, *imágenes de 8 bits por píxel (imágenes en escala de grises)* y *las imágenes de 24 bits por píxel (imágenes RGB)*.

Imágenes binarias

En este tipo de imágenes cada píxel sólo puede tener uno de dos valores: 1 ó 0. Como basta un bit para definir esa alternativa, se les llama "imágenes de 1 bit", también son conocidas como "imágenes de mapa de bits, de alto contraste, de línea o binarias".



Figura 1.9. Imagen binaria

Imágenes en escala de grises

En estas imágenes cada píxel puede tener 256 valores diferentes (*las 256 posibilidades combinatorias de un byte*). Este es el modo de las imágenes digitales que comúnmente conocemos como imágenes en blanco y negro.



Figura 1.10. Imagen en escala de grises

Imágenes RGB

En estas imágenes a cada píxel se le asignan tres bytes, con lo cual se dispone de 24 bits en tres grupos de ocho, y se le asigna color siguiendo el sistema de los monitores de televisión, que se basan en tres canales de luz de color (*Rojo, Verde y Azul*). De este modo podemos distinguir hasta 16.777.216 millones de tonos de color (*256 Rojo x 256 Verde x*

256 Azul). En realidad, lo que se hace es superponer tres canales de luz, uno rojo, otro verde y otro azul, cada uno con 256 posibles valores.



Figura 1.11. Imagen RGB

1.3.3.2 Iluminación

El objetivo de la iluminación es optimizar el contraste de una imagen, es muy importante el escoger la fuente de iluminación correcta, ya que una buena selección de iluminación nos permite obtener una imagen más exacta, lo que nos ahorra tiempo y procesamiento.

Existen varias fuentes y técnicas de iluminación, la selección correcta de estas depende del entorno y del tipo de información que será adquirida por el sistema de visión artificial.

Fuentes:

- *Luz de día*. No es la más conveniente por que depende de la intensidad de la luz y el color que varia según la hora del día.
- *Luz incandescente (foco)*, Son económicas, fáciles de utilizar y pueden funcionar durante periodos largos de tiempo, pero no son muy recomendables debido al gran calor que desprenden y a que la frecuencia del bulbo puede causar conflicto con el sistema.
- *Luz fluorescente*. La ventaja de este tipo de luz es que nos proporciona un campo de iluminación homogéneo, el inconveniente es que hay disminución del nivel de iluminación con el paso del tiempo.
- *Diodo emisor de Luz (LED)*. Las ventajas de esta fuente es que son económicos, consumen poca energía, son pequeños y tienen un periodo de vida largo, el único inconveniente que presentan es una intensidad de iluminación muy baja.

- *Luz Estroboscópica.* Permite analizar objetos en movimiento y su iluminación es muy intensa, las desventajas que presenta son que necesita sincronización con la cámara y que su intensidad va decreciendo con el tiempo.
- *Luz láser.* Es altamente monocromática, es capaz de enfocar en áreas muy pequeñas y permite generar una gran diversidad de patrones de luz, presenta un mal comportamiento frente a superficies que absorben luz.

Técnicas

- *Iluminación frontal.* Técnica en la cual la fuente de iluminación se coloca frente a la escena de tal forma que esta se perciba de manera adecuada.

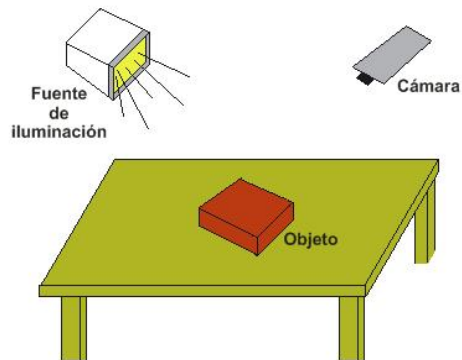


Figura 1.12. Iluminación Frontal

- *Iluminación a contraluz.* Técnica en la cual la fuente de iluminación se coloca detrás de la escena y de un material difusor de tal forma que esta sea percibida según convenga.

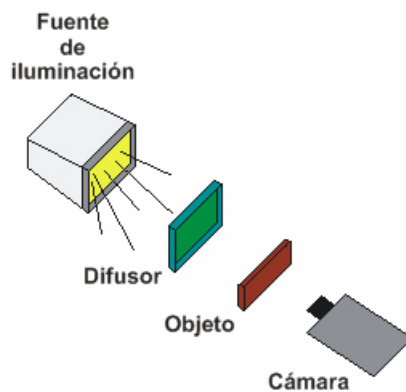


Figura 1.13. Iluminación a contraluz

- *Iluminación Estructurada.* Técnica en la cual la fuente de iluminación es colocada con una inclinación conocida, la cual a través de su proyección sirve para obtener ciertas características de la escena.

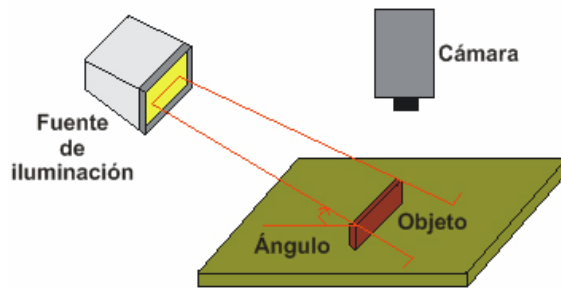


Figura 1.14. Iluminación estructurada

1.3.4 Mejoramiento

El objetivo del mejoramiento de imágenes es acentuar ciertas características de la imagen, tales como bordes, fronteras, o el contraste para generar una representación grafica más útil para su análisis posterior. El mejoramiento de imágenes incluye la manipulación de los niveles de gris y del contraste, realce de contornos, reducción de ruido, suavizado de la imagen, filtrado, decimación, interpolación y otras técnicas mencionadas en la figura 1.15. La gran dificultad para el mejoramiento de imágenes es escoger la técnica adecuada para nuestras necesidades, por lo cual muchas de las técnicas son empíricas y requieren ser probadas para obtener resultados satisfactorios.

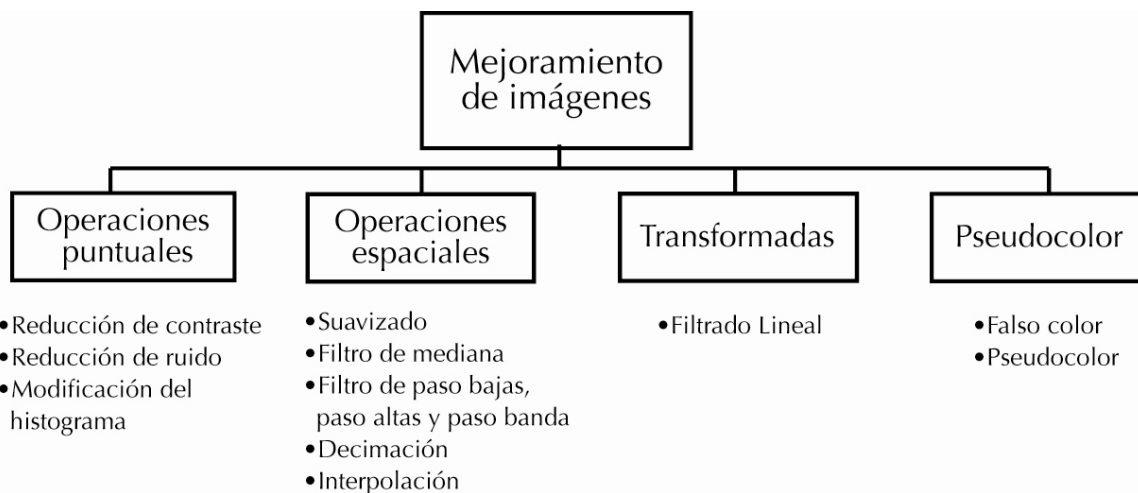


Figura 1.15. Algunas técnicas utilizadas para el mejoramiento de imágenes

Para abordar este tema, dividimos el mejoramiento de imágenes en dos partes: realce y restauración, dado que el presente proyecto hizo necesaria la utilización únicamente de técnicas de restauración presentamos de una manera breve algunas técnicas de realce para posteriormente ahondar más en el tema de restauración, el cual constituye una parte importante de este proyecto.

1.3.4.1 Realce

El realce de imágenes consiste en destacar ciertos valores o regiones dentro de una imagen para distinguir mejor los detalles finos de una imagen o intensificar detalles que han sido difuminados, ya sea por efecto natural en la adquisición de la imagen o por error.

Realce a nivel de gris

Para la mejora del contraste de la imagen se utiliza la técnica de la igualación o equalización del histograma que consiste en uniformizar los niveles de gris de la imagen, es decir, distribuir los niveles de gris de una manera uniforme a lo largo de todo el rango de valores de niveles de gris. En la figura 1.16 se muestra la imagen original (a) y su histograma (c) y el histograma ecualizado (d) y su imagen correspondiente (b).

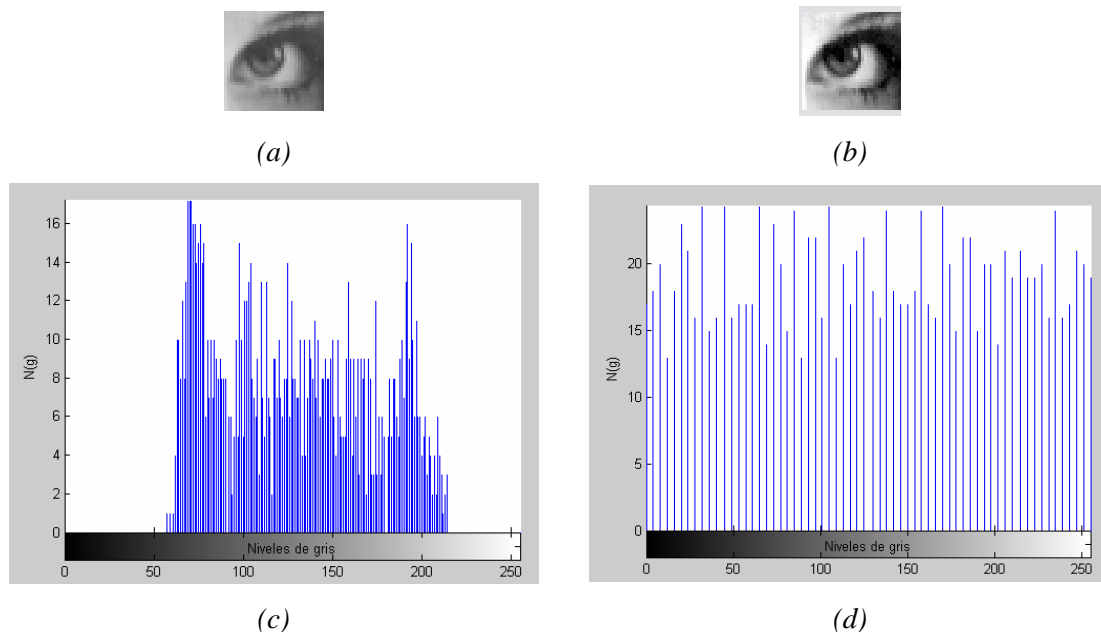


Figura 1.16 Imagen a nivel de gris (a), histograma (c), imagen ecualizada(b), histograma ecualizado(d),

Realce de Contornos

Es un método en el cual se cambian los valores de intensidad en una imagen digital para mejorar la detección de características de borde en la escena. Es llevado a cabo por filtros matemáticos paso altas, los que enfatizan los detalles en una imagen.

Una de las técnicas más usadas es aplicar a la imagen un filtro o kernel de factores de corrección, siendo los más comunes el de 3x3 píxeles. Este kernel de transformación se coloca sobre un píxel en particular y se calcula su nuevo valor dependiendo de la función elegida para el filtro.

Realce de color

El realce de color en una imagen requiere el mejoramiento del balance del color o del contraste en la imagen a color. El realce en imágenes de color resulta una tarea difícil, no solo por la dimensión de los datos que se adquieren, sino también por la complejidad de la percepción del color.

Un acercamiento práctico al realce de imágenes a color se muestra en el algoritmo mostrado en la figura 1.17

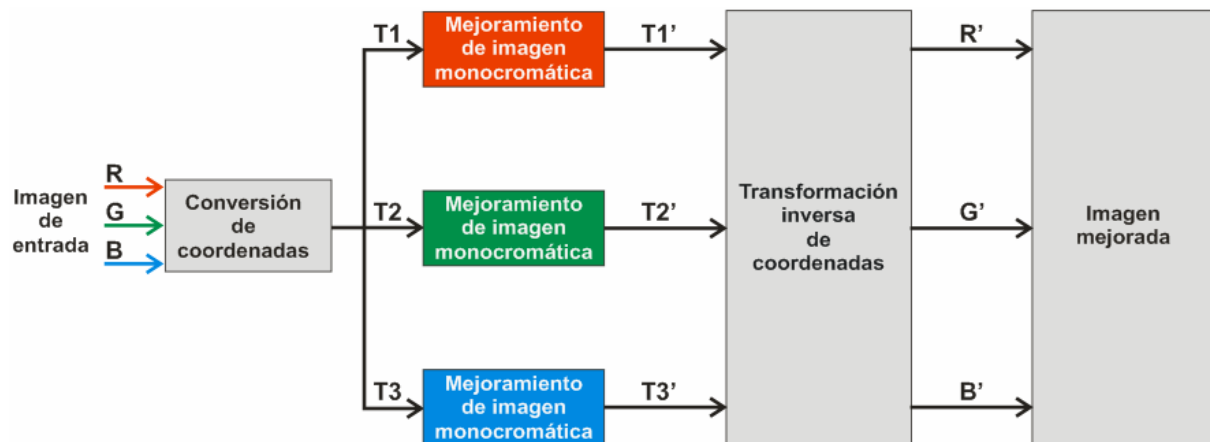


Figura 1.17. Realce de imágenes

Las coordenadas de entrada de color de cada píxel son transformadas independientemente en otro sistema de coordenadas de color, donde la imagen en cada coordenada es realizada por su propio algoritmo (monocromático) de realce de imágenes, que puede ser elegido de una variedad de algoritmos existentes. Las coordenadas de la

imagen realzada T'_1 , T'_2 , T'_3 , son transformadas inversamente para obtener el despliegue de la imagen ya mejorada en sus colores correspondientes R' , G' , B' .

1.3.5 Restauración

El objetivo principal de las técnicas de restauración consiste en la mejora de una imagen, ya que al adquirirla con una cámara, se presentan en esta, distintos tipos de degradaciones, ya sean debidas al propio dispositivo o al ambiente de iluminación por las condiciones atmosféricas si es una escena exterior.

Para lograr dicho objetivo el procesamiento digital de imágenes se vale de distintas técnicas que trabajan en el dominio de la frecuencia o que toman en cuenta los valores de intensidad de los pixeles de una determinada región de la imagen, estas últimas también conocidas como “operaciones espaciales”.

En el presente trabajo solo se tratarán las operaciones espaciales, debido a su fácil manejo y a las características de las imágenes con las que se trabajó³.

De forma esquemática la restauración el objetivo principal de la restauración de imágenes digitales se muestra en la figura 1.18 (a), es una imagen $I(x,y)$ que se le añade algún tipo de ruido $R(x,y)$, y la imagen que en realidad obtenemos restaurada es la 1.18 (b).

$$I'(x, y) = I(x, y) + R(x, y)$$



(a) Imagen degradada



(b) Imagen restaurada

Figura 1.18.

1.3.5.1 Ruido

Se puede definir el “ruido” en una imagen digital, como la información no deseada presente en la imagen y que es debida a una gran variedad de fuentes, que añaden valores aleatorios a los valores exactos de la intensidad en determinados pixeles.

Los tipos de ruido más comunes son el *gaussiano* y *el sal y pimienta*, existen otros tipos, pero por lo general estos son los que más a menudo se presentan en una imagen.

³ En este trabajo solo se trabajo con imágenes binarias, por lo tanto el procesamiento requerido es simple

El ruido puede ser *aditivo*, en este caso se suma un valor aleatorio al valor exacto de intensidad de un píxel, la otra forma en la que se puede presentar el ruido es de forma *multiplicativa*, en cuyo caso los valores de intensidad de los píxeles de la imagen se ven multiplicados por un factor.

Los modelos de ruido antes descritos se describen con las siguientes ecuaciones.

$$\text{Ruido aditivo : } I'(x, y) = I(x, y) + R$$

$$\text{Ruido multiplicativo : } I'(x, y) = R \times I(x, y)$$

Ruido gaussiano

En este caso, el valor exacto de cualquier píxel es diferente cada vez que se captura la misma imagen. Este efecto, suma o resta un determinado valor al nivel de gris real y es independiente de los valores que toma la imagen. En la figura 1.19 se puede apreciar el efecto que tiene el ruido gaussiano en las imágenes.

El ruido gaussiano tiene un efecto general en toda la imagen, es decir, la intensidad de cada píxel de la imagen se ve alterada en cierta medida con respecto a la intensidad en la imagen original.

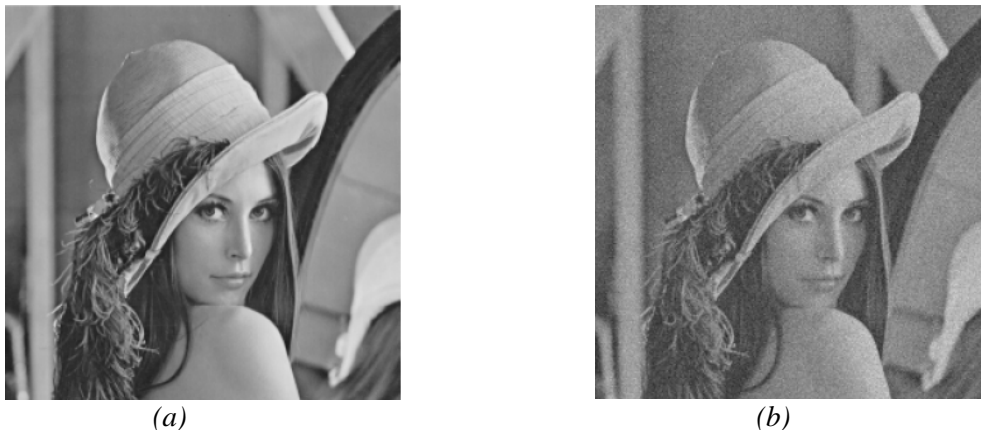


Figura 1.19. (a) Imagen original, (b) imagen con ruido gaussiano

Ruido sal y pimienta

En este tipo de ruido existen sólo dos posibles valores, a y b y la probabilidad de cada uno es típicamente menor que el 0.1 del total del histograma de la imagen, con valores mayores el ruido puede dominar la imagen. Para una imagen de 256 niveles de gris el valor típico de la pimienta es 0 y de la sal 255. En la figura 1.20 se muestra el efecto del ruido de sal y pimienta en una imagen.

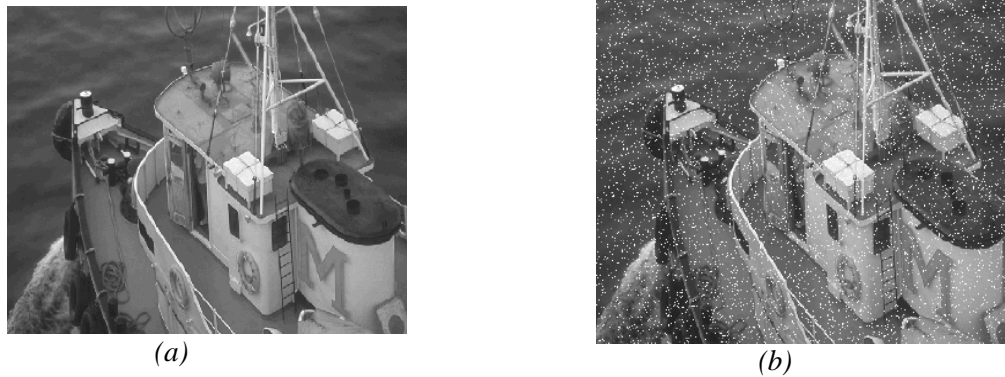


Figura 1.20. (a)Imagen original, (b) imagen con ruido sal y pimienta

1.3.5.2 Operaciones espaciales

Como se mencionó anteriormente, para la restauración de imágenes se utilizan algunas técnicas, tanto en el dominio de la frecuencia como en el espacial. El aplicar una operación en el dominio espacial se reduce prácticamente a la aplicación de un filtro a una imagen.

Los filtros que se pueden aplicar a una imagen se dividen en tres tipos, el filtro lineal, el filtro no lineal y el filtro estadístico, siendo los dos primeros los tratados en esta sección, ya que los filtros estadísticos se utilizan principalmente en el procesamiento de imágenes satelitales e imágenes médicas.

Retomando el término de “*dominio espacial*”, se dice que este se refiere al plano de la imagen misma, y los métodos usados en este dominio se basan en la manipulación directa de los píxeles de la imagen. Para llevar a cabo dichas manipulaciones es necesario definir otro concepto, el de “*vecindad*”.

Se define como vecindad alrededor de un punto $C(x,y)$, a la región cuadrada o rectangular con centro en $C(x,y)$ figura 1.21.

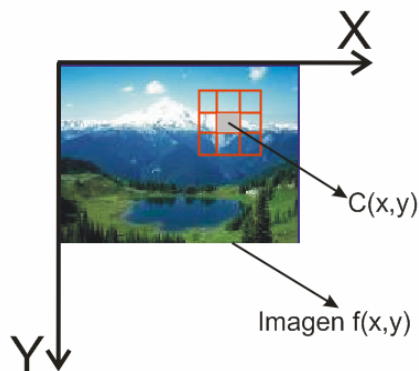


Figura 1.21. Vecindad de 3×3 alrededor del punto $c(x,y)$ sobre una imagen

Cuando nos referimos a alguna operación que involucra el concepto anterior, solo intervienen en esta los pixeles que se encuentran dentro de la vecindad para obtener el resultado.

Filtrado espacial

Cuando se habla de filtrado de una imagen se involucran manipulaciones de los pixeles de esta y para llevarlas a cabo es necesario lo siguiente.

1. Definir el tamaño de la vecindad, así como de su punto central.
2. Realizar la operación determinada, cuidando que esta únicamente tome en cuenta los pixeles de la vecindad.
3. obtener el resultado de la operación, como la respuesta del proceso en ese punto.
4. repetir el proceso para cada píxel de la imagen.

Al proceso de mover el punto central $c(x,y)$ y crear una nueva vecindad de igual tamaño, una vez para cada píxel de una imagen, se le denomina *procesamiento de vecindad o filtrado espacial*.

Filtrado lineal

El filtrado lineal consiste en la multiplicación de cada píxel que se encuentra dentro de la vecindad por un determinado coeficiente y sumar los resultados de los productos para obtener la respuesta en cada punto.

Si la vecindad es de orden $n \times m$, se requieren mn coeficientes, los cuales son acomodados en un arreglo matricial llamado filtro, máscara, kernel, plantilla o ventana.

Para tener una idea clara sobre el proceso del filtrado lineal se presenta la figura 1.22, en esta figura se indica de manera gráfica que la máscara se coloca sobre cada punto de la imagen y que además el recorrido sobre esta se hace de izquierda a derecha y de arriba para abajo, cabe mencionar que en los puntos que corresponden a los límites de la imagen, al hacer coincidir el centro de la máscara con ellos, la máscara estaría intentando ocupar puntos que ya no pertenecen a la imagen, por lo tanto a la hora de operar en estos puntos se hace una adición "*virtual*" de ceros, los cuales no influyen en el resultado del filtrado, y se añaden tantos como los requiera la máscara para operar adecuadamente.

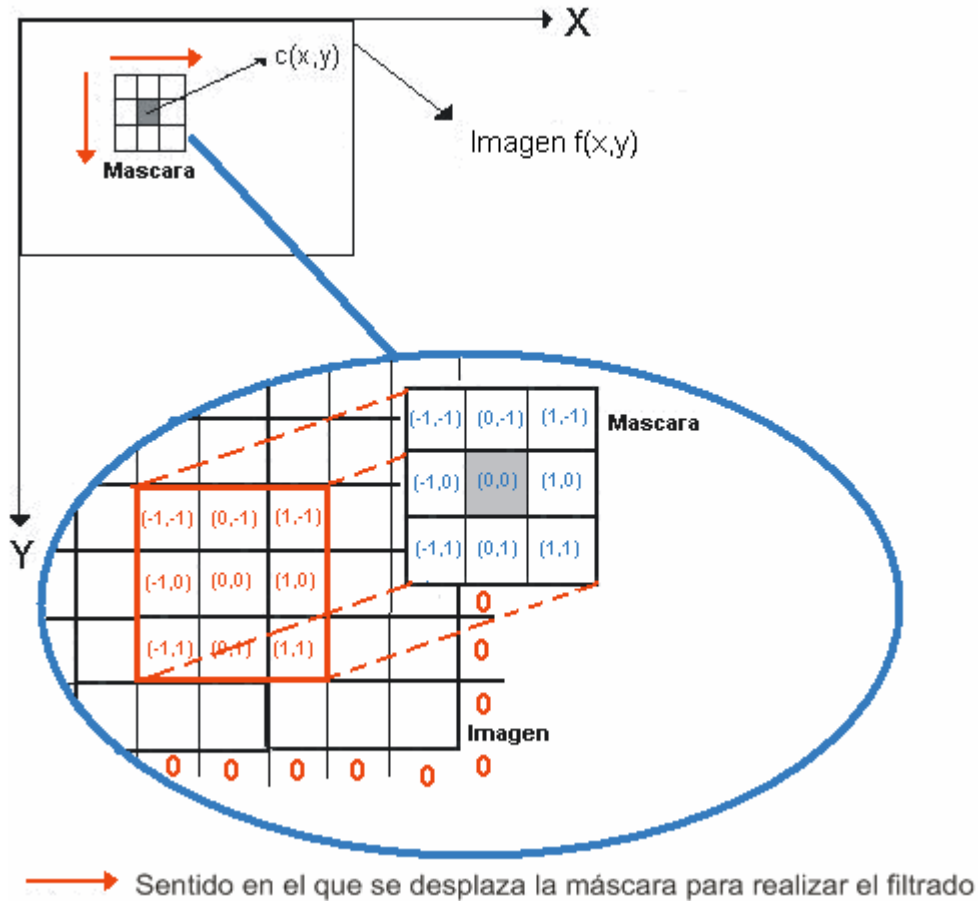


Figura 1.22. Filtrado lineal

El proceso consiste simplemente en mover el centro de la máscara w de píxel en píxel sobre la imagen f .

La respuesta del filtro en cada punto de f es igual a la suma de los productos correspondientes a los coeficientes de la máscara por los píxeles que se encuentran dentro del vecindario al colocar la máscara sobre un punto (figura 1.23).

Por convención, el tamaño⁴ de las máscaras ($m \times n$) está dado por $m=2a+1$, $n=2b+1$, donde a y b son enteros positivos y mayores iguales a uno.

⁴ Por lo general se utiliza tamaño de máscara impar, logrando con esto una mejor operatividad, ya que se tiene un único punto central.

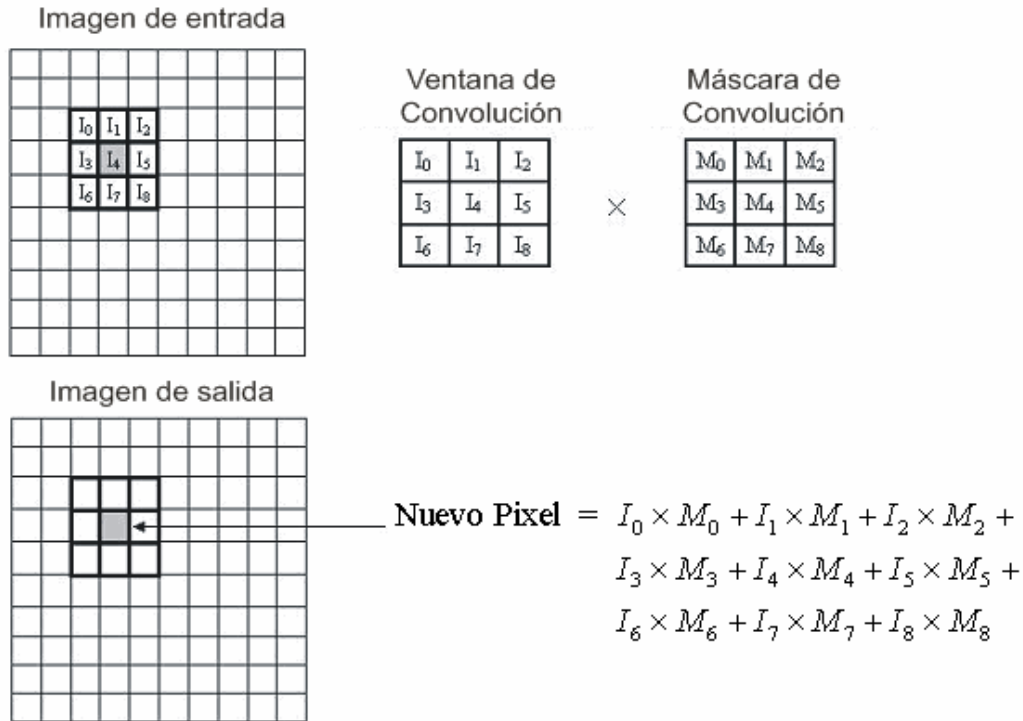


Figura 1.23.

Cuando se trabaja con filtrado lineal⁵ es necesario el manejo de dos conceptos muy importantes, *correlación* y *convolución*⁶.

Correlación: es el proceso de mover la máscara por cada uno de los píxeles de una imagen como se ilustró en la figura 1.22.

Convolución: es el mismo proceso que la correlación, pero en este caso la máscara se rota primero 180°.

En la figura 1.24, se ilustra en (a) la imagen (5x5) y una máscara (3x3) que se utiliza para ejemplificar los procesos de correlación y de convolución, en (b) la misma imagen, solo que en este caso se ha expandido con ceros (ahora es de 9x9) en los límites para que la máscara de 3x3 pueda operar en ellos, en (c) la posición inicial de la máscara w para el proceso de correlación, en (d) el resultado final de la correlación, una vez que la máscara recorrió los 81 puntos que conforman la imagen, (e) posición inicial de la máscara w' para el proceso de la convolución, nótese que no es la máscara w , ya que para la convolución la

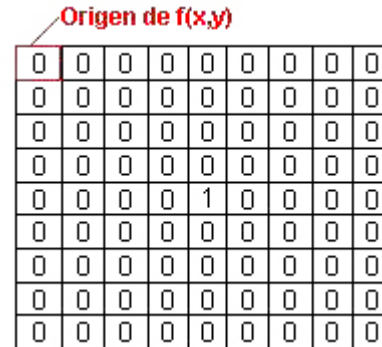
⁵ En Matlab se utilizan las funciones `imfilter`.

⁶ Operación lineal aplicable únicamente a sistemas lineales invariantes en el tiempo (SLIT).

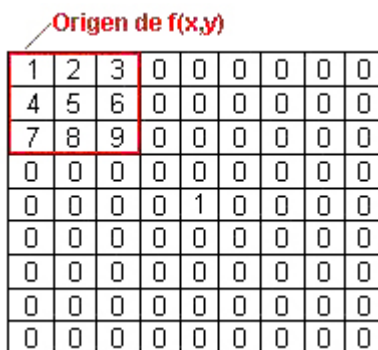
máscara original w primero tiene que ser rotada 180° y por ultimo en (d) el resultado final de la convolución.



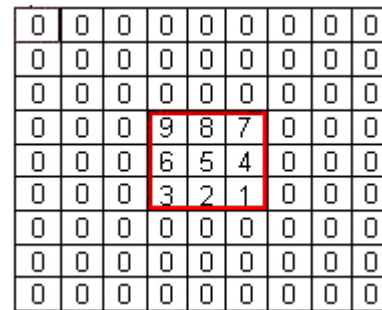
(a) Imagen y máscara de 3x3



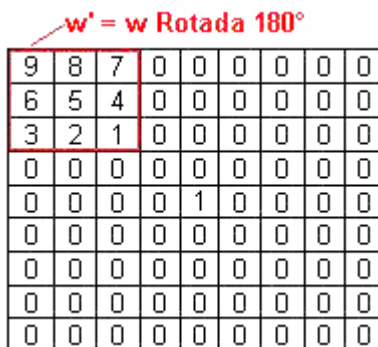
(b) Imagen con ceros añadidos



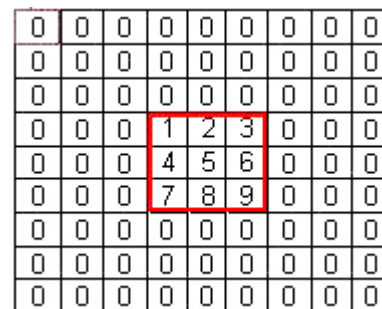
(c) Inicio del proceso de correlación



(d) Resultado de la correlación



(e) Inicio del proceso de convolución



(f) Resultado de la convolución

Figura 1.24. Proceso de correlación y convolución

El filtrado lineal se utiliza para eliminar o reducir el ruido de tipo gaussiano y multiplicativo en imágenes que se encuentran en escala de grises.

Por lo general, para el filtrado lineal se utiliza alguna de las siguientes máscaras:

0	-1	0
-1	5	-1
0	-1	0

(a) *Máscara H1*

-1	-1	-1
-1	9	-1
-1	-1	-1

(b) *Máscara H2*

1	-2	1
-2	5	-2
1	-2	1

(c) *Máscara H3*

Figura 1.25. Máscaras utilizadas para el filtrado lineal

Con respecto al filtrado lineal de frecuencia en imágenes se pueden definir distintos tipos de filtros, dependiendo de lo que se requiera hacerle a la imagen.

Estos filtros pueden ser:

- *Filtros paso bajas*
- *Filtros paso altas*
- *Filtros paso banda*

Filtrado no lineal

En el filtrado no lineal⁷, la definición de la vecindad y el proceso de mover un filtro por todos y cada uno de los pixeles de una imagen son exactamente iguales que los descritos para el filtrado lineal, sin embargo, mientras que en el filtrado lineal se manejan sumas y productos (operaciones lineales), en el filtrado no lineal se parte de la utilización de operaciones no lineales que involucran a todos los pixeles de una vecindad. Se podría pensar que ambos filtrados son iguales, pero es importante aclarar que una diferencia inmediata es que el concepto de máscara no es extensible para este filtrado, en este caso el filtro puede visualizarse como una función no lineal cuya respuesta esta constituida por la respuesta de la función sobre los pixeles de la vecindad con centro en un determinado punto.

El filtrado no lineal se utiliza principalmente para eliminar o reducir ruido de tipo sal y pimienta en imágenes en escala de grises.

Algunos filtros no lineales, también conocidos como filtros de orden, son el filtro de media y el filtro de mediana.

⁷ En Matlab se utilizan las funciones colfilt, nlfilt y ordfilt2

Por lo general para realizar un filtrado no lineal se utiliza alguno de los siguientes filtros.

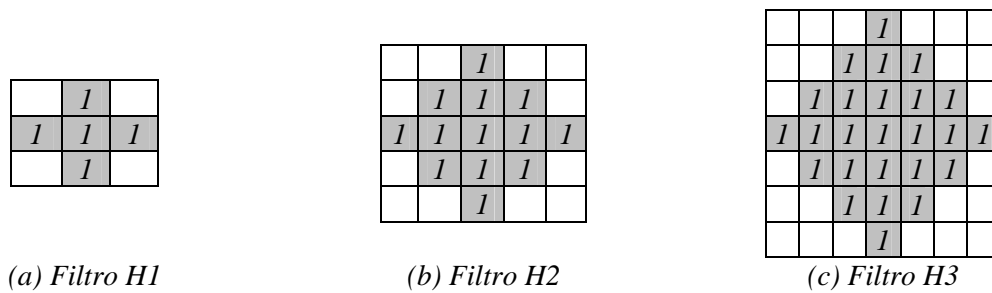


Figura 1.26. Máscaras de filtros utilizadas para el filtrado no lineal

Filtro de mediana

Este filtro ordena de menor a mayor los niveles de gris de los pixeles de la vecindad que intervienen en el filtrado, y posteriormente toma el valor de en medio del arreglo y lo sustituye en la posición central de la vecindad figura 1.27.

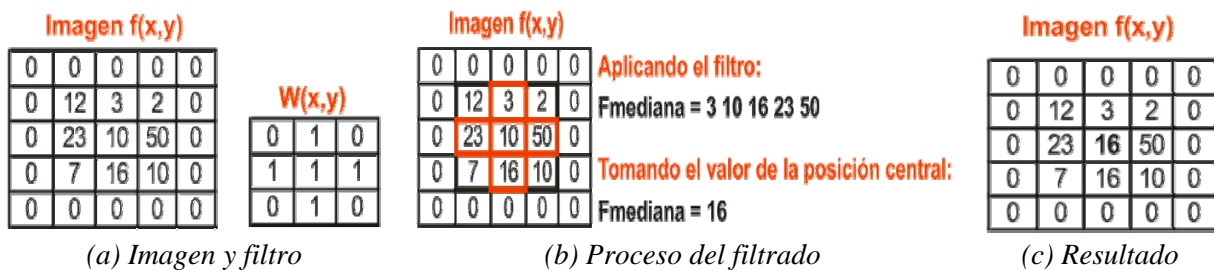


Figura 1.27. Filtrado de mediana

Filtro de media

Este filtro obtiene la media aritmética de los niveles de gris de los pixeles de la vecindad que intervienen en el proceso, posteriormente sustituye este valor en el punto central de la vecindad.

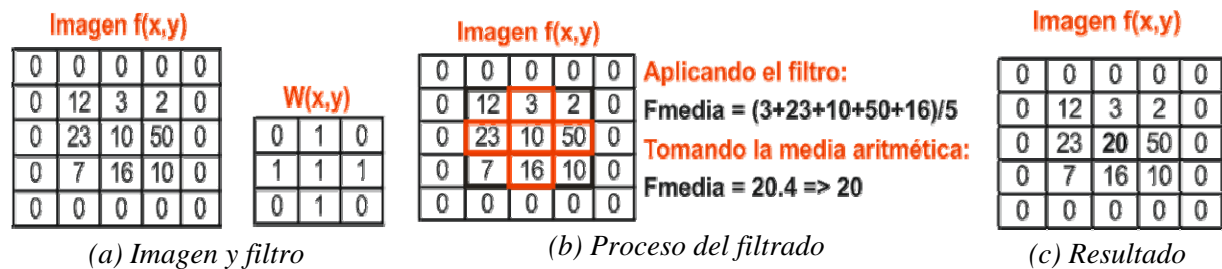


Figura 1.28. Filtrado de media

1.3.5.3 Morfología matemática

La palabra morfología significa forma y estructura de un objeto. Para imágenes binarias se definen operaciones morfológicas, y con estas se constituye una herramienta de extracción de componentes de una imagen, que son útiles en la representación y descripción de la forma de las regiones.

Las operaciones básicas de la morfología matemática son: “dilatación” se puede simplificar a decir que es agregar pixeles a un objeto, hacerlo mas grande, y luego “erosión” es hacerlo mas chico en la región de sus bordes.

Luego la combinación de estas da origen a los operadores “open” y “close”. El primero consiste en aplicar una erosión seguida de una dilatación aplicando la misma forma estructurante, como resultado esta tiende a "abrir pequeños huecos". El close es la aplicación de las operaciones básicas en el sentido inverso, y resulta en "cerrar los huecos".

Conjuntos

Para adentrarnos en este concepto empezaremos por definir que es un conjunto. Un conjunto es una colección bien definida de elementos, en términos de imágenes digitales (binarias)⁸, se puede considerar a estas como un conjunto de puntos 2D, debido a que cada punto esta dado por un par de coordenadas (x,y), donde todos los puntos con valor binario igual a uno son el conjunto X (objetos en la imagen), y los todos los puntos con valor binario igual a cero pertenecen al conjunto complementario X^c (fondo de la imagen).

⁸ Una imagen digital binaria es aquella en la cual el valor de todos los pixeles es solamente uno o cero.

Para el manejo de conjuntos se tienen algunas operaciones elementales, las cuales hemos decidido presentar en forma de diagramas de Venn⁹, para una mejor comprensión.

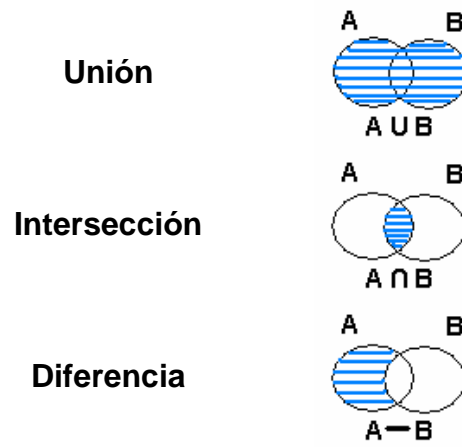


Figura 1.29. Operaciones con conjuntos

Dilatación

Es una operación¹⁰ que expande los objetos que se encuentran en una imagen binaria. La manera en la que se expanden estos objetos es controlada por otro objeto llamado “*elemento estructurante*”, que es un objeto que se desplaza a lo largo y ancho del dominio de la imagen, cuando el origen del elemento estructurante coincide con algún píxel que tiene valor uno, dicho píxel es expandido (se añaden píxeles a su alrededor) en la medida que el elemento estructurante lo permita, es decir, el objeto de la imagen se expande hacia los píxeles que tienen valor cero, siempre y cuando los elementos que componen al elemento estructurante lo indiquen (estos elementos deben de tener el valor de uno).

Matemáticamente el elemento estructurante se representa como una matriz de unos y ceros. Las formas en las que podemos tener al elemento estructurante es en forma de diamante, de disco, de octágono, de línea, de cuadrado y de rectángulo figura 1.30.

⁹ Un diagrama de Venn es un gráfico consistente en dos o más círculos, en el que cada uno de ellos representa un conjunto.

¹⁰ En Matlab se utiliza la función `imdilate`

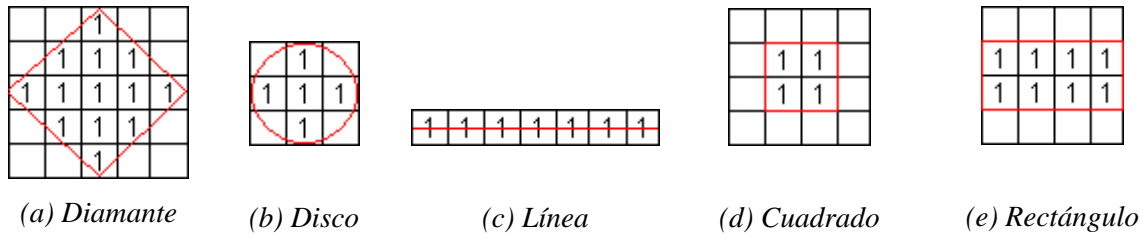


Figura 1.30. Principales formas del elemento estructurante

Matemáticamente la operación de dilatación se expresa como:

$$A \oplus B = \left\{ z \mid \left(\hat{B} \right)_z \cap A \neq \emptyset \right\}$$

A es la imagen

B es el elemento estructura

A continuación se ilustra el procedimiento.

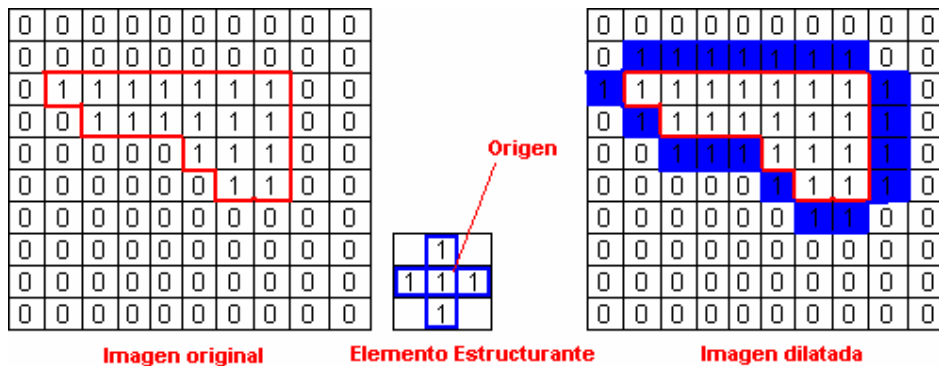


Figura 1.31. Proceso de la dilatación

Erosión

Es una operación¹¹ que contrae los objetos que se encuentran en una imagen binaria. Al igual que la dilatación, la manera en la que se contraen los objetos esta controlada por el elemento estructurante.

El proceso para realizar la erosión se lleva a cabo de la siguiente manera, el elemento estructurante se desplaza por todo el dominio de la imagen, de izquierda a derecha y de

¹¹ En Matlab se utiliza la función `imerode`

arriba para abajo, cuando el origen del elemento estructurante se encuentra sobre un píxel de valor uno, todos los pixeles cuyo valor es igual a uno sobre los que se encuentran superpuestos los demás elementos que componen al el elemento estructurante, incluyendo el píxel sobre el que se encuentra dicho origen, son borrados.

Matemáticamente la operación de erosión se expresa de la siguiente forma:

$$A \ominus B = \left\{ z \mid \left(\begin{matrix} B \\ z \end{matrix} \right) \cap A^c \neq \emptyset \right\}$$

A es la imagen

B es el elemento estructurante

A continuación se ilustra el procedimiento.

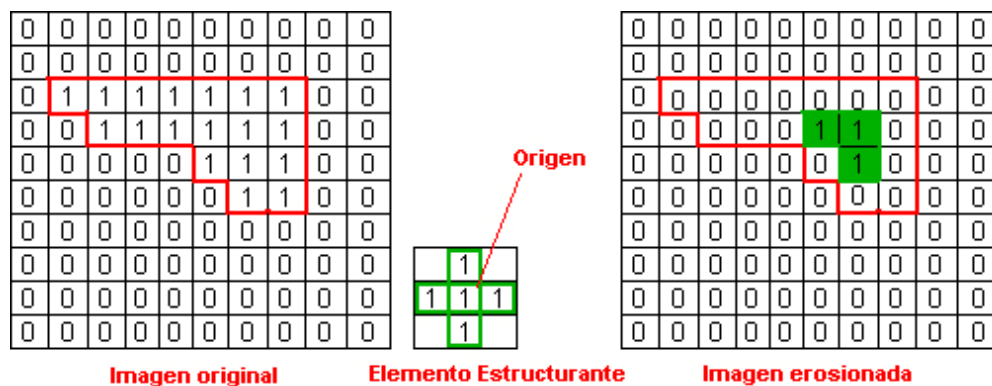


Figura 1.32. Proceso de la erosión

En la práctica, a menudo son utilizadas algunas combinaciones de las dos operaciones anteriores, pudiendo utilizarse distintos elementos estructurantes.

Las combinaciones más importantes que se manejan son conocidas como “opening” y “closing”¹².

¹² En matlab se utilizan las funciones imopen e imclose respectivamente.

Opening

Esta operación está compuesta de una erosión seguida de una dilatación del resultado anterior.

Matemáticamente queda expresada de la siguiente forma:

$$A \circ B = (A \ominus B) \oplus B$$

A es la imagen

B es el elemento estructurante

Closing

Esta operación está compuesta por una dilatación seguida de una erosión del resultado anterior.

Matemáticamente queda expresada de la siguiente forma:

$$A \bullet B = (A \oplus B) \ominus B$$

A es la imagen

B es el elemento estructurante

1.3.5.4 Etiquetado

Tomando una imagen binarizada (imagen de ceros y unos), este método consiste en etiquetar las componentes conexas de forma que el resultado final sea una región. Componentes conexas quiere decir que todos los píxeles que tienen un valor binario "1" y están conectados entre sí por un camino o conjunto de píxeles con el mismo valor, se les asigna una misma etiqueta identificativa, que debe ser única de la región a la cual pertenecen los píxeles y constituye su identificador.

El siguiente ejemplo muestra una imagen en su representación binaria (a) y la segunda matriz es la matriz de etiquetado, que es una imagen del mismo tamaño que la imagen de entrada donde las diferentes regiones en la imagen se distinguen por valores enteros distintos en la matriz de salida (b)

$$\begin{aligned}
 &A = \\
 & \begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0; \\
 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1; \\
 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1; \\
 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0; \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0; \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0; \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0; \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0;
 \end{bmatrix} \quad (a)
 \end{aligned}$$

$$\begin{aligned}
 &B = \\
 & \begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0; \\
 0 & 1 & 1 & 0 & 0 & 3 & 3 & 3; \\
 0 & 1 & 1 & 0 & 0 & 0 & 3 & 3; \\
 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0; \\
 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0; \\
 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0; \\
 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0; \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0;
 \end{bmatrix} \quad (b)
 \end{aligned}$$

En la matriz de salida, los unos representan el primer objeto, los dos el segundo objeto y así sucesivamente.

Otro ejemplo es con la siguiente imagen binaria (figura 1.33), en la cual se observa del lado izquierdo su imagen binaria y del lado derecho la representación de esta imagen etiquetada, dividiéndola en dos regiones (azul y rojo).

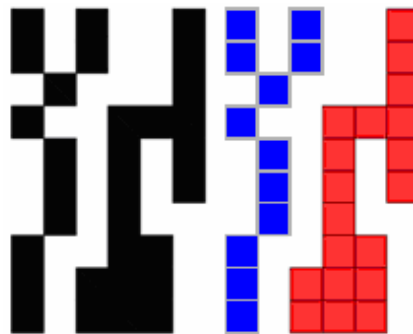


Figura 1.33. Imagen etiquetada

1.4 Reconocimiento de Patrones

Es una disciplina científica cuyo objetivo es la clasificación de objetos físicos o abstractos en un cierto número de categorías o clases, obteniendo información que permita establecer propiedades entre dichas clases de los objetos. Dependiendo de la aplicación estos objetos

pueden ser: imágenes, formas de ondas o señales, o cualquier tipo de medidas que necesitan ser clasificadas. Estos objetos son llamados de forma genérica “*patrones*”.

Existen diferentes enfoques en el reconocimiento de patrones, tales como, el estadístico, el sintáctico, el lógico combinatorio de patrones y las redes neuronales.

Reconocimiento de patrones estadístico: Este enfoque se basa en la teoría de probabilidad y estadística y supone que se tiene un conjunto de medidas numéricas con distribuciones de probabilidad conocidas y a partir de ellas se hace el reconocimiento.

Reconocimiento de patrones sintáctico: Este enfoque se basa en encontrar las relaciones estructurales que guardan los objetos de estudio, utilizando la teoría de lenguajes formales. El objetivo es construir una gramática que describa la estructura del universo de objetos.

Reconocimiento Lógico Combinatorio de Patrones: Este enfoque se basa en la idea de que la modelación del problema debe ser lo más cercana posible a la realidad del mismo, sin hacer suposiciones que no estén fundamentadas.

Redes Neuronales: Este enfoque supone que tiene una estructura de neuronas interconectadas que se estimulan unas a otras, las cuales pueden ser “*entrenadas*” para dar una cierta respuesta cuando se le presentan determinados valores.

Entre otros aspectos el reconocimiento de patrones presenta los siguientes problemas:

- *Selección de variables*: Consiste en determinar cual es el conjunto de características mas relevantes para describir a los objetos a partir de un conjunto total que describe a estos y permite hacer una mejor representación de estos.
- *Clasificación no supervisada*: Consiste en que dada una muestra no clasificada, encontrar la estructuración en clases de la misma, la cual se puede realizar de forma restringida (el número de clases esta previamente definido) o de forma libre (el número de clases depende de los datos).

- Clasificación Supervisada: Consiste en clasificar nuevos objetos basándose en la información de una muestra ya clasificada.

En el presente trabajo utilizamos el reconocimiento de patrones para identificar si en la escena en cuestión se encuentra o no el objeto de interés (el icono), por lo tanto, es importante mencionar que para este fin existen varios métodos, los cuales se clasifican en:

- Descripción de contornos
- Descripción de regiones
- Descripción de texturas

Descripción de contornos: Hablamos de contorno cuando el trazo de línea se une en un mismo punto. Para la descripción de contornos existen diversas técnicas como los códigos de cadenas, firmas, aproximaciones poligonales, esqueletización y descriptores de Fourier entre otras.

Descripción de regiones: Como sabemos, los elementos de las imágenes son los píxeles y las propiedades de un píxel son su posición y su nivel de gris. En las imágenes aparecen ciertas zonas que se caracterizan por el hecho de que constituyen agrupaciones de píxeles conectados entre sí, y además presentan propiedades o características comunes, estas zonas son llamadas regiones.

Algunos descriptores de estas regiones son los siguientes:

Centro de gravedad

El centro de gravedad, también llamado centro de masa es el punto de aplicación de la resultante de todas las fuerzas de gravedad que actúan sobre las distintas masas materiales de un cuerpo. El centroide de un objeto X es la intersección de todos los hiperplanos que dividen a X en dos partes de igual cantidad de movimiento con respecto al hiperplano, Informalmente es el promedio de todos los puntos de X. El centroide es un concepto puramente geométrico mientras que los otros dos términos se relacionan con las propiedades físicas de un cuerpo. Para una imagen se calcula con la fórmula:

$$\bar{x} = \frac{\sum_{x=0}^C \sum_{y=0}^F x \cdot f(x, y)}{A} \quad \bar{y} = \frac{\sum_{x=0}^C \sum_{y=0}^F y \cdot f(x, y)}{A}$$

Área

El área de una región se define como el número de píxeles contenidos dentro de su frontera, es decir, la superficie comprendida dentro de un perímetro. Se calcula mediante la siguiente fórmula:

$$A = \sum_{x=0}^C \sum_{y=0}^F f(x, y)$$

Perímetro

El perímetro de una región es la longitud de su frontera. Este se calcula con la siguiente fórmula:

$$P = \sum_{x=0}^C \sum_{y=0}^F f(x, y) \quad \forall f(x, y) \text{ con algún píxel de } N_4(f(x, y)) = 0$$

Un descriptor que es más útil y que surge de los dos anteriores es la “compacticidad” de una región, y se define como: $\text{perímetro}^2/\text{área}$, Esta es una cantidad adimensional, sensible a cambios de escala e insensible a la orientación.

Momentos invariantes

Estos se pueden calcular cuando la región se da en términos de sus puntos interiores, se puede describir la región mediante un conjunto de momentos que son invariantes a cambios de escala, rotación y a la traslación.

Descripción de texturas: La característica principal de una textura es la representación de un patrón básico. La estructura del patrón básico puede no ser determinística, sino estadística y la repetición del patrón básico puede no ser ni regular ni determinista, sino estadísticamente regular.

Además de la distribución de las instancias de este patrón repetitivo básico en el espacio puede no ser idéntica, esas imágenes pueden estar relacionadas por distorsiones geométricas que permiten obtener la forma a partir de la textura.

Existen varios descriptores de texturas, entre los cuales se encuentran: la descripción por energía, entropía, inercia y homogeneidad.

1.5 Software

Para el desarrollo del sistema tratado en este trabajo se hizo necesaria la utilización de dos lenguajes de programación, *Visual Basic* y *Matlab*, los cuales fueron elegidos y utilizados debido a su facilidad de uso y potencial, además de que es posible desarrollar aplicaciones en Visual Basic que interactúen con programas desarrollados en Matlab a través del componente "*Matlab Automation Server*".

1.5.1 Visual Basic

Visual Basic es una herramienta de diseño de aplicaciones para Windows, en la que estas se desarrollan en una gran parte a partir del diseño de una interface gráfica. En una aplicación Visual Basic, el programa está formado por una parte de código puro, y otras partes asociadas a los objetos y controles que forman la interface gráfica.

Es importante mencionar que el funcionamiento de una aplicación desarrollada con Visual Basic es orientado a eventos, como el click, doble click, presionar una tecla, entre otros, y que la respuesta a dichos eventos se denomina "*Procedimiento*"

Un programa puede hacerse solamente con la programación de los distintos procedimientos que acompañan a cada objeto. Sin embargo, Visual Basic ofrece la posibilidad de establecer un código de programa separado de estos eventos. Este código puede introducirse en unos bloques llamados "*Módulos*" y en otros bloques llamados "*Funciones*", estos bloques no responden a un evento relacionado a un control o formulario, sino que responden a un evento producido durante la ejecución del programa.

1.5.2 Matlab

Matlab hace referencia a *matrix laboratory*, es un entorno de desarrollo interactivo cuyo elemento básico es un arreglo, funciona sin la necesidad de desarrollar programación basada en bucles anidados para realizar operaciones entre estos arreglos.

Este lenguaje da altas prestaciones para la computación en todas aquellas áreas basadas en procesamiento de datos de cualquier índole. Además del lenguaje, Matlab

integra en un entorno de uso bastante sencillo y amigable, una gran cantidad de capacidades de cómputo, visualización y programación.

Matlab se apoya de la ayuda de las cajas de herramientas (*toolboxes*) que consisten en colecciones de funciones para Matlab (*m-functions*) desarrolladas para resolver problemas de un tipo particular entre las cuales podemos citar: procesamiento de señales, procesamiento de imágenes, adquisición de imágenes, sistemas de control, redes neuronales, lógica difusa, caracterización y modelado de sistemas, estadística, entre muchas otras.

Los usos más típicos de Matlab son: análisis matemático y simulaciones numéricas, cálculo simbólico, desarrollo y prueba de algoritmos, modelado de sistemas, análisis estadísticos, gráficos científicos e ingenieriles, desarrollo de aplicaciones y productos finales incluyendo interfaces gráficas, entre otras.

1.5.2.1 Matlab Automation Server

Al utilizar Matlab en el sistema operativo Windows se tiene la capacidad de utilizar un servidor COM¹³ de automatización (Matlab Automation Server), el cual es un protocolo que permite a una aplicación o componente (el cliente) controlar otra aplicación o componente (el servidor). Así un servidor Matlab puede ser controlado por cualquier programa en Windows que pueda ser un controlador de automatización como Excel, Access, Visual Basic y Visual C++.

1.6 Hardware

En esta sección mencionaremos las características generales del *hardware* que compone al sistema desarrollado en este trabajo.

- Cámara
- Tarjeta de red
- Computadora

¹³ *Component Object Model, COM*. Nombre que Microsoft ha dado a un grupo de tecnologías y herramientas "estratégicas" orientadas a objetos.

1.6.1 Cámara

La cámara utilizada es una “*Veo Wireless Observer*”, la cual se comunica con la computadora a través de una red inalámbrica, lo cual implica que se debe de contar con una tarjeta de red inalámbrica, de esta forma la conexión entre la cámara y la computadora puede ser a través de un “*Punto de Acceso*” o una conexión de “*punto a punto*”.

Especificaciones

Marca	Veo
Modelo	Wireless Observer
Sensor	CCD VGA resolución 640 x 480.
Sensitividad	1 a 10,000 lux.
Interfaz de conexión	802.11b
Pan/ Tilt	120° / 60°
Procesador	120 MIPS -RISC
Resoluciones	640x480, 320x240,160x120
FPS	Hasta 10
Formato Video	AVI
Formato Imágenes	TPEG
Profundidad de color	24 bits
Lente	F/2.0, 5 lentes, distancia focal fija
Alimentación	5 volts DC



Conexión Inalámbrica

Modos de conexión	Infraestructura o Adhoc
Rango	Hasta 90 metros.
Cifrado	WEP 64 ó 128 bits
Frecuencia	2.4 GHz DSSS

Tabla 1.1. Especificaciones de la Cámara Veo Wireless Observer

1.6.2 Tarjeta de red

Utilizamos la tarjeta “*3Com® 11 Mbps Wireless LAN PCI Adapter*” que es una potente tarjeta PCI a 40 bits que se instala de una manera rápida y fácil en la computadora permitiendo conectarse con cualquier red inalámbrica que cumpla con el protocolo 802.11b de la IEEE. Esta conexión puede hacerse en modalidad “*punto a punto*” o en modalidad “*infraestructura*” para conectarse mediante un punto de acceso inalámbrico.

Especificaciones

Marca	3Com
Modelo	3CRDW696
Interfaz de conexión	802.11b
Rango	Hasta 100 [m].
Cifrado	WEP 128 bits.
Transmisión de datos	11, 5.5,26 1 [Mbps].
Tipo de bus	PCI
Alimentación	3. 3 V
Frecuencia	2.4 - 2.835 [GHz].



Tabla 1.2. Especificaciones de la tarjeta de red inalámbrica

1.6.3 Computadora

En cuanto al equipo de cómputo utilizado los requerimientos mínimos con los que se debe contar para que el sistema funcione adecuadamente se presentan en la siguiente tabla, y son el resultado del análisis de cada pieza de *software* y *hardware* utilizado en el sistema.

Especificaciones

Procesador	Intel Pentium 4 a 1.0 [Ghz]
Disco duro	40 [GB]
Memoria RAM	512 [Mhz]
Sistema Operativo	Microsoft Windows XP "Profesional"
Visual Basic	v. 6.0
Matlab	v. 7.0
Software de VEO	v. 5.8



Tabla 1.3. Especificaciones del equipo de cómputo

Capítulo 2

Planteamiento del problema

2. Introducción

En el presente capítulo se pretende establecer el objetivo del proyecto, así como definir formalmente el problema para su posterior análisis, y finalmente plantearemos la propuesta de solución.

2.1 Objetivo del proyecto

Una vez revisadas las secciones anteriores del trabajo podemos definir el objetivo de la siguiente manera:

“Diseñar e implementar un sistema de visión artificial mediante el cual se pueda obtener, en tiempo real, la posición exacta de un objeto dentro de un ambiente cerrado por medio del reconocimiento de iconos”

Dicho de otra manera, el objetivo principal se centra en el desarrollo de un programa que se comunique con una cámara para obtener información de su entorno, dicha información deberá contener los parámetros suficientes para poder determinar su localización exacta y posteriormente mostrar una representación gráfica del entorno.

2.2 Definición del problema

En el Departamento de Electrónica y Automatización del Instituto de Investigación en Matemáticas Aplicadas y en Sistemas (IIMAS) de la UNAM, se investigan y desarrollan tecnologías en las áreas de electrónica y computación para la solución de problemas reales tanto en el ambiente académico como en la industria, las disciplinas de estudio son la computación, la electrónica y las matemáticas, y las áreas de especialidad la Visión para Robots y Robots Móviles.

El reconocimiento de patrones dentro de del área de visión por computadora, es una especialización fuertemente explorada y con resultados excelentes en aplicaciones de automatización de procesos para inspección, clasificación o análisis de escenas.

Los algoritmos de reconocimiento por el tiempo de consumo computacional en ocasiones son imprácticos para aplicaciones de tiempo real, por eso nuestro trabajo propone el tener un sistema de reconocimiento de una base iconográfica y su interpretación en tiempo real y

hacer posible la integración de un lenguaje básico para generar la información de localización del objeto observador móvil dentro del entorno cerrado, dando como resultado un sistema de posicionamiento local usando visión.

La idea de tener un sistema que proporcione la posición de un objeto dentro de un entorno cerrado, es potencialmente una herramienta muy poderosa para múltiples aplicaciones en la utilización de trayectorias de movimientos en manipuladores y agentes autónomos que realicen tareas independientes dentro de ambientes cerrados, tal es el caso también de personas invidentes que necesitan desplazarse en ambientes que representen su entorno de desplazamiento.

Por lo antes mencionado, surgió la inquietud de diseñar y construir un sistema que automatice la localización de objetos en entornos cerrados decorados con una base iconográfica, la cual, gracias a su interpretación nos dará los parámetros suficientes para obtener el posicionamiento local del objeto dentro del entorno cerrado. Los iconos pueden ser una base de símbolos que representen movimientos y trayectorias en 3D o incluso símbolos que hagan referencia a ciertas zonas de la estructura cerrada, con el objetivo de que dichos símbolos puedan ser fácilmente captados por una cámara de video y posteriormente extraídos de la escena para su interpretación final.

Habiendo definido el problema se hace necesario establecer los alcances del proyecto ya que el campo de la visión artificial es muy amplio y el no hacerlo implicaría desarrollar un método ineficiente al tomar muchos parámetros en cuenta, material para un trabajo futuro en otras tesis.

Alcances.

- El entorno operativo del sistema deberá ser cerrado y de color blanco.
- Los iconos deberán tener fondo blanco.
- La posición de los iconos dentro del entorno deberá ser tal que:
 - Solo un icono este presente en cada pared del entorno
 - Solo un icono este presente a la hora de extraerlo de la escena
- La respuesta del sistema deberá ser proporcionada en tiempo real una vez que el centro del icono encontrado se encuentre alineado con el eje óptico de la cámara.

- La respuesta del sistema estará conformada por un par de coordenadas (x,y) , las cuales estarán dadas con respecto a un punto de referencia y corresponderán a las coordenadas en las cuales se encuentra ubicado el objeto dentro del entorno cerrado.

2.3 Análisis del problema

Una vez comprendido el objetivo del proyecto y revisado la definición del problema, se establecen los siguientes requerimientos del sistema para cumplir con el objetivo planteado:

a) Selección de cámara.

Además de elegir una cámara digital con la resolución y características adecuadas, debemos tomar en cuenta que la cámara a seleccionar cuente con algún dispositivo de movimiento, el cual será utilizado por el sistema para la búsqueda del icono. Este dispositivo deberá lograr que la cámara sea capaz de moverse libremente con un grado de libertad (paneo).

b) Comunicación PC/Cámara

Debido a que la cámara puede estar en movimiento y alejada de la computadora, la opción que mejor se adapta a este requerimiento es la utilización de una cámara que pueda ser manipulada vía inalámbrica, para evitar conflictos y mal funcionamiento debido a los cables de comunicación.

c) Definición de la base iconográfica para la representación del entorno.

Para la definición de la base iconográfica se debe de pensar en utilizar formas geométricas básicas, dispuestas de tal forma que se facilite su análisis posterior.

d) Construcción del entorno para pruebas de laboratorio (maqueta).

Las dimensiones de este deben de estar en función del espacio de trabajo disponible en el laboratorio y adecuadas para el funcionamiento de la cámara.

e) Adquisición de la imagen en formato estándar para formar una “estructura de datos de la imagen”.

Para la correcta adquisición de las imágenes se debe prestar atención a las limitantes planteadas anteriormente, ya que de estas depende que la estructura de datos de la imagen formada sea útil en la operación de los siguientes algoritmos:

a) Algoritmo para la extracción de la región de interés (icono) de la escena.

Este algoritmo debe ser capaz de detectar la presencia de los iconos diseñados y lograr su extracción para su análisis e interpretación.

b) Algoritmo para la interpretación del icono y obtención de la localización de la cámara.

Este algoritmo deberá de recibir como entrada una estructura de datos de la imagen correspondiente al icono, a la cual se le realizará un análisis que en conjunto con un modelo matemático nos proporcionará las coordenadas en las cuales se encuentra situada la cámara dentro del entorno.

Realización de pruebas.

Una vez implementado el sistema se llevaran a cabo las pruebas y ajustes necesarios para que la respuesta obtenida sea lo más apegado al planteamiento de objetivos.

2.4 Propuesta de solución

La solución planteada a continuación es una de las varias soluciones que puede tener este problema, sin embargo, la decisión de llevar a cabo esta se realizó tomando como criterio el cubrir los objetivos planteados de una manera rápida y eficiente.

Por otra parte hay que mencionar que la metodología utilizada para la construcción del sistema final se basa en la metodología clásica del desarrollo de sistemas, la cual está integrada por las siguientes etapas:

- Análisis
- Diseño
- Desarrollo
- Implementación
- Pruebas

2.4.1 Selección de la cámara.

Como actor principal de nuestro sistema tenemos la cámara, para la cual debemos de considerar una que sea fácil de adquirir y que además cuente con las características anteriormente descritas, siendo la opción mas adecuada la cámara *Veo Wireless Observer* debido a que además de satisfacer estos dos requerimientos, los fabricantes cuentan con un soporte para desarrolladores, donde se encuentran las librerías y documentación necesarios para poder manipular la cámara desde una aplicación desarrollada en los lenguajes *Microsoft Visual Basic, Java o C++*.

2.4.2 Diseño de la base iconográfica.

Para el diseño de la base iconográfica se tiene que pensar en figuras sencillas o compuestas, de tal forma que podamos relacionar cada una con una zona específica del entorno, además de que dichas figuras deben proporcionar parámetros suficientes para su posterior análisis.

Por lo tanto, se propone el uso de cuatro iconos, los cuales serán semejantes al mostrado en la figura 2.1, y se deberá colocar cada uno de ellos en las posiciones que se muestran en la figura 2.2 con el propósito de cumplir con las limitantes antes expuestas. Cada uno de los iconos está conformado por cuatro cuadrados, de los cuales sus centroides corresponden a las aristas de un rectángulo imaginario, se proponen estas posiciones con el objetivo de que al analizar posteriormente sus desplazamientos, ocasionados por la perspectiva, sea posible con ayuda de modelos geométricos, obtener la distancia que existe entre el lente de la cámara y el centro del icono, además, también será posible obtener el ángulo que existe entre el eje óptico de la cámara y la perpendicular que atraviesa el centro del icono (*ángulo de visión*).

En cuanto a las dimensiones planteadas para el diseño del icono es importante mencionar que fueron seleccionadas de tal forma que estas, son las que mejor se adecuan a la resolución y alcance de la cámara seleccionada dentro del espacio de trabajo y fueron determinadas mediante prueba y error.

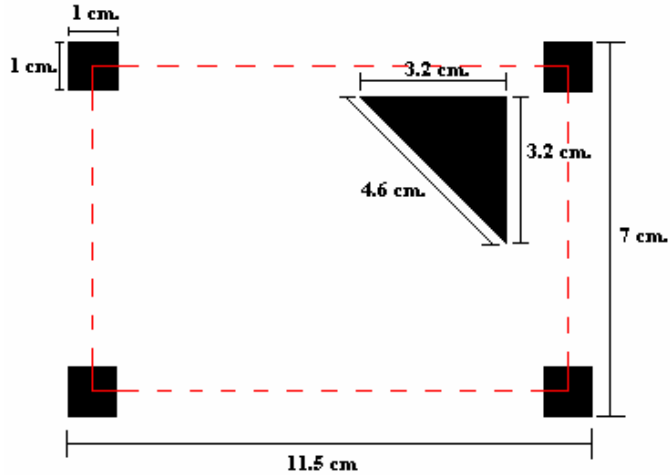


Figura 2.1. Dimensiones icono

Posición propuesta para los iconos dentro del espacio de trabajo.

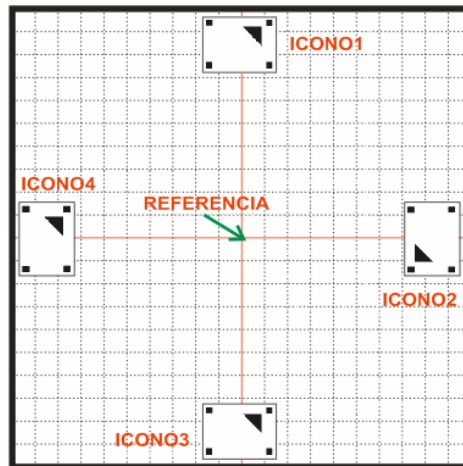


Figura 2.2. Posición de los iconos en el espacio de trabajo

Por otra parte se puede apreciar que el icono también cuenta con un triángulo que está más cercano a uno de los cuatro cuadrados que componen al icono. La posición del triángulo varía en cada uno de los iconos con la finalidad de proporcionarnos la identificación de cada uno de ellos dentro del entorno de trabajo.

2.4.3 Especificaciones para el diseño de la maqueta.

Para el diseño de la maqueta se tomó en cuenta que se iba a trabajar con una resolución en la cámara de 320x240 píxeles, ya que de las tres resoluciones con las que es posible trabajar, es un tamaño de imagen adecuado para un procesamiento rápido. Se realizaron varias tomas de prueba, para ver los rangos de distancia entre el lente de la cámara y el

centro del icono con los cuales era aceptable trabajar, siendo la distancia máxima resultante de 60[cm] y la mínima de 20[cm], las cuales se determinaron gracias al análisis visual de las imágenes obtenidas a dichas distancias, ya que para distancias fuera de este rango era imposible de obtener los parámetros necesarios para la obtención de la distancia y ángulo antes descritos. Una vez obtenidas estas distancias el parámetro siguiente a determinar fue el ángulo máximo en el que es posible obtener una imagen de la cual se puedan extraer los parámetros determinantes de la distancia y el ángulo, resultando como valor del ángulo máximo 55°.

Una vez establecidos estos tres parámetros es posible determinar una región operativa para cada icono, en la cual el sistema será capaz de obtener la posición de la cámara, dicha región es aplicable para cada uno de los cuatro iconos y será denominada región operativa del entorno de trabajo (figura 2.3).

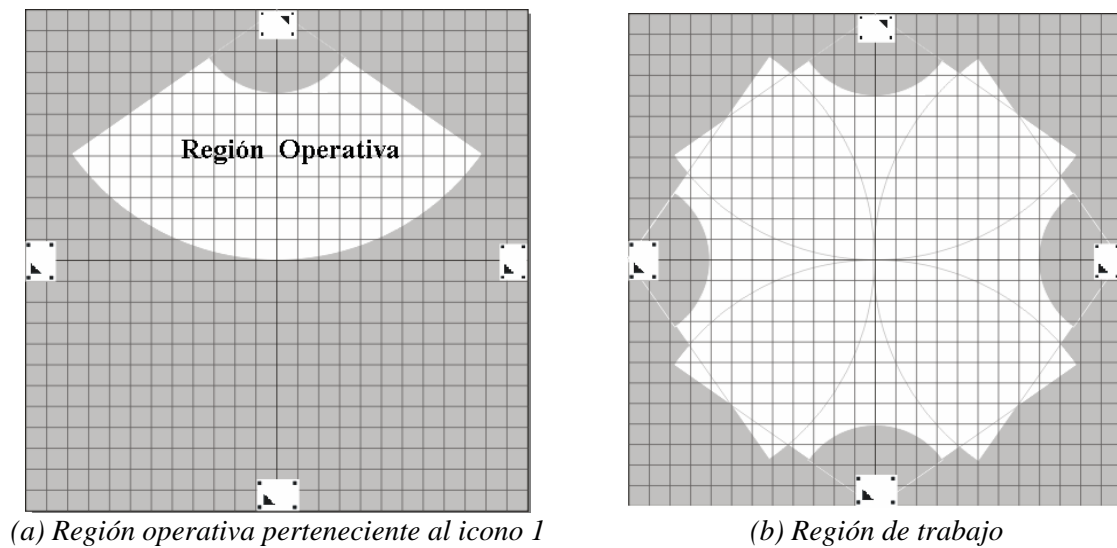


Figura 2.3. Vista superior del espacio de trabajo

Además de los parámetros anteriores también resultó necesario el obtener la altura mínima que debe tener la maqueta para que al adquirir las imágenes no se presenten agentes externos al entorno de trabajo, los cuales actúan en forma de ruido, ocasionando fallas en el procesamiento de las mismas, dicha altura resultó conveniente establecerla de 50[cm].

Tomando en cuenta lo antes mencionado, las dimensiones resultantes propuestas para la construcción de la maqueta de pruebas serán: 120[cm] de largo, 120[cm] de ancho y 50[cm] de alto.

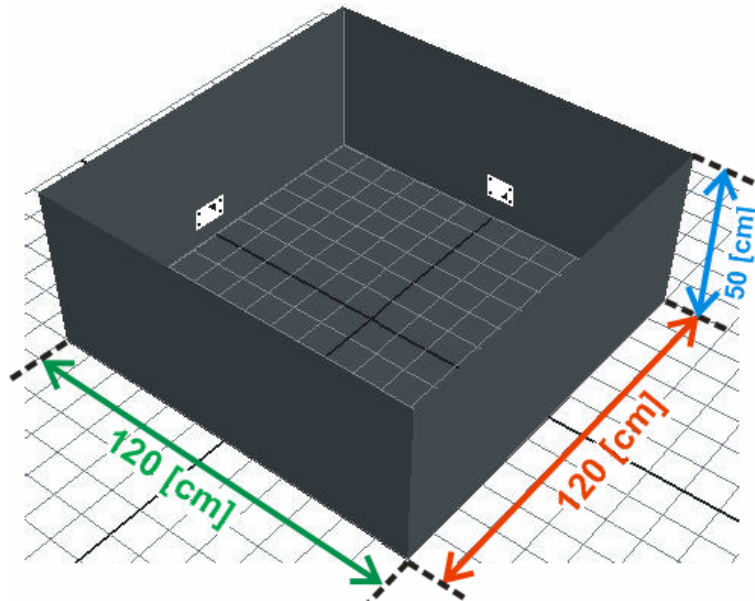


Figura 2.4. Medidas del espacio de trabajo

2.4.4 Especificaciones para la programación.

Las tareas que se espera desarrolle el sistema en conjunto con la cámara serán las siguientes:

- Encontrar la posición de referencia (HOME) al iniciarse el sistema.
- Después de colocar la cámara en una posición válida dentro de la región de trabajo, y una vez que se le da el orden de inicio al sistema, este comenzará a controlar el movimiento de panning de la cámara, realizando entre cada movimiento la búsqueda de un icono que se encuentre a una distancia válida, es decir, más de 20[cm] y menos de 60[cm].
- Una vez reconocido el icono, el sistema debe realizar los movimientos necesarios de la cámara para centrarlo y reconocer cual de los cuatro iconos es el que se encuentra en la escena.
- Con el icono centrado, se obtiene la distancia del lente de la cámara al centro del icono y el ángulo de visión.
- Una vez obtenidos los parámetros anteriores se obtendrán, por medio de un modelo matemático, las coordenadas (x , y) de la posición de la cámara dentro del entorno.

- Por ultimo se obtiene una representación gráfica de la posición de la cámara dentro del entorno.

Adicionalmente, los objetivos generales del sistema se centrarán en los siguientes puntos:

- Ser de fácil uso.
- Tener una interfaz agradable para el usuario.
- Permitir la calibración de este, para un posible cambio de la base iconográfica o una posible expansión del espacio de trabajo.

Analizando esta sección decidimos que la solución más factible para la implementación del sistema es desarrollarlo en el lenguaje de programación *Microsoft Visual Basic 6.0*, ya que este es un lenguaje muy potente, fácil de usar y además nos brinda la posibilidad de interactuar con Matlab, herramienta seleccionada para llevar a cabo el procesamiento de las imágenes, de esta manera lograremos cubrir los objetivos planteados de una manera rápida y eficiente.

Finalmente, una vez que se ha revisado de forma general el problema con sus limitantes y requerimientos, es necesario comenzar con la siguiente fase del proyecto, sin embargo, antes de continuar es importante establecer una terminología adecuada, referente a ciertos aspectos de este proyecto, los cuales en lo sucesivo serán ampliamente empleados.

Terminología

Espacio de trabajo:

Al utilizar este término hacemos referencia al espacio diseñado e implementado, en el cual se llevan a cabo las etapas de calibración y pruebas del sistema, y este corresponde a la maqueta de dimensiones: 120 [cm] x 120 [cm] x 50 [cm].

Región operativa:

Este término es empleado para referirse a la región que se forma al establecer las limitantes de distancia mínima y máxima, así como el ángulo máximo, valores para los cuales el sistema de visión desarrollado trabaja de manera correcta, aplicables a cada uno de los

cuatro iconos. Dicha región queda formada en su totalidad por dos secciones (izquierda y derecha), delimitadas por la perpendicular que atraviesa el centro de cada icono.

Región de trabajo:

Este término se refiere a una región dentro del espacio de trabajo, la cual se forma al establecer la región operativa de cada icono, y está compuesta por la intersección de las cuatro regiones resultantes.

Ángulo de visión:

El ángulo de visión se forma cuando la cámara tiene a un respectivo icono centrado en la escena que está captando, y es el ángulo existente entre la línea imaginaria que representa al eje óptico y la línea imaginaria que representa a la perpendicular que atraviesa el centro del mismo icono.

Capítulo 3

Diseño e implementación

Introducción

En este capítulo se presentan la etapa de diseño y la de implementación del sistema SLOAC (*Sistema de Localización de Objetos en Ambientes Cerrados*) tratado en esta tesis, el cual decidimos dividir para su diseño en dos partes, la parte de *software* y la parte de *hardware*.

- La parte de *software* comprende el modelado general del sistema, el diseño de los algoritmos utilizados y la implementación de la arquitectura del SLOAC.
- En cuanto a *hardware* fue necesario utilizar un controlador de apoyo, para el cual se muestra su diseño e implementación al final de este capítulo.

En la figura 3.1 se ilustra la arquitectura completa del sistema.

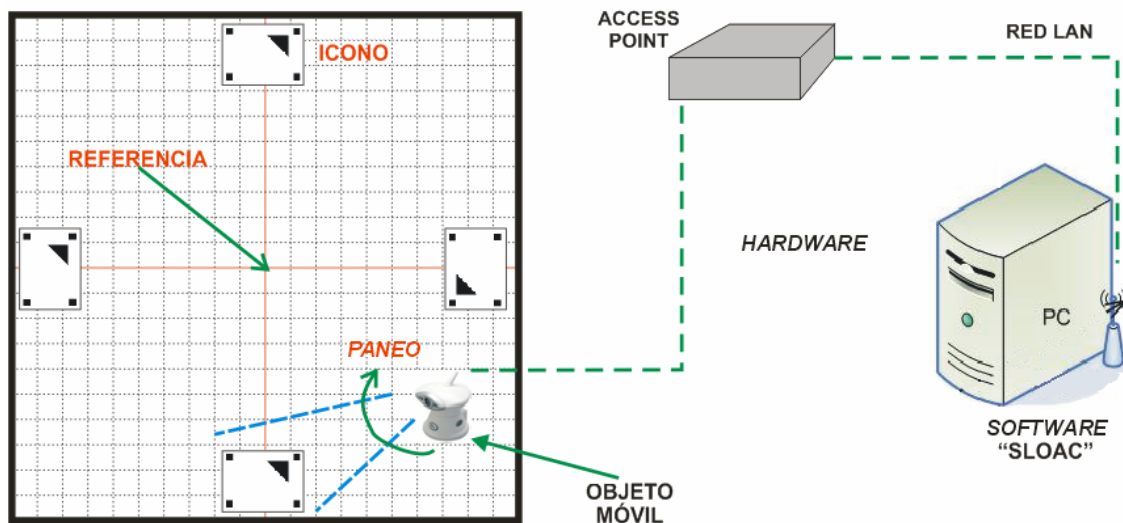


Figura 3.1. Arquitectura del Sistema

3.1 Software

3.1.1 Modelado del sistema

El modelado consiste en representar de manera lógica al sistema para facilitar su posterior programación.

El sistema, como se describió anteriormente, tiene como propósito obtener las coordenadas (dadas con respecto a un punto de referencia) de una cámara de video, dentro de un espacio de trabajo, además de que también puede ser recalibrado para trabajar en un

espacio de trabajo de mayores dimensiones, y se conforma de cuatro módulos, los cuales se muestran en la figura 3.2, la cual ilustra su interacción.

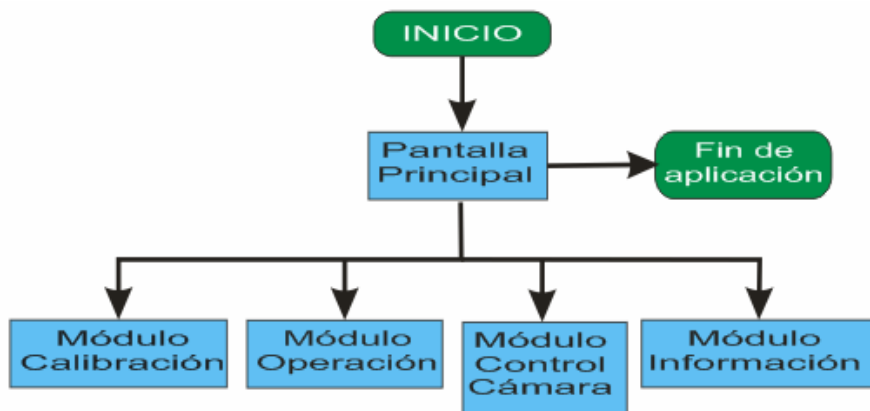


Figura 3.2. Diagrama General del Sistema

Módulo - Calibración

El modulo de calibración se encuentra dentro de la pantalla principal en el menú *calibración*, sirve para introducir los parámetros con los cuales va a trabajar el sistema, los que se mencionan a continuación:

- Límites. Abre una ventana, donde aparecen tres botones: foto de cerca, foto de lejos y guardar. Con los primeros dos se toman las fotos de los límites superior e inferior que delimitan la región operativa de un icono y una vez obtenidas estas imágenes se presiona el último botón que guarda la información que se obtiene de las imágenes para establecer los parámetros.
- Parámetros. Esta acción abre una ventana donde se introducen los datos de las características generales del espacio de trabajo necesarias para el entrenamiento del sistema (distancia mínima y máxima al icono, distancia entre curvas, ángulo máximo y la resolución en grados).
- Coordenadas iconos. Abre una ventana donde el usuario establece las coordenadas a las cuales se encuentran los iconos del sistema con respecto a nuestro sistema de referencia.
- Entrenamiento. Con la información que se obtuvo de las ventanas anteriores, esta acción se encarga de adquirir imágenes de la cámara en distintas posiciones (como se explica más adelante en el ejemplo 1 de la sección 3.1.5), procesarlas y obtener

los parámetros apropiados para poder determinar posteriormente las coordenadas de la cámara dentro del espacio de trabajo.

- Ecuación Final. Después de haber realizado el entrenamiento del sistema para cada una de las curvas previamente definidas, esta acción nos permite obtener la ecuación final por medio de la cual el sistema va a realizar el cálculo para obtener la posición de la cámara dentro del espacio de trabajo.

Módulo - Control Cámara.

Este módulo se compone de varios elementos mediante los cuales se realizan todas las funciones de control de la cámara los cuales se describen a continuación:

- Acceso a la cámara. Este se realiza en la pantalla inicial, al presionar el botón *Login* nos abre una ventana donde nos pide usuario y contraseña que nos permiten acceder al sistema de la cámara para su posterior manipulación.
- Manipulación de movimiento (pan/tilt). Esta se realiza en dos partes de nuestro sistema, la primera es con el control que se encuentra en ella que nos permite moverla hacia arriba, abajo, lado izquierdo ó derecho con tal solo presionar el botón de la flecha que nos indique la dirección hacia la cual deseamos mover la cámara. La segunda parte es el movimiento que realiza la cámara al presionar el botón *Encuentra icono* mediante el cual la cámara realiza un movimiento hacia la derecha hasta encontrar un icono en la región de trabajo.
- Adquisición de imágenes. Es la función de la cámara que nos permite obtener imágenes para su posterior procesamiento e interpretación de resultados mediante los cuales se toman decisiones para el funcionamiento del sistema. Esta se utiliza en el módulo de calibración del sistema como se mencionó anteriormente y también se utiliza durante la búsqueda del icono, la cámara realiza un movimiento a la derecha, el sistema adquiere una imagen y decide si la información que se obtiene de esta nos permite obtener la posición de la cámara en el espacio de trabajo, en caso contrario continua con la búsqueda.

Módulo - Operación

Mediante éste módulo se ejecutan las funciones adecuadas para la obtención de la posición de la cámara en el entorno y el despliegue gráfico de resultados dentro de nuestra

pantalla principal. Para esto se ocupan varios algoritmos que son descritos más adelante en las secciones 3.1.2 a 3.1.6 de este capítulo.

Módulo - Información

En éste modulo se presenta información útil para el usuario, como un archivo de ayuda donde se describe a grandes rasgos el sistema, un tutorial que muestra el funcionamiento general del sistema, así como información de los distintos parámetros previamente establecidos en el modulo de calibración.

En conjunto con el diseño de los módulos, surgió la necesidad de diseñar e implementar algunos algoritmos, los cuales son la base del funcionamiento del sistema y se describen a continuación.

3.1.2 Detección del icono

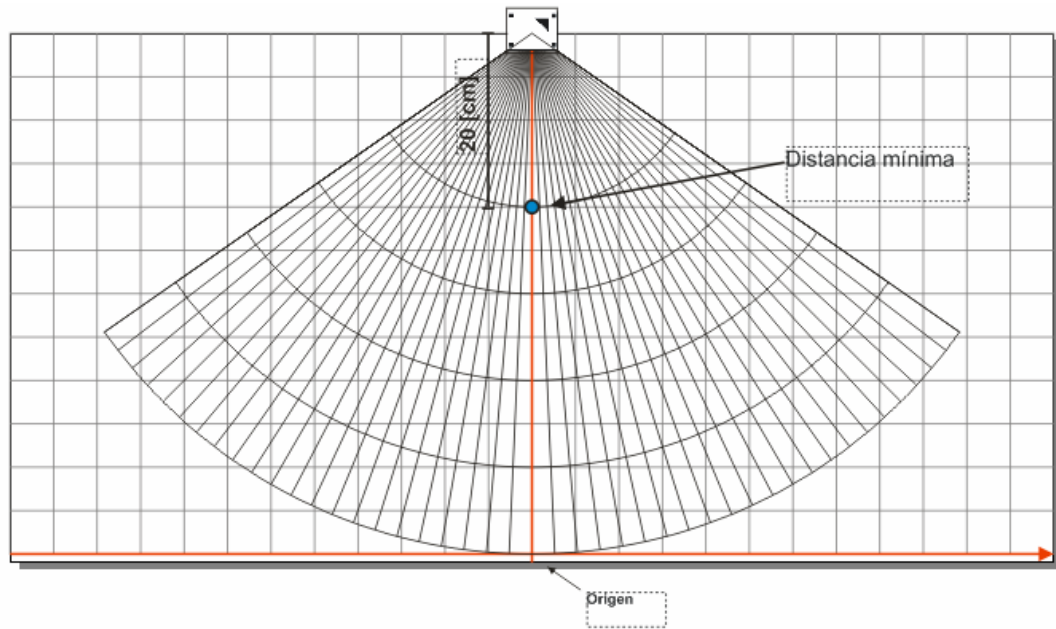
Para poder detectar si existe o no uno de los cuatro iconos en una escena captada dentro del espacio de trabajo, decidimos hacer uso de las distancias mínima y máxima que definen la región operativa de cada uno de ellos, esto con la finalidad de que si el icono es captado desde algún punto fuera de esta no sea tomado en cuenta.

El procedimiento para la detección del icono consistió primeramente en seleccionar un descriptor de regiones que se ajustara a nuestras necesidades, es decir, tenia que ser un descriptor de calculo rápido y que además sacara provecho de las condiciones que se plantearon para el espacio de trabajo (Icono en color negro sobre fondo blanco), por lo cual el descriptor que decidimos aplicar es el que se conoce como “ $p2a$ ”, el cual utiliza el perímetro y el área de un objeto en color negro sobre fondo blanco de una imagen binaria, dicho descriptor está dado por la siguiente formula.

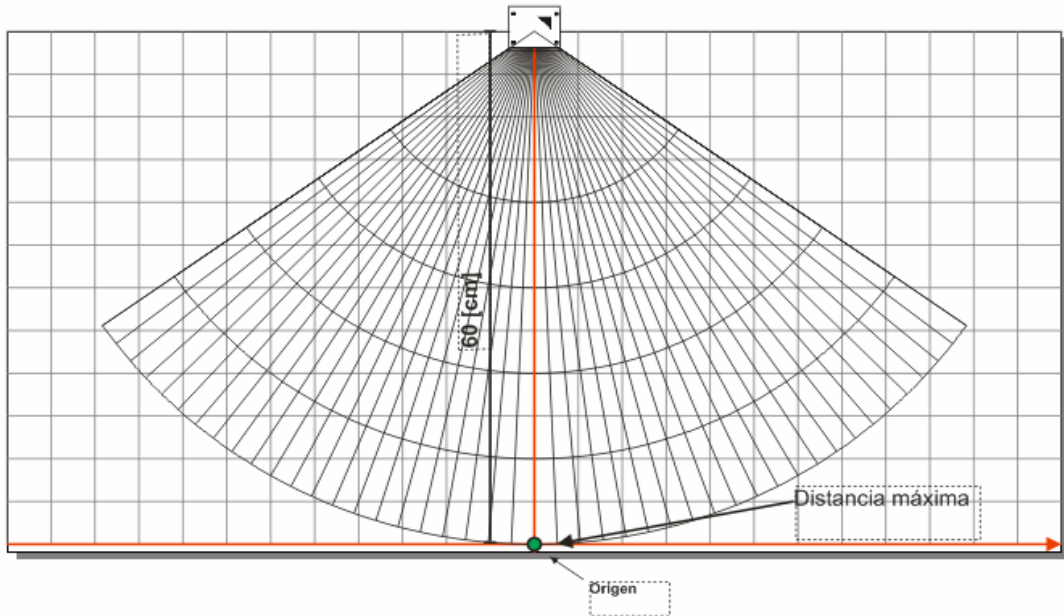
$$p2a = \frac{(\text{perímetro})^2}{\text{área}} \quad \dots(\text{I})$$

Una vez que definimos el descriptor el siguiente paso fue obtener con él, los valores máximo y mínimo dentro de una de las regiones operativas, para lo cual se obtuvieron dos imágenes de este, la primera colocando la cámara a la distancia mínima (*figura 3.3 (a)*)

dentro de la región operativa y la segunda colocando la cámara a la distancia máxima (figura 3.3 (b)) dentro de la misma región.



(a) Vista superior de la región operativa ubicando la posición de la cámara a 20 [cm]



(b) Vista superior de la región operativa ubicando la posición de la cámara a 60 [cm]

Figura 3.3. Obtención del intervalo de valores del descriptor $p2a$

Con este procedimiento se obtuvieron los siguientes valores de $p2a$:

$$p2a_{\min} = 51.319$$

$$p2a_{\max} = 69.566$$

Todo lo anterior tiene el objetivo de establecer los límites superior e inferior, los cuales le permiten saber al sistema, en tiempo real, si en la escena que está captando la cámara se encuentra presente uno de los cuatro iconos y además si este debe o no ser tomado en cuenta.

3.1.3 Identificación del icono

Como es sabido, la función principal de la base iconográfica es proporcionarnos los parámetros suficientes para poder determinar la distancia que existe de la cámara a un icono, el ángulo de visión y por ultimo la región del espacio de trabajo en la que se encuentra la cámara, por estas razones resulta necesario dotar al sistema de tres algoritmos, uno que sea capaz de obtener la distancia, otro el ángulo y por ultimo uno que sea capaz de identificar cual de los cuatro iconos se encuentra presente en la escena, hasta aquí es importante mencionar que para lograr dichos objetivos se requiere que en la imagen de la escena captada por la cámara, el icono se encuentre centrado, con el fin de que este sea apreciado en su totalidad y que al aplicarle las ecuaciones correspondientes, los valores obtenidos sean precisos.

El algoritmo de identificación desarrollado aplica un proceso de binarización y etiquetado sobre la imagen para diferenciar los elementos que componen a cada icono ya que el procedimiento de identificación consiste en obtener los centroides de cada objeto para posteriormente calcular por medio de la ecuación (II), las distancias existentes entre el centroide del elemento tres (triángulo) y cada uno de los centroides de los otros cuatro elementos (cuadros) ver figura 3.4.

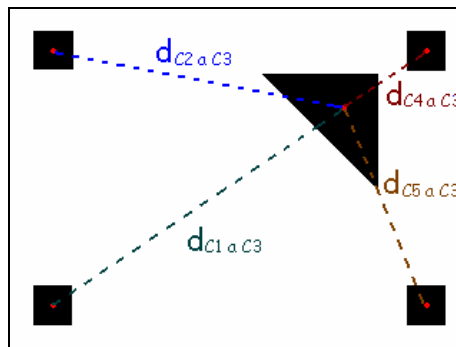


Figura 3.4. Representación de las distancias entre centroides

Ecuación de distancia entre dos puntos:

$$d_{\overline{AB}} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \quad \dots(\text{II})$$

Una vez obtenidas estas cuatro distancias el algoritmo las compara y determina cual es la menor. Esto sirve para determinar gráficamente hacia que cuadrado del icono está más cercano el triángulo y debido al diseño de los iconos nos permite identificar de cual de ellos se trata (tabla 3.1), lo que implica a su vez, que podemos determinar en que cuadrante del espacio de trabajo se encuentra la cámara, debido a que cada icono se encuentra ubicado exactamente en la división de dos cuadrantes y el icono únicamente puede ser captado desde estos dos cuadrantes.

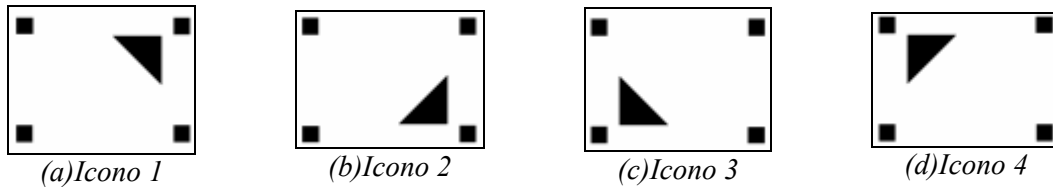


Figura 3.5. Base Iconográfica

Distancia Menor	Icono Reconocido
d_{C4aC3}	Icono 1
d_{C5aC3}	Icono 2
d_{C1aC3}	Icono 3
d_{C2aC3}	Icono 4

Tabla 3.1.

3.1.4 Cálculo de la distancia al icono

Para obtener la distancia que existe entre el lente de la cámara y el centro del icono, hacemos uso de la siguiente formula¹.

¹ Formula obtenida del artículo “Determinación de la posición de un objetivo por análisis de perspectiva, aplicada a un robot basketbolista”

$$d = \frac{k}{\text{alto aparente}} \quad \dots(1.1)$$

$$k = d \times \text{alto aparente} \quad \dots(1.2)$$

donde:

d = distancia

k = constante de proporcionalidad

alto aparente = altura del icono (en pixeles)

El procedimiento consiste en obtener la distancia por medio del análisis de perspectiva, en el cual se relacionan las magnitudes de los objetos con la distancia a la que son tomados, ya que las magnitudes aparentan ser menores a mayor distancia y viceversa, esta relación esta dada por la ecuación (1.1).

Para hacer uso de la ecuación (1.1) el primer paso a seguir fue realizar un proceso de etiquetado (ver figura 3.6), con el cual se tienen bien identificados cada uno de los elementos presentes en el icono.

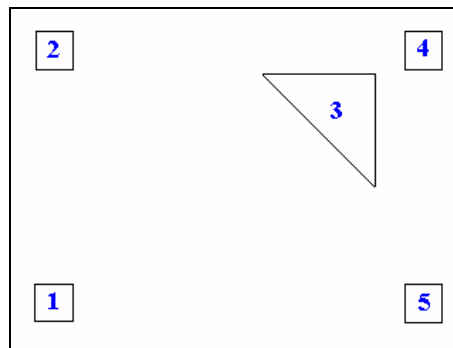


Figura 3.6. Identificación de los objetos en la imagen, después de etiquetarla

Posteriormente se calcularon los centroides de cada elemento con el objetivo de poder obtener dos puntos, uno correspondiente al punto medio que existe entre el centroide del elemento 2 y el centroide del elemento 4 y el segundo perteneciente al punto medio entre el centroide del elemento 1 y el centroide del elemento 5, nótese que para el calculo de la distancia el centroide del elemento 3 (triángulo) no es utilizado y que para obtener los puntos anteriores se hizo uso de las ecuaciones (2 y 3), una vez obtenidos estos puntos se calcula la magnitud que existe entre ellos (denominada “*alto aparente*”) por medio de la ecuación 4 (ver figura 3.7).

$$PM1_x = \frac{x_{C2} + x_{C4}}{2} \quad \dots(2)$$

$$PM1_y = \frac{y_{C2} + y_{C4}}{2} \quad \text{donde:}$$

$PM1_x, PM1_y$ corresponden a las coordenadas (x,y) del punto medio entre los centroides 2 y 4
 x_{C2}, y_{C2} son las coordenadas (x,y) del centroide 2
 x_{C4}, y_{C4} son las coordenadas (x,y) del centroide 4

$$PM2_x = \frac{x_{C1} + x_{C5}}{2} \quad \dots(3)$$

$$PM2_y = \frac{y_{C1} + y_{C5}}{2} \quad \text{donde:}$$

$PM2_x, PM2_y$ corresponden a las coordenadas (x,y) del punto medio entre los centroides 1 y 5
 x_{C1}, y_{C1} son las coordenadas (x,y) del centroide 1
 x_{C5}, y_{C5} son las coordenadas (x,y) del centroide 5

Ecuación utilizada para el cálculo del alto aparente.

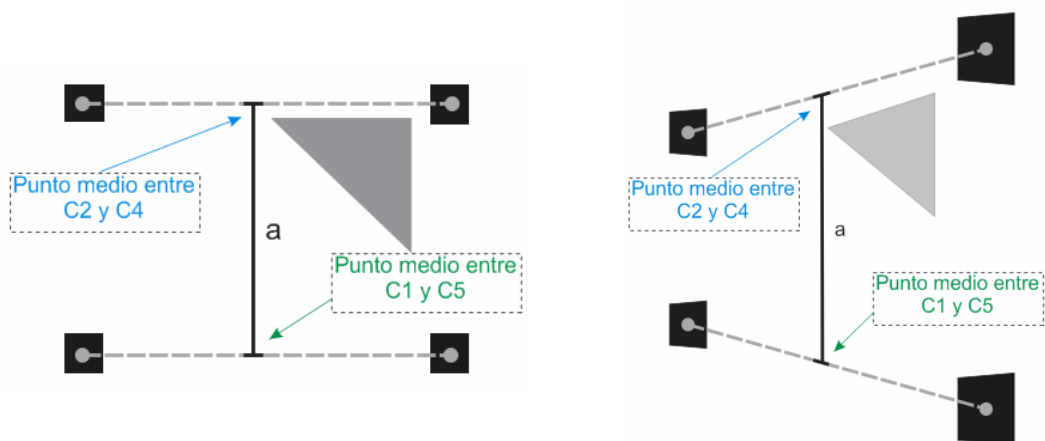
$$a = \sqrt{(PM2_x - PM1_x)^2 + (PM2_y - PM1_y)^2} \quad \dots(4)$$

donde:

a alto aparente

$PM1_x, PM1_y$ corresponden a las coordenadas (x,y) del punto medio entre los centroides 2 y 4

$PM2_x, PM2_y$ corresponden a las coordenadas (x,y) del punto medio entre los centroides 1 y 5



(a) Imagen Original del icono

(b) Imagen del icono tomada a 55° con respecto a la perpendicular que lo atraviesa

Figura 3.7. Vista del icono a 0° y 55°.

Hasta este punto solo hemos obtenido un parámetro de la ecuación (1.1), por lo que el siguiente paso es obtener la constante de proporcionalidad k , la cual depende de las características del lente de cada cámara y de cada escena, por lo tanto la mejor manera de obtenerla es por medio de pruebas de laboratorio.

Las pruebas realizadas para la obtención de “ k ” se llevaron a cabo de la siguiente manera:

1. Colocamos la cámara a una distancia conocida “ d ” de tal forma que se pudiera obtener una imagen centrada del icono.
2. Con la imagen adquirida obtuvimos el parámetro “*alto aparente*”.
3. Se sustituyeron los valores “ d ” y “*alto aparente*” en la ecuación (1.2) para obtener una constante de proporcionalidad “ k_i ”.
4. Se repitieron los pasos 1 a 3 para 5 distancias diferentes pertenecientes a la región operativa de un icono (ver tabla 1).
5. Se calculó el promedio de las “ k_i ” con el objetivo de encontrar una constante “ k ” que nos permitiera obtener el valor de “ d ” en cualquier punto dentro de la región operativa de cada icono.

Distancia [cm]	K_i	<i>Alto aparente</i> [píxeles]
20	2985.8	149
30	2990.3	100
40	3007.9	75
50	3000.3	60
60	3001.7	50

$$K_{promedio} = 2997.2$$

Tabla 3.2

3.1.5 Cálculo del ángulo de visión

Para la obtención del ángulo de visión mediante este algoritmo, fue necesario realizar una calibración, la cual consistió en obtener el ángulo de visión real dentro de una de las regiones operativas por medio de parámetros visuales extraídos de los iconos, dichos parámetros se explican a continuación:

Ángulo Θ . Es el ángulo que se forma al superponer dos de los vértices de un triángulo rectángulo imaginario sobre los centroides de los cuadrados 2 y 4 del icono (figura 3.8.a).

Relación entre magnitudes. Es la relación que existe entre las magnitudes que se ilustran en la (figura 3.8.b). Es importante mencionar que estas magnitudes también sirven para detectar el lado en el que se encuentra la cámara con respecto al icono. Para obtener este parámetro se utiliza la siguiente formula.

$$relmag = \frac{a}{b} \begin{cases} a = m1, b = m2 & \text{si } m1 \geq m2 & \text{Lado derecho} \\ a = m2, b = m1 & \text{si } m2 > m1 & \text{Lado izquierdo} \end{cases}$$

donde:

$relmag$ = relación entre magnitudes

$m1$ = magnitud C1 a C2

$m2$ = magnitud C4 a C5

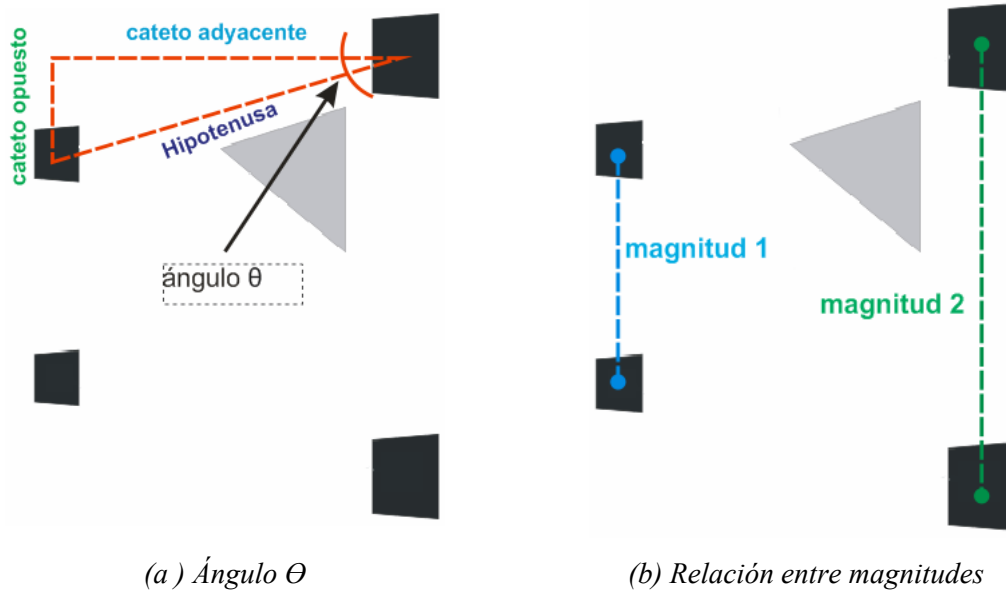


Figura 3.8. Parámetros a extraer de un icono

Tomando en cuenta la región operativa de cada icono se decidió que para la obtención del ángulo de visión se tenía que dividir esta región con curvas a una cierta distancia al icono y con rectas representativas de dicho ángulo (figura 3.9).

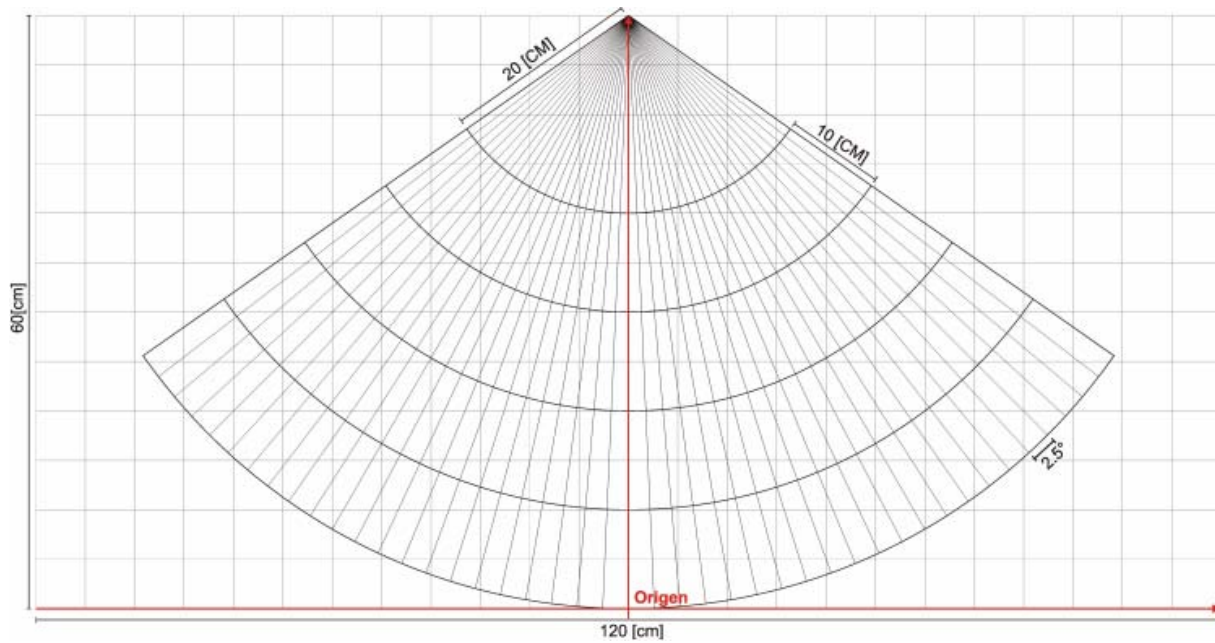


Figura 3.9. Vista superior de la región operativa dividida

Para obtener los valores más adecuados se estableció que la distancia entre estas curvas debía ser de 10 [cm] y que el ángulo entre las rectas fuera de 2.5°. Una vez establecidas estas condiciones, el siguiente paso fue realizar una calibración del sistema sobre la región operativa del icono 1, con el objetivo de establecer los parámetros generales (ecuaciones de la forma $y=m\chi+b$) que caracterizan dicha región.

Los parámetros generales tienen la función principal de proporcionarnos el ángulo de visión en el que se encuentra la cámara dentro de la región operativa, esto por medio de la extracción de los parámetros visuales anteriormente descritos.

Para ilustrar el proceso de calibración y el funcionamiento del algoritmo se presenta el ejemplo 1, el cual comprende únicamente la calibración a lo largo de una curva (a 30 [cm] del icono), ya que el procedimiento para calibrar completamente el sistema es iterativo.

Ejemplo 1.

Una vez que la región operativa fue acotada como se mostró en la figura 3.9, el primer paso para calibrar el sistema a una distancia de 30 [cm], existente entre el lente de la cámara y el centro del icono, consiste en colocar la cámara en cada uno de los puntos que muestra la figura 3.9, correspondientes a puntos equidistantes del centro del icono y con un ángulo de separación entre ellos de 2.5° , para posteriormente tomar una imagen del icono en cada uno de los puntos.

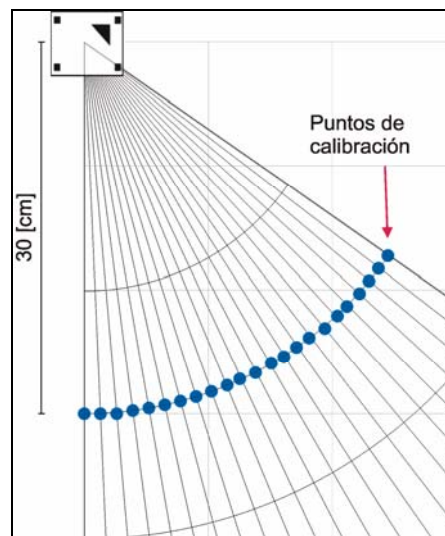


Figura 3.10. Vista superior que ilustra la posición de los puntos de calibración

El siguiente paso es tomar cada una de las imágenes, obtener el Ángulo Θ y relacionarlo con el ángulo real (ángulo Φ) al que fue tomada la imagen para que por medio de una regresión lineal (Mínimos Cuadrados ver apéndice B) obtengamos una ecuación que nos permita obtener el ángulo Φ con solo saber el ángulo Θ , como se ilustra a continuación:

ángulo Θ	ángulo Φ
0.4769	0
0.6459	2.5
1.0584	5
1.3982	7.5
1.5287	10
1.8862	12.5
2.5682	15
2.7678	17.5
3.0986	20
3.5197	22.5
3.8516	25
4.3241	27.5
4.7523	30
5.1966	32.5
5.8994	35
6.5067	37.5
7.1515	40
7.7201	42.5
8.4479	45
9.0083	47.5
9.8049	50
10.905	52.5
11.951	55

Tabla 3.3

Al aplicar la regresión lineal se obtiene una ecuación de la forma:

$$\text{angulo } \phi = m(\text{angulo } \theta) + b \quad \dots(\text{III})$$

donde:

angulo ϕ : es el ángulo real

al que fue tomada la imagen

angulo θ : es el ángulo obtenido de la imagen

m: es la pendiente de la recta

b: es la ordenada al origen de la recta

Ecuación obtenida:

$$\text{angulo } \phi = 4.4643 \times \text{angulo } \theta + 4.7445 \quad \dots(\text{III.1})$$

Al obtener la ecuación (III.1), el siguiente paso fue tomar otras tres imágenes al azar, sobre la misma curva y obtener el valor del ángulo Θ para posteriormente sustituirlo en (III.1) y comprobar el cumplimiento de la relación, sin embargo, los valores obtenidos no resultaron como se esperaba, esto debido a que los lentes de las cámaras presentan algunas aberraciones ópticas que influyen en la imagen, por lo tanto tuvimos que hacer un análisis para encontrar la manera de cuantificar el error ocasionado por dichas aberraciones, encontrando que podíamos manipular la ecuación (III.1) de tal forma que se obtuvieran los valores que compensaran ese error, logrando con esto la siguiente ecuación:

Despejando de (III.1) la variable “ángulo Θ ” se tiene:

$$\text{angulo } \theta = \frac{\text{angulo } \phi - 4.7445}{4.4643} \quad \dots(\text{III.2a})$$

Haciendo un cambio de variable

$$\theta_{\text{ideal}} = \frac{\text{angulo } \phi - 4.7445}{4.4643} \quad \dots(\text{III.2b})$$

Al obtener la ecuación (III.2b) estamos cuantificando el error, ya que si sustituimos los valores del ángulo Φ estaremos obteniendo los valores de Θ_{ideal} , es decir, los valores que se requieren para ser valuados en la ecuación (III.1) y que nos proporcionan el valor real del ángulo Φ al cual fue tomada la imagen.

Ahora contamos con una ecuación y la cuantificación del error, pero para poder obtener la cuantificación del error a partir de la imagen captada del icono, necesitamos relacionar los valores de la cuantificación con otro parámetro, por lo que ahora hacemos uso de la relación entre magnitudes. De igual manera que se obtuvo la ecuación (III.1) ahora se relaciona Θ_{ideal} con la relación entre magnitudes, por medio de una regresión lineal, para obtener una segunda ecuación como se ilustra a continuación.

<i>relmag</i>	Θ_{ideal}
0.997	-1.0627646
1.0075	-0.50276639
1.027	0.05723182
1.0409	0.61723002
1.0501	1.17722823
1.0645	1.73722644
1.0941	2.29722465
1.1027	2.85722286
1.1148	3.41722106
1.1355	3.97721927
1.1498	4.53721748
1.1643	5.09721569
1.1747	5.6572139
1.1918	6.2172121
1.2095	6.77721031
1.2275	7.33720852
1.247	7.89720673
1.261	8.45720494
1.2734	9.01720314
1.2847	9.57720135
1.2912	10.1371996
1.3161	10.6971978
1.3223	11.257196

Al aplicar la regresión lineal se obtiene una ecuación de la forma:

$$\theta_{ideal} = m(\text{relmag}) + b \quad \dots(\text{IV})$$

donde:

θ_{ideal} : es el ángulo que satisface la ecuación III.1

relmag: es la relación entre magnitudes, obtenida de la imagen

m: es la pendiente de la recta

b: es la ordenada al origen de la recta

Ecuación obtenida:

$$\theta_{ideal} = 37.195 \times \text{relmag} - 38.129 \quad \dots(\text{IV.1})$$

Tabla 3.4

Una vez obtenidas las ecuaciones III.1 y IV.1 el proceso de calibración para la curva en cuestión queda completo y para poder obtener el ángulo sobre cualquier punto de la curva se procede de la siguiente manera:

1. Colocamos la cámara sobre cualquier punto perteneciente a la curva
2. Obtenemos una imagen en la cual el icono se encuentre centrado
3. Se procesa para la obtención del valore de *relmag*
4. Se sustituye el valor de *relmag* en la ecuación IV.1 para obtener el valor de Θ_{ideal}
5. Se sustituye el valor de Θ_{ideal} en la ecuación III.1 obteniéndose con ello valor del ángulo real al cual fue obtenida la imagen

Para calibrar la parte derecha de la región operativa del icono 1 se realizaron los pasos descritos en el ejemplo anterior en cada una de las curvas que comprende dicha región (curvas a 20, 30, 40, 50 y 60 [cm]), con lo cual se obtuvieron un total de diez ecuaciones (2 por curva), finalmente se tomo la decisión de implementar dos grupos de ecuaciones llamados GI y GII, los cuales son generadores de las diez ecuaciones anteriores. El grupo GI se encarga de generar ecuaciones del tipo (III.1) y el grupo GII se encarga de generar ecuaciones del tipo (IV.1).

Para obtener GI y GII se decidió relacionar los elementos de las ecuaciones de cada curva con su respectiva distancia.

Para GI:

Relacionando la distancia con las pendientes de las ecuaciones del tipo III.1

Distancia	Pendientes(III.1)
20	3.3392
30	4.7668
40	6.2487
50	7.9983
60	9.0485

Al aplicar la regresión lineal se obtiene la siguiente ecuación:

$$m_{principal} = 1.465 \times \text{Distancia} + 1.8853$$

Tabla 3.5

Relacionando la distancia con las ordenadas de las ecuaciones del tipo III.1

Distancia	Ordenadas (III.1)
20	4.2708
30	3.8261
40	3.4671
50	3.4388
60	3.4422

Al aplicar la regresión lineal se obtiene la siguiente ecuación:

$$b_{principal} = -0.2044 \times \text{Distancia} + 4.3024$$

Tabla 3.6

Replanteando la ecuación III en términos de las ecuaciones de GI:

$$\text{angulo}\phi = m_{\text{principal}} \times \theta_{\text{ideal}} + b_{\text{principal}} \quad \dots(\text{V})$$

Para GII:

Relacionando la distancia con las pendientes de las ecuaciones del tipo IV.1

Distancia	Pendientes(IV.1)
20	30.106
30	33.8469
40	35.9072
50	34.0587
60	36.3423

Al aplicar la regresión lineal se obtiene la siguiente ecuación:

$$m_{\text{error}} = 1.2684 \times \text{Distancia} + 30.247$$

Tabla 3.7

Relacionando la distancia con las ordenadas de las ecuaciones del tipo IV.1

Distancia	Ordenadas (IV.1)
20	-30.985
30	-34.5366
40	-36.4428
50	-34.3096
60	-36.4328

Al aplicar la regresión lineal se obtiene la siguiente ecuación:

$$b_{\text{error}} = -1.0669 \times \text{Distancia} - 31.341$$

Tabla 3.8

Replanteando la ecuación IV en términos de las ecuaciones de GII:

$$\theta_{\text{ideal}} = m_{\text{error}} \times \text{relmag} + b_{\text{error}} \quad \dots(\text{VI})$$

Para obtener el ángulo de visión en cualquier punto de la región operativa del icono 1 se sigue el siguiente procedimiento.

1. Colocamos la cámara en cualquier punto de la región, de tal forma que se pueda obtener una imagen en la que el icono se encuentre centrado dentro de esta.
2. Tomamos la imagen
3. A la imagen adquirida se le aplica el algoritmo “Obtención de distancia”
4. Se evalúa el valor de “Distancia” en las ecuaciones de GI y GII
5. Obtenemos el valor de “relmag” de la imagen

6. Evaluamos el valor de “ $relmag$ ” en la ecuación (VI) para obtener “ θ_{ideal} ”
7. Por ultimo sustituimos “ θ_{ideal} ” en la ecuación (V)

Ahora que se tienen los grupos de ecuaciones GI y GII tenemos la forma de obtener el ángulo de visión real al que se adquiere una imagen del icono 1 en cualquier punto perteneciente a la mitad derecha de la región operativa de este, sin embargo, como dicha región es simétrica y los parámetros visuales extraídos del icono en una imagen tomada en la parte izquierda de la región son iguales en magnitud a los extraídos de una imagen tomada en la parte derecha, también podemos obtener el ángulo de visión real de cualquier punto perteneciente a la parte izquierda de la región operativa del icono 1 y por lo tanto el ángulo de visión de cualquier punto perteneciente a toda la región operativa de dicho icono.

También podemos decir que las ecuaciones obtenidas para la región operativa del icono1 son aplicables para las tres regiones restantes, ya que los iconos no cambian en su dimensión y por lo tanto los parámetros visuales extraídos de estos son los mismos.

En conclusión, podemos obtener el ángulo de visión para cualquier punto perteneciente a la región de trabajo con solo realizar la calibración antes descrita debido a la configuración que utilizamos para el posicionamiento de los iconos y a las características físicas de estos.

3.1.6 Obtención de la posición de la cámara

Para poder obtener la posición en la cual se encuentra la cámara dentro de la región de trabajo es necesario obtener los siguientes tres parámetros:

1. Identificar el icono
2. Obtener la distancia que existe entre el icono y la cámara
3. Obtener el ángulo de visión

El primer paso para determinar las coordenadas de la cámara es hacer uso de la distancia y del ángulo para obtener el vector de posición de esta con respecto al icono, como se muestra en la figura 3.11.

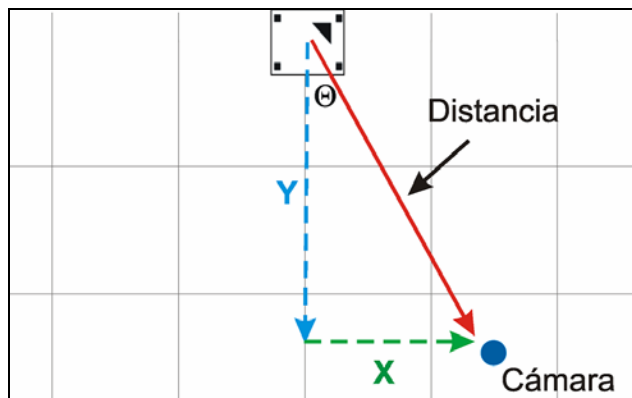


Figura 3.11. Obtención del vector de posición

Para poder obtener los vectores X, Y se hace uso de las siguientes formulas.

$$X = \text{Distancia} \times \sin \theta$$

$$Y = \text{Distancia} \times \cos \theta$$

Como el vector de posición obtenido es relativo a cada uno de los iconos, y las coordenadas con respecto a nuestro sistema de referencia de estos son conocidas, es posible obtener las coordenadas de la cámara con respecto al sistema de referencia, para esto primero es necesario analizar si la cámara se encuentra del lado derecho o izquierdo del icono en cuestión y posteriormente se aplican las siguientes formulas:

Lado derecho

Icono 1	Icono 2	Icono 3	Icono 4
$X_{Cámara} = X_{Icono1} + X$	$X_{Cámara} = X_{Icono2} - X$	$X_{Cámara} = X_{Icono3} - X$	$X_{Cámara} = X_{Icono1} + X$
$Y_{Cámara} = Y_{Icono1} - Y$	$Y_{Cámara} = Y_{Icono2} - Y$	$Y_{Cámara} = Y_{Icono3} + Y$	$Y_{Cámara} = Y_{Icono1} + Y$

Lado izquierdo

Icono 1	Icono 2	Icono 3	Icono 4
$X_{Cámara} = X_{Icono1} - X$	$X_{Cámara} = X_{Icono2} - X$	$X_{Cámara} = X_{Icono3} + X$	$X_{Cámara} = X_{Icono1} + X$
$Y_{Cámara} = Y_{Icono1} - Y$	$Y_{Cámara} = Y_{Icono2} + Y$	$Y_{Cámara} = Y_{Icono3} + Y$	$Y_{Cámara} = Y_{Icono1} - Y$

La siguiente figura ilustra la posición de la cámara con respecto a nuestro sistema de referencia.

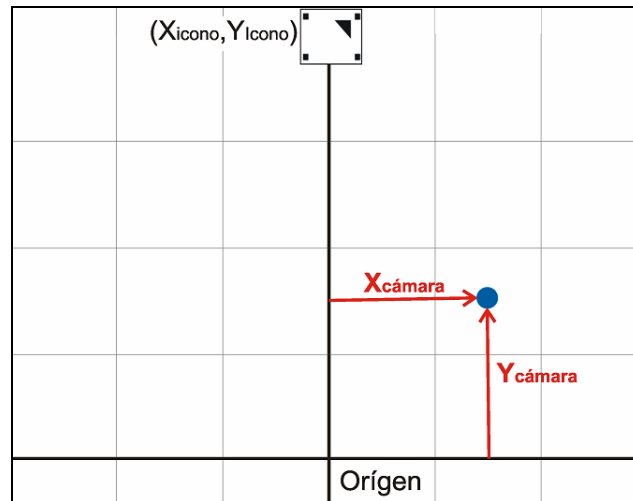


Figura 3.12. Coordenadas de la cámara

3.2 Implementación

Como sabemos, el objetivo principal del sistema es proporcionarnos las coordenadas de la cámara, además de que el sistema puede ser recalibrado, por lo cual se implementaron cada una de las interfaces, capaces de manipular la información de entrada y mostrar información referente a la operación realizada.

El propósito de desarrollar estas interfaces es que el usuario se pueda comunicar de manera sencilla con el sistema y que además sepa en todo momento que es lo que está ocurriendo con este.

3.2.1 Descripción gráfica del sistema

A continuación se muestra la estructura de las interfaces al arrancar el sistema, las cuales interactúan con el usuario por medio de eventos (figura 3.13).

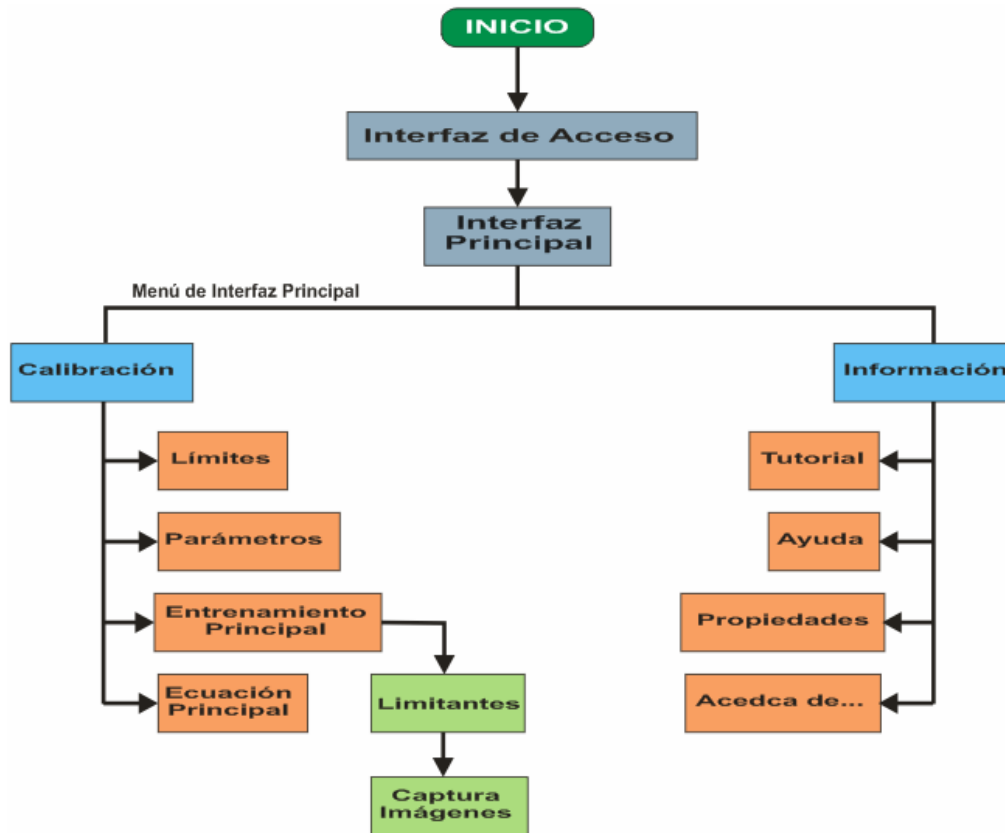


Figura 3.13. Estructura del sistema

3.2.2 Interfaz de inicio

Al iniciar el sistema se muestra la pantalla de inicio de este y al mismo tiempo se inicializan algunas variables (figura 3.14). Al presionar el botón *Login* se mostrará la pantalla de acceso al sistema y si se presiona el botón cerrar se finalizará este.

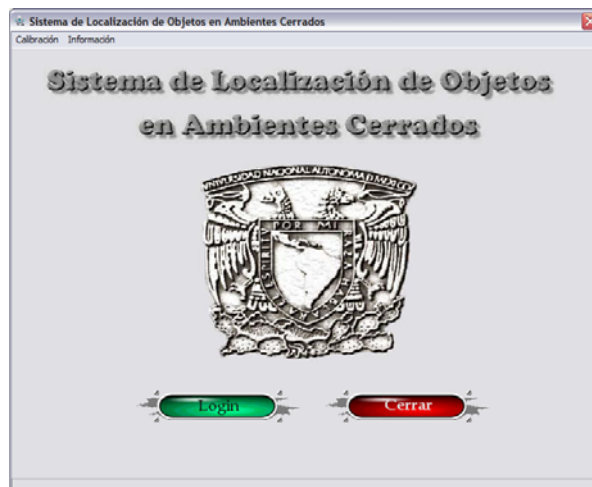


Figura 3.14. Pantalla de inicio

3.2.3 Interfaz de acceso

Esta interfaz solicita el nombre de usuario y contraseña, datos previamente establecidos en la configuración de la cámara para acceder al sistema (figura 3.15).

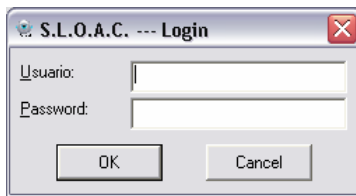


Figura 3.15. Pantalla de acceso

En caso de que los datos introducidos en los campos sean correctos se mostrará la pantalla principal del sistema, de lo contrario se mostrará un mensaje de error.

3.2.4 Interfaz principal

Esta pantalla es desde donde se opera al sistema, cuenta con dos menús (Calibración e Información), los cuales a su vez contienen submenús que serán detallados mas adelante. También cuenta con controles de movimiento y brillo de la cámara, botones de inicio, detención y cancelación de la búsqueda del icono y adicionalmente muestra información de la cámara con la que está trabajando este (figura 3.16).



Figura 3.16. Interfaz Principal

La figura 3.17.a muestra el menú Calibración y cuenta con las opciones Límites, Parámetros, Coordenadas Iconos, Entrenamiento y Ecuación Final, las cuales, en conjunto

con las opciones Tutorial, Ayuda, Propiedades y Acerca de, pertenecientes al menú Información figura 3.17.b fueron descritas en la sección 3.1.



Figura 3.17. Menús Principales

Dentro de la pantalla principal al presionar el botón Buscar Icono, se ejecutan las acciones necesarias para la búsqueda de un icono, así como también se activa dentro de la misma pantalla una ventana que nos muestra la escena que esta captando la cámara con un procesamiento de binarización, el cual sirve en primer lugar para detectar cuando el icono se encuentra centrado en la escena (figura 3.18).



Figura 3.18. Interfaz Principal con ventana de búsqueda

Es importante mencionar que el botón Detener búsqueda solo deshabilita la ventana de búsqueda, regresando a la pantalla principal mostrada en la figura 3.16 y el botón Reiniciar Posición realiza las acciones necesarias para dejar al sistema en su posición de HOME.

Una vez que el icono fue encontrado y procesado se deshabilita la ventana de búsqueda y se habilita la ventana de resultados, la cual contiene una representación de la ubicación de la cámara dentro del espacio de trabajo (figura 3.19).

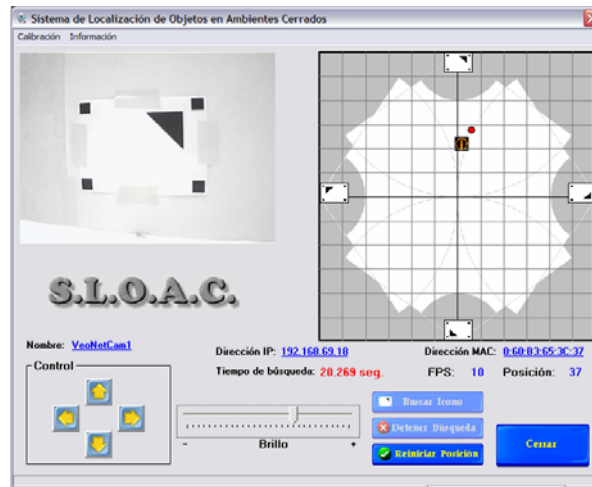


Figura 3.19. Interfaz Principal con ventana de resultados

Dentro de la ventana de resultados se aprecia el símbolo de información a un lado de la representación de la cámara, el cual al ser presionado nos muestra la pantalla de información de la posición de la cámara (figura 3.20).



Figura 3.20. Pantalla de información de la posición de la cámara

3.2.5 Interfaz Calibración Límites

Al seleccionar la opción Límites del menú calibración se presenta la pantalla mostrada en la figura 3.21, la cual cuenta con tres botones, los cuales se encargan de obtener y procesar las imágenes correspondientes a los límites superior e inferior de la región operativa, teniendo en cuenta para ello que la cámara debe ser colocada en dichas posiciones.



Figura 3.21. Interfaz Calibración Límites

3.2.6 Interfaz Captura Parámetros

Esta interfaz es mostrada al seleccionar la opción Parámetros del menú Calibración. En ella se capturan y se guardan, en un archivo, los valores con los cuales fue delimitada la región operativa (figura 3.22).

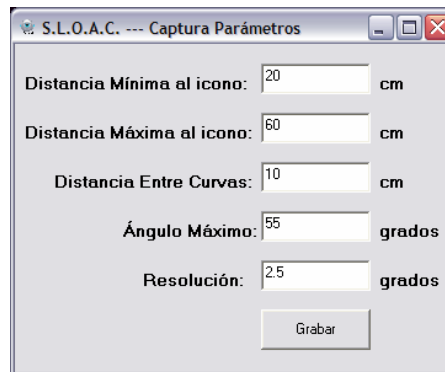


Figura 3.22. Interfaz Captura Parámetros

3.2.7 Interfaz Coordenadas Iconos

Si se selecciona la opción Coordenadas Iconos del menú Calibración se muestra la pantalla mostrada en la figura 3.23, la cual permite al usuario introducir las coordenadas x, y de cada uno de los iconos y almacenarlas en un archivo.

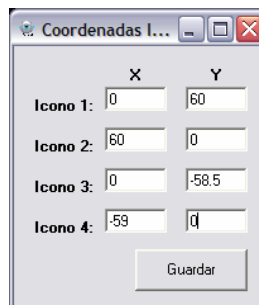


Figura 3.23. Interfaz Coordenadas Iconos

3.2.8 Interfaces Entrenamiento

Esta pantalla es mostrada al seleccionar la opción Entrenamiento del menú Calibración. En esta pantalla se muestran tres botones, Tomar Fotos, Entrena Matlab y Salir (figura 3.24).



Figura 3.24. Interfaz Entrenamiento

Al dar click en el botón Tomar Fotos se muestra la interfaz Limitantes mostrada en la figura 3.25, en la cual se introduce solo el parámetro de la distancia a la cual se va a calibrar en ese momento, ya que los otros dos parámetros mostrados son obtenidos del archivo guardado con la pantalla de Captura de Parámetros.

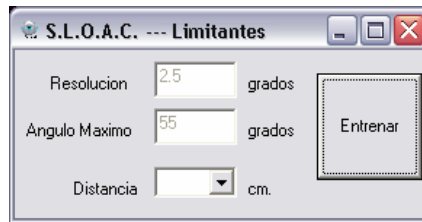


Figura 3.25. Interfaz Limitantes

Al introducir la distancia y presionar el botón Entrenar se muestra la pantalla de Captura de Imágenes que se muestra en la figura 3.26, la cual se encarga de ir tomando cada una de las fotos a lo largo de una curva de entrenamiento como se explicó en el ejemplo 1 de la sección 3.1.5.

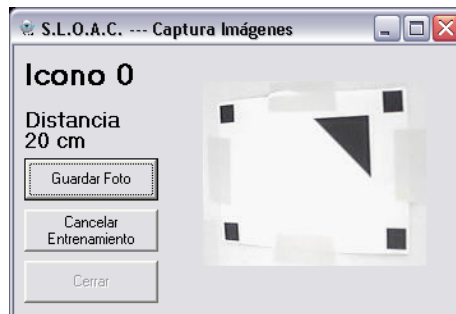


Figura 3.26. Interfaz Captura de Imágenes

Cuando se han terminado de tomar todas las fotos pertenecientes a una curva, automáticamente se habilita el botón Entrena Matlab de la interfaz principal de entrenamiento, este se encarga de realizar las acciones necesarias para caracterizar la curva correspondiente y almacenar los datos obtenidos en un archivo.

Finalmente, una vez que se han caracterizado todas las curvas, se selecciona la opción Ecuación Final del menú Calibración, la cual carga todos los datos de las caracterizaciones de las curvas para poder obtener así, las ecuaciones finales pertenecientes a los grupos GI y GII explicadas en el ejemplo 1 de la sección 3.1.5, al seleccionar esta opción no se muestra ninguna interfaz, sólo se muestra un mensaje que nos indica si el proceso terminó con éxito (figura 3.27).

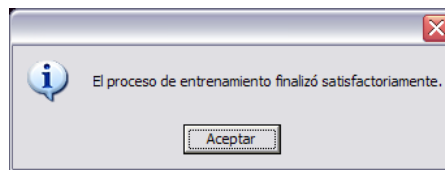


Figura 3.27. Mensaje final de la calibración

3.2.9 Interfaz del Tutorial

Al acceder a la opción Tutorial del menú Información se muestra la pantalla mostrada en la figura 3.28, la cual nos reproduce los videos que nos explican el funcionamiento del sistema.

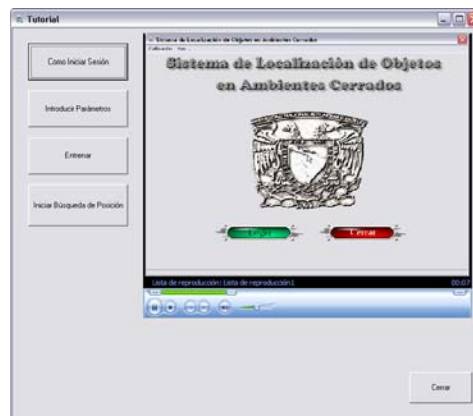


Figura 3.28. Tutorial

3.2.10 Archivo de Ayuda

Al seleccionar la opción Ayuda del menú información se muestra un archivo de ayuda que describe a grandes rasgos el sistema (figura 3.29).



Figura 3.29. Archivo de Ayuda

3.2.11 Pantalla Propiedades

Esta pantalla es mostrada al seleccionar la opción Propiedades del menú Información y nos presenta información referente al sistema, como las coordenadas en las que se encuentran los iconos, resolución con la que trabaja la cámara, dimensiones del espacio de trabajo, etc. (figura 3.30).

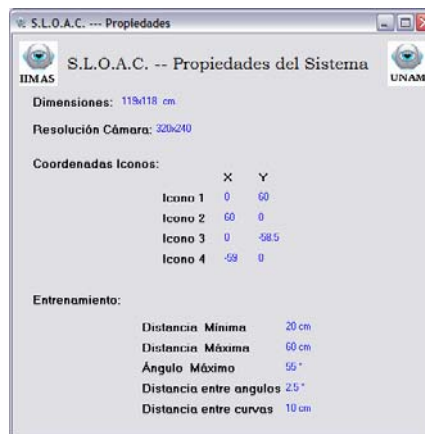


Figura 3.30. Pantalla Propiedades del Sistema

3.2.12 Pantalla Acerca de

En esta pantalla se muestra el título del sistema, la versión de este, el objetivo del mismo y los nombres de los creadores, es mostrada cuando se pulsa la opción Acerca de en el menú Información (figura 3.31).

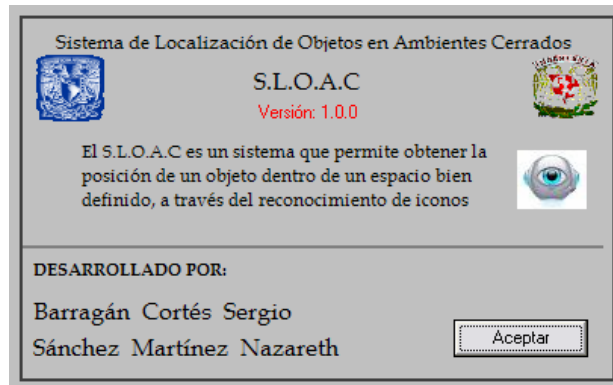


Figura 3.31. Pantalla Acerca de

3.3 Hardware

3.3.1 Diseño del controlador

Debido a las características de fabricación de la cámara fue necesario diseñar un dispositivo que nos permitiera expandir el ángulo de paneo de ésta, ya que existen zonas dentro de la región de trabajo para las cuales el ángulo máximo que logra cubrir (120°) es insuficiente para alcanzar a detectar algún icono.

Para el diseño de dicho dispositivo fue necesario tomar en cuenta dos aspectos, el peso de la cámara y la velocidad a la cual se requiere que se realice el paneo.

En cuanto al peso de la cámara decidimos utilizar un servomotor modificado (para poder trabajar con corriente directa) marca *Futaba* modelo S3003 (véase apéndice D), el cual tiene un bajo consumo de voltaje y soporta hasta 2kg de carga.

Para controlar la velocidad de paneo a la cual se quiere hacer girar el motor, decidimos hacerlo por medio de una señal cuadrada, la cual es generada por el circuito integrado LM555, conectado en una configuración “*astable*” como se muestra en la figura 3.32.

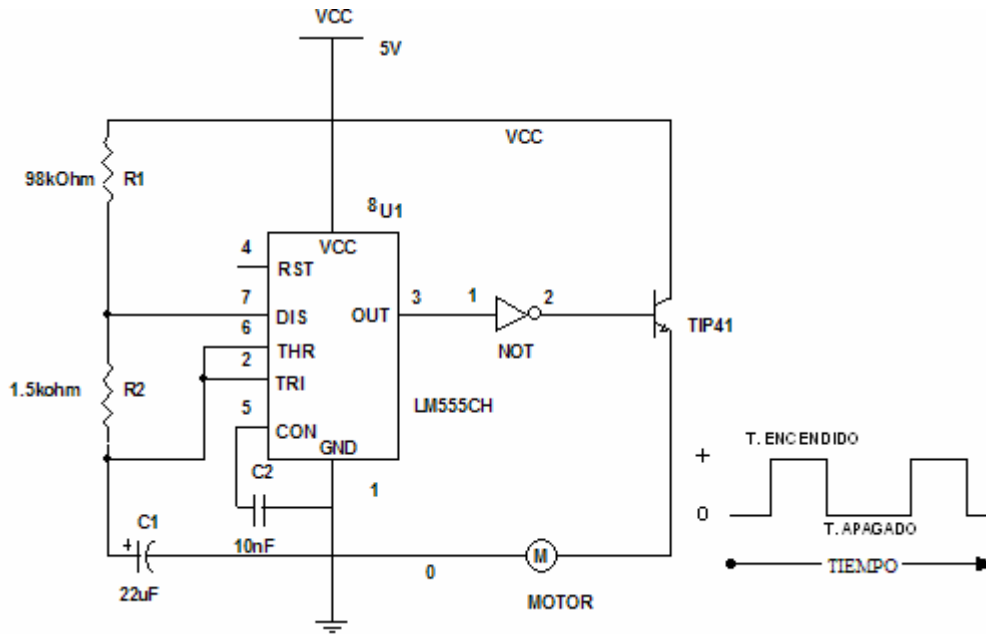


Figura 3.32. Diagrama del controlador

Los valores de las resistencias y los capacitores utilizados para obtener la señal deseada se calcularon mediante las siguientes ecuaciones:

Para calcular tiempo en alto:

$$T_H = 0.693 \times (R_1 + R_2) \times C_1 \quad \dots(1)$$

Para calcular tiempo en bajo:

$$T_L = 0.693 \times R_2 \times C_1 \quad \dots(2)$$

Para calcular la frecuencia:

$$f = \frac{1}{0.693 \times (R_1 + 2R_2) \times C_1} \text{ [Hz]} \quad \dots(3)$$

Para calcular el ciclo de trabajo:

$$CT = \frac{R_1 + R_2}{R_1 + 2R_2} \times 100 \text{ [%]} \quad \dots(4)$$

Para lograr la señal adecuada se fijaron lo siguientes valores:

$$T_H = 1.5 \text{ [s]}. \quad T_L = 0.02 \text{ [s]}. \quad C_1 = 22 \text{ [\mu F]}.$$

Sustituyendo en (2).

$$0.02 = 0.693 \times R_2 \times 22\mu$$

$$R_2 = \frac{0.02}{0.693 \times 22\mu}$$

$$R_2 = 1.4 [k\Omega]$$

Sustituyendo en (1).

$$1.5 = 0.693 \times (R_1 + 1368) \times 22\mu$$

$$R_1 = \frac{1.5}{0.693 \times 22\mu} - 1400$$

$$R_1 = 98.2 [k\Omega]$$

Sustituyendo los valores de R_1 , R_2 , y C_1 en (3) y (4) obtenemos el valor de la frecuencia y el ciclo de trabajo.

$$f = \frac{1}{0.693 \times (98200 + (2 \times 1400)) \times 22\mu} [Hz]$$

$$f = 0.6493 [Hz]$$

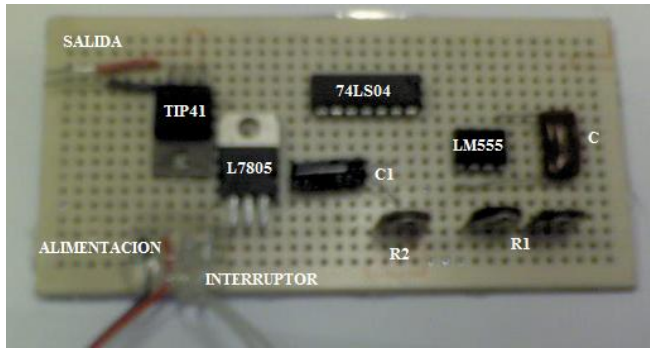
$$CT = \frac{98200 + 1400}{98200 + 2(1400)} \times 100$$

$$CT = 98.6 \%$$

Para diseñar un circuito como el anterior hay que tener en cuenta que el tiempo en alto debe ser mayor al tiempo en bajo, esto debido a condiciones del circuito LM555, por lo cual, decidimos diseñar el circuito con lógica inversa y posteriormente hacer uso de una compuerta 74LS04 para invertir los niveles de voltaje y así obtener la salida necesaria.

3.3.2 Implementación del controlador

Para construir el controlador decidimos realizar el circuito impreso de este en una placa fenolica de 5 [cm] x 10 [cm] (figura 3.33.a), la cual se introduce en una base de 12 [cm] x 13 [cm] x 3 [cm] que se diseño para contener a este y al motor (figura 3.33.b).



(a)

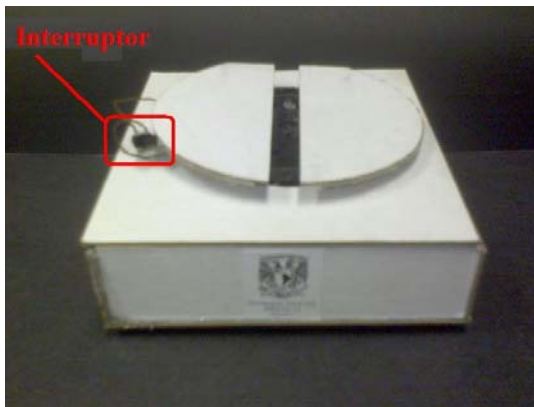


(b)

Figura 3.33. (a) Circuito del controlador, (b) base con motor.

Una vez armado el controlador fue necesario determinar la forma en la que este debía ser activado cuando el paneo de la cámara llegara a su ángulo máximo, por lo cual se agregó un interruptor a la base antes mencionada.

Para activar el interruptor existen varias formas, sin embargo por cuestiones de practicidad decidimos agregarle una extensión rígida a la cámara, la cual tiene la función de activar el interruptor cuando esta llega a su ángulo máximo de paneo (figura 3.34).



(a) Base con motor y controlador en su interior e interruptor en el exterior.

(b) Cámara con extensión para habilitar el controlador

Figura 3.34. Sistema de control de apoyo para el paneo

Capítulo 4

Resultados Experimentales

Introducción

Una fase importante en el desarrollo de sistemas es la elaboración de pruebas y obtención de resultados, ya que estos indican el comportamiento del sistema, lo que además de caracterizarlo lleva a tomar decisiones para la corrección de errores o para mejorarlo en algún aspecto. En el presente capítulo se mencionan las pruebas realizadas así como los resultados obtenidos.

4.1 Pruebas de precisión

Para la realización de estas pruebas se utilizaron las regiones operativas de cada icono, las cuales se dividieron en tres zonas, ubicadas a 20, 40 y 55 [cm] (figura 4.1), esto con el objetivo de obtener un parámetro que nos indicara que tan precisas son las respuestas del sistema dentro de estas regiones.

Se eligieron ocho puntos al azar pertenecientes a cada una de las tres zonas y se obtuvieron las coordenadas reales de estos puntos, posteriormente se colocó la cámara en cada uno de ellos y se ejecutó el sistema para obtener su respuesta y compararla con los valores reales, obteniendo así las tablas 4.1 a 4.4, las cuales reflejan la comparación para de cada una de las zonas para los cuatro iconos.

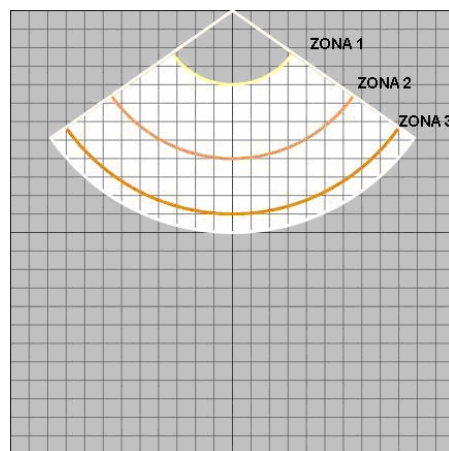


Figura 4.1. Vista superior de la región

Los datos obtenidos a través de estos experimentos fueron los siguientes:

ICONO 1					ICONO 2			
Medidas Reales		Medidas del Sistema		Medidas Reales		Medidas del Sistema		
X	Y	X	Y	X	Y	X	Y	
1	17.50	51.00	17.14	48.78	49.00	-16.30	48.46	-16.00
2	13.80	44.80	14.11	44.27	44.50	-12.90	44.45	-13.42
3	10.20	42.60	10.54	41.78	41.50	-7.50	40.47	-8.30
4	6.30	41.20	6.36	40.00	40.00	0.00	38.77	0.54
5	0.00	40.30	-0.83	38.55	40.80	5.00	40.08	6.26
6	-7.90	41.20	-8.49	40.71	42.30	8.60	41.89	9.72
7	-10.50	40.00	-11.19	39.56	44.20	11.80	44.15	12.71
8	-14.00	44.00	-14.82	43.79	46.00	15.50	45.61	15.58
9	30.60	33.00	29.40	32.16	37.3	-33.3	35.18	-31.32
10	25.80	29.00	25.14	28.24	33.30	-30.00	31.88	-28.44
11	20.00	25.00	18.82	23.93	29.00	-25.00	28.04	-24.29
12	10.20	21.50	9.89	20.86	22.50	-14.00	22.13	-14.26
13	-10.00	21.00	-9.82	20.61	20.00	0.00	19.21	-0.57
14	-17.70	24.00	-18.23	23.65	21.00	8.00	20.48	8.84
15	-23.50	29.00	-25.15	28.46	24.30	17.50	24.25	18.60
16	-29.00	33.00	-29.58	33.12	30.00	26.00	29.54	26.15
17	42.30	24.50	41.35	23.56	26.00	-47.30	25.02	-45.98
18	35.50	18.50	32.92	15.97	19.30	-39.00	18.59	-38.11
19	29.50	13.50	28.34	12.76	13.20	-31.30	15.14	-34.00
20	0.00	4.80	1.80	4.53	6.40	-16.30	6.86	-17.76
21	-6.00	5.00	-5.75	4.83	5.00	0.00	4.56	-1.80
22	-18.50	11.00	-21.27	12.06	7.20	14.00	7.85	17.27
23	-30.70	16.50	-33.98	19.03	12.90	27.50	13.74	29.01
24	-37.50	28.00	-38.18	29.47	23.00	39.50	22.62	39.41

Tabla 4.1.

Tabla 4.2.

ICONO 3					ICONO 4				
Medidas Reales		Medidas del Sistema			Medidas Reales		Medidas del Sistema		
X	Y	X	Y	X	Y	X	Y	X	Y
1	17.00	-47.50	16.94	-46.11	-48.50	16.00	-47.50	15.53	
2	12.00	-43.50	12.32	-42.82	-45.70	13.60	-46.03	14.51	
3	6.80	-41.50	7.18	-40.64	-43.10	10.40	-42.97	11.50	
4	-0.40	-39.00	-0.47	-37.69	-40.80	5.00	-39.84	6.03	
5	-5.80	-39.50	-7.20	-38.83	-40.00	0.00	-38.68	0.60	
6	-12.50	-42.40	-13.69	-41.57	-40.80	-5.00	-39.69	-6.10	
7	-15.30	-44.00	-15.60	-44.03	-42.30	-9.00	-41.64	-9.68	
8	-15.30	-42.50	-15.97	-42.37	-46.30	-15.00	-46.64	-14.85	
9	32.50	-37.00	30.61	-34.30	-31.00	27.50	-30.88	27.83	
10	28.00	-32.80	26.90	-30.85	-24.70	20.00	-24.77	20.75	
11	23.00	-28.00	22.07	-26.53	-21.00	10.00	-20.27	9.95	
12	13.00	-21.70	12.22	-20.45	-20.00	0.00	-19.62	0.67	
13	0.00	-20.00	-0.90	-19.18	-20.00	-5.00	-19.81	-6.16	
14	-16.00	-24.00	-18.66	-24.15	-21.00	-10.00	-20.78	-11.02	
15	-22.90	-29.50	-24.84	-28.98	-24.00	-18.50	-23.91	-18.73	
16	-32.00	-37.71	-31.72	-35.36	-29.40	-26.80	-28.51	-25.56	
17	41.00	-31.30	37.27	-26.54	-19.00	37.40	-20.37	38.66	
18	33.00	-23.20	31.48	-21.50	-10.00	24.00	-10.40	24.54	
19	20.30	-16.00	19.40	-15.57	-5.80	10.00	-5.29	11.03	
20	0.00	-10.00	1.72	-9.37	-5.00	0.00	-4.99	2.22	
21	-11.00	-11.00	-13.32	-11.12	-6.00	-10.00	-5.80	-8.30	
22	-21.00	-17.50	-22.03	-16.91	-8.80	-20.00	-7.71	-16.88	
23	-32.30	-25.00	-33.09	-25.50	-14.00	-30.00	-14.75	-31.78	
24	-39.00	-31.00	-37.42	-28.50	-20.00	-38.00	-19.73	-36.74	

Tabla 4.3.

Tabla 4.4.

Con los valores obtenidos realizamos la graficación de los puntos para ejemplificar de una manera más clara el comportamiento de dichos valores, representando las medidas reales con círculos azules y las medidas entregadas por el sistema con triángulos rojos.

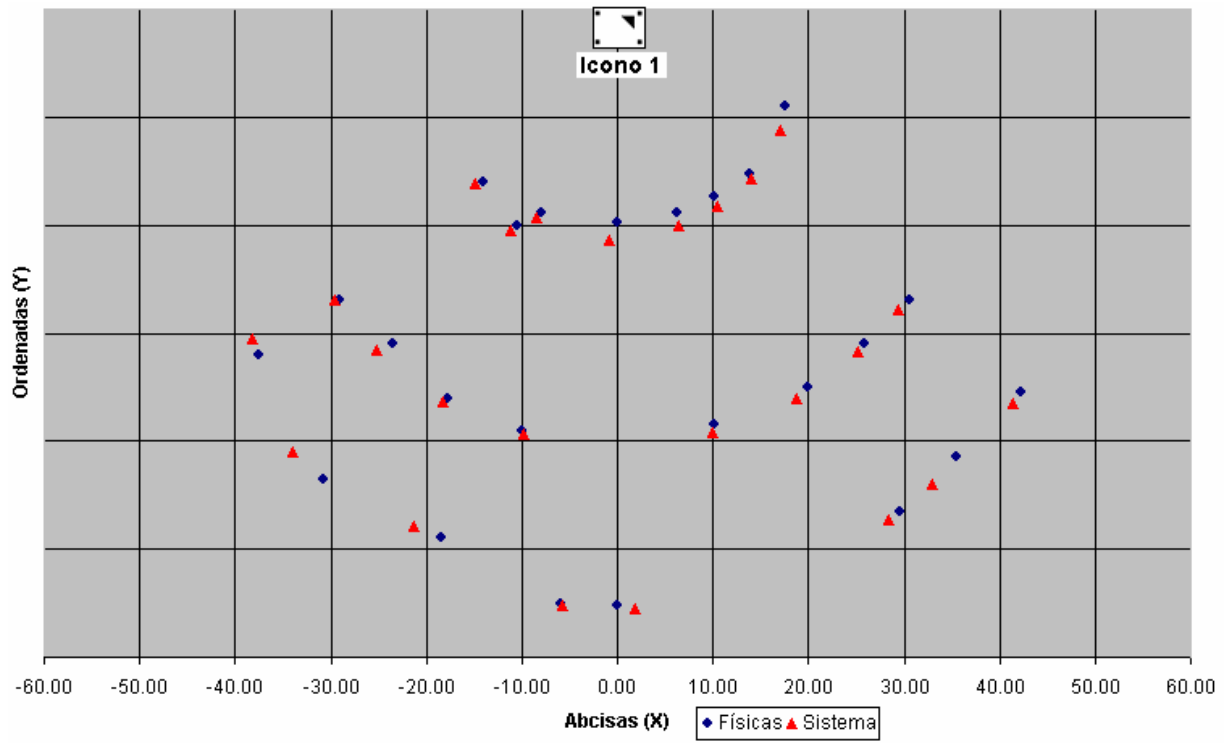


Figura 4.2.

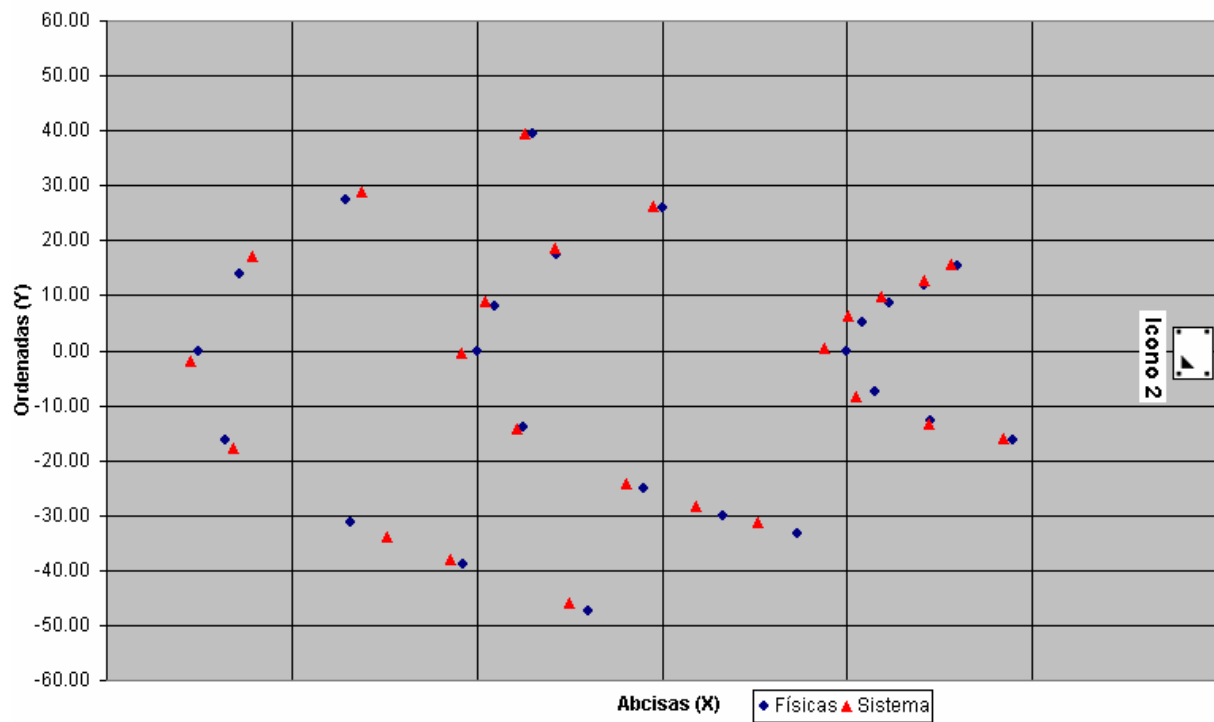


Figura 4.3.

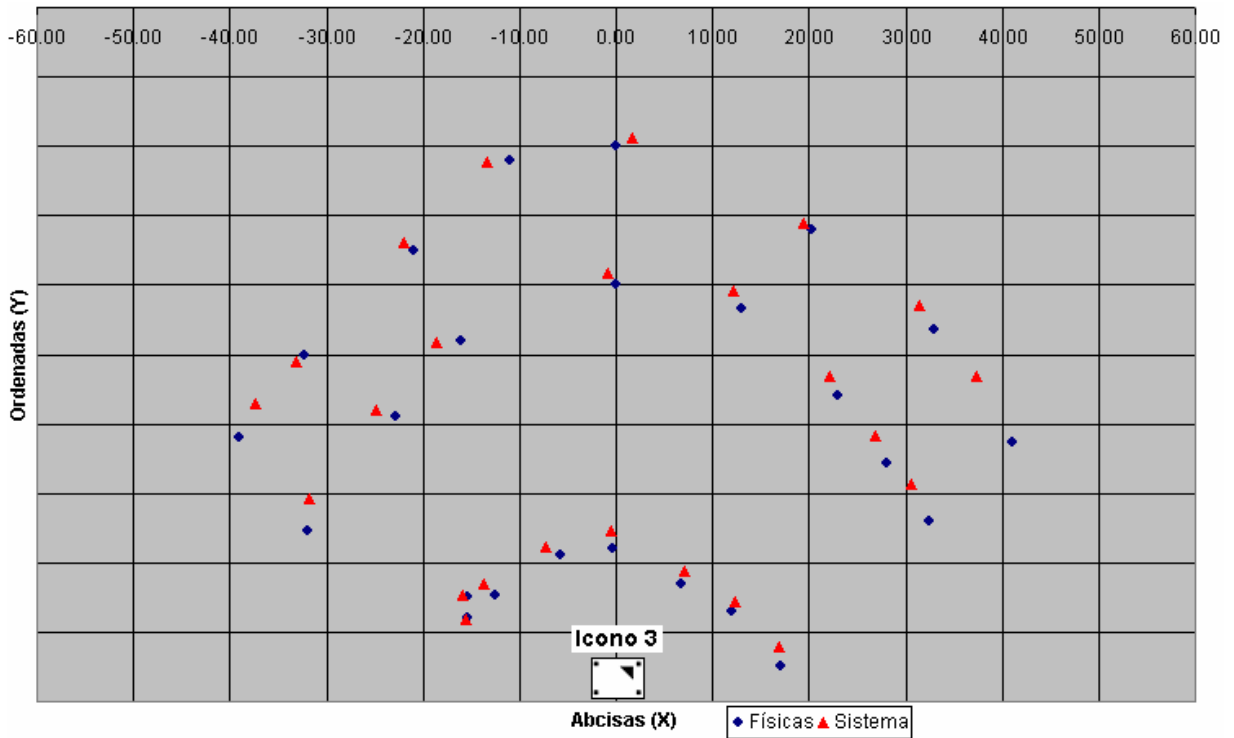


Figura 4.4.

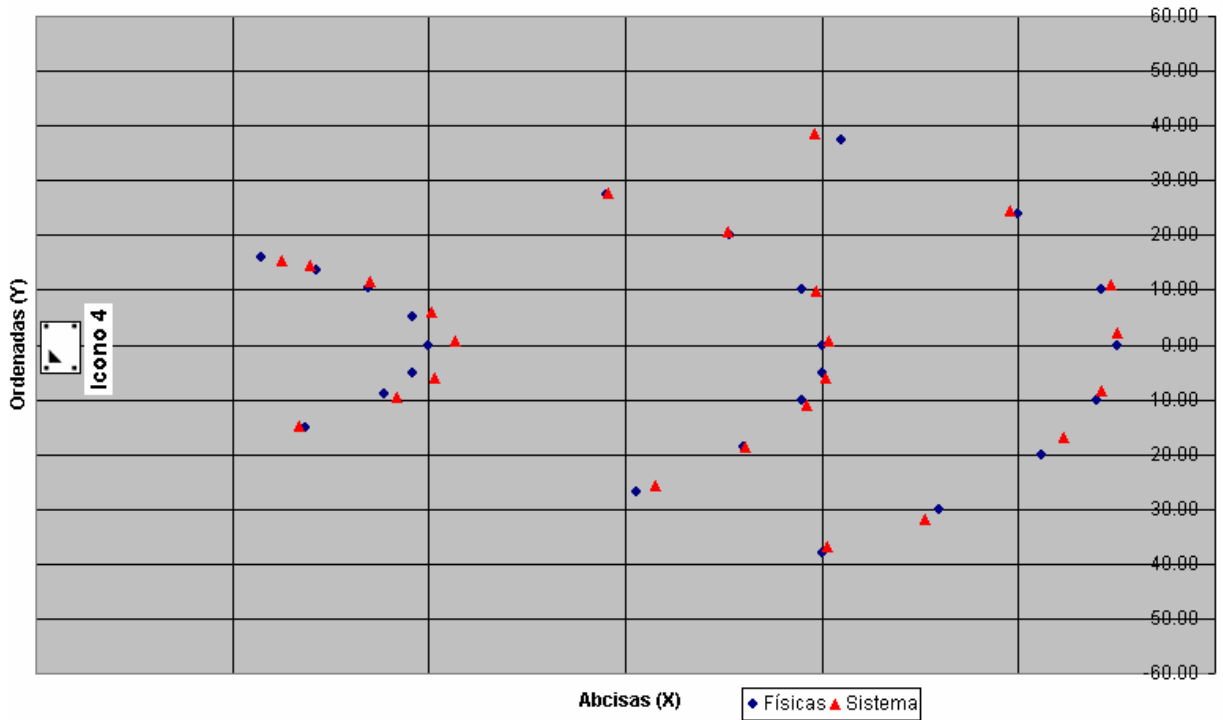


Figura 4.5.

Con los datos anteriores obtuvimos el error para cada una de las lecturas, el cual fue calculado a través de la ecuación 4.1, dando como resultado las siguientes tablas.

$$\text{Error} = \text{Medida Real} - \text{Medida Sistema} \quad \dots(4.1)$$

ICONO 1		ICONO 2		ICONO 3		ICONO 4	
X	Y	X	Y	X	Y	X	Y
0.36	2.22	0.54	0.30	0.06	1.39	1.00	0.47
0.31	0.53	0.05	0.52	0.32	0.68	0.33	0.91
0.34	0.82	1.03	0.80	0.38	0.86	0.13	1.10
0.06	1.20	1.23	0.54	0.07	1.31	0.96	1.03
0.83	1.75	0.72	1.26	1.40	0.67	1.32	0.60
0.59	0.49	0.41	1.12	1.19	0.83	1.11	1.10
0.69	0.44	0.05	0.91	0.30	0.03	0.66	0.68
0.82	0.21	0.39	0.08	0.67	0.13	0.34	0.15
1.20	0.84	2.12	1.98	1.89	2.70	0.12	0.33
0.66	0.76	1.42	1.56	1.10	1.95	0.07	0.75
1.18	1.07	0.96	0.71	0.93	1.47	0.73	0.05
0.31	0.64	0.37	0.26	0.78	1.25	0.38	0.67
0.18	0.39	0.79	0.57	0.90	0.82	0.19	1.16
0.53	0.35	0.52	0.84	2.66	0.15	0.22	1.02
1.65	0.54	0.05	1.10	1.94	0.52	0.09	0.23
0.58	0.12	0.46	0.15	0.28	2.35	0.89	1.24
0.95	0.94	0.98	1.32	3.73	4.76	1.37	1.26
2.58	2.53	0.71	0.89	1.52	1.70	0.40	0.54
1.16	0.74	1.94	2.70	0.90	0.43	0.51	1.03
1.80	0.27	0.46	1.46	1.72	0.63	0.01	2.22
0.25	0.17	0.44	1.80	2.32	0.12	0.20	1.70
2.77	1.06	0.65	3.27	1.03	0.59	1.09	3.12
3.28	2.53	0.84	1.51	0.79	0.50	0.75	1.78
0.68	1.47	0.38	0.09	1.58	2.50	0.27	1.26

Tabla 4.5.

Tabla 4.6.

Tabla 4.7.

Tabla 4.8.

Obteniendo los promedios para las lecturas en X y para las lecturas en Y, podemos representar el comportamiento del error en cada una de las zonas (figura 4.6 y figura 4.7)

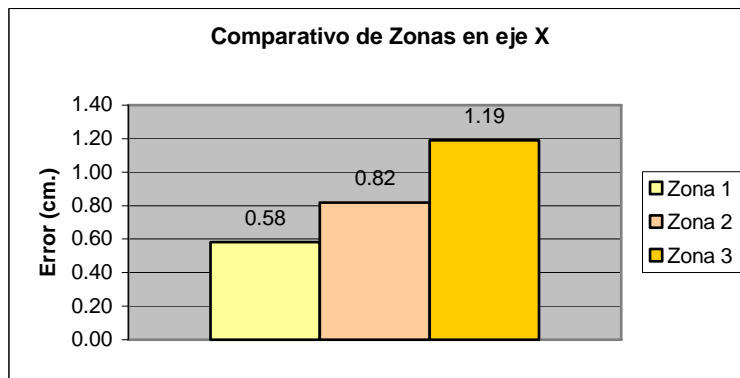


Figura 4.6.

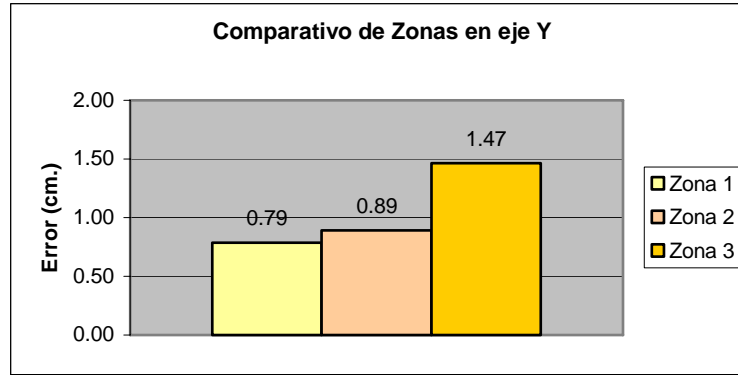


Figura 4.7.

Con los datos anteriores se puede establecer un promedio de error en x y un promedio de error en y para cada una de las zonas, con el objetivo de apreciar en cual de las tres el comportamiento del sistema es mas preciso:

Zona 1	Zona 2	Zona 3
$\bar{x} = 0.58 [cm]$	$\bar{x} = 0.82 [cm]$	$\bar{x} = 1.19 [cm]$
$\bar{y} = 0.79 [cm]$	$\bar{y} = 0.89 [cm]$	$\bar{y} = 1.47 [cm]$

Tabla 4.9.

Con estas pruebas podemos decir que mediante el análisis y la comparación de los valores obtenidos durante el desarrollo de estas, dichos valores nos indican que la precisión de los resultados que proporciona el sistema son directamente proporcionales a la distancia a la que es captado el icono, además también se puede apreciar en las gráficas de las figuras 4.2 a 4.5 que mientras mayor es el ángulo de visión, el error presente en los datos obtenidos a través del sistema se incrementa con respecto a los valores adquiridos a un ángulo menor.

4.2 Funcionamiento

Se realizaron pruebas que consistieron en probar el sistema completo (Software y Hardware), lo cual se hizo mediante la selección de diez puntos aleatorios dentro del espacio de trabajo, para posteriormente colocar la cámara junto con el controlador de apoyo en ellos y comparar la respuesta entregada por el sistema contra los valores de posición reales.

Los resultados obtenidos de las pruebas realizadas fueron los siguientes:

	Medidas Reales [cm]		Medidas del Sistema [cm]		Tiempo [s]
	X	Y	X	Y	
1	20.00	35.00	20.31	34.36	84.782
2	40.00	3.00	38.48	3.04	143.368
3	-4.00	5.00	-4.95	6.93	50.442
4	15.00	-31.50	18.70	-36.23	141.694
5	34.00	-16.50	33.47	-17.21	43.432
6	-19.00	-25.20	-20.97	-25.07	92.844
7	-40.00	-1.00	-39.64	-0.67	119.512
8	15.00	3.50	14.66	2.48	57.232
9	-25.00	29.00	-24.98	29.16	82.899
10	-25.30	29.60	-25.87	30.21	82.889
11					143.24

Tabla 4.10.

Como podemos apreciar en la tabla 4.10 se realizó una lectura extra con el objetivo de ilustrar que el sistema es capaz de distinguir cuando la cámara se encuentra fuera de la región de trabajo, y que tarda aproximadamente 143.24 [s] en determinarlo.

Capítulo 5

Conclusiones

Introducción.

Con la finalidad de evaluar el trabajo realizado a lo largo de esta tesis, se presentan las conclusiones, las cuales a su vez servirán como base para plantear al final de esta sección algunas opciones de trabajo a futuro.

5.1 Con respecto a las características de construcción del sistema.

Para cumplir con el objetivo de este proyecto fue necesario plantear algunas características físicas con las que debía contar el espacio de trabajo, tales como color, tamaño, y forma. Estas características fueron seleccionadas de forma ideal con la finalidad de facilitar el procesamiento de las imágenes y además contar con un ambiente en el cual fuera posible realizar pruebas de funcionamiento, sin embargo, la implementación del sistema es tal que es posible hacer una recalibración de éste para que tenga funcionalidad en espacios de trabajo de características semejantes a mayor escala.

En cuanto al diseño de los iconos, se realizó pensando en que estos estuvieran conformados por figuras geométricas básicas, ya que la técnica de reconocimiento de patrones empleada nos proporcionaba mejores resultados que con figuras geométricas compuestas.

Para la parte de calibración del sistema fue necesario elaborar una plantilla, en la cual se tuviera bien acotada la región operativa, lo cual influye directamente en la obtención de resultados, ya que la precisión de los modelos matemáticos empleados para el cálculo de la posición de la cámara depende de la precisión con la cual se coloque esta en los puntos de calibración determinados por la plantilla antes mencionada.

5.2 Con respecto al software desarrollado.

Se decidió implementar la interfaz del sistema en Visual Basic debido a que es un lenguaje que permite elaborar interfaces de una manera sencilla con el aspecto de cualquier interfaz del sistema operativo Windows, con las cuales se encuentra familiarizado el usuario, además de que la cámara con la que se trabajo cuenta con soporte para desarrolladores en este lenguaje de programación.

Para el procesamiento de las imágenes se empleo Matlab debido a que es una herramienta con la que se había trabajado anteriormente, la cual cuenta con un “*toolbox*” que incluye una gran cantidad de funciones de procesamiento digital de imágenes, las cuales fueron aprovechadas para la implementación de nuestros algoritmos, reduciendo de manera considerable la programación de estos.

Otro punto importante para la selección de las herramientas antes mencionadas, fue la facilidad con la que se pueden comunicar ambas por medio de un componente de Matlab llamado “*Matlab Automation Server*”.

Como es sabido, se desarrollaron algunos algoritmos que principalmente tienen la finalidad de llevar a cabo procesamiento de imágenes digitales, tal es el caso del algoritmo de detección del icono, el de cálculo de la distancia y el de identificación del icono, los cuales no representan mucha complejidad, sin embargo, el algoritmo mas complejo que se desarrollo fue el del cálculo del ángulo de visión ya que al igual que con el algoritmo del calculo de la distancia, se decidió en primer lugar, basarse en la técnica utilizada en el articulo “*Determinación de la posición de un objetivo por análisis de perspectiva, aplicada a un robot basketbolista*”, lo cual no nos proporcionaba buenos resultados, esto debido al diseño de nuestros iconos, por lo que basándonos en el hecho de que el sistema visual humano se puede modelar linealmente, decidimos obtener un modelo matemático que se apegara a esta característica, con lo cual resolvimos que se podía linealizar el comportamiento presentado, obteniendo así los resultados esperados.

5.3 Con respecto al funcionamiento del sistema.

Una vez obtenidos los resultados de las pruebas mencionadas en el capítulo 4, fue posible evaluar el desempeño del sistema, de esta evaluación se encontró que los resultados entregados por el sistema:

- Tienen mejor precisión cuando la cámara se encuentra más cerca del icono y van perdiendo ésta conforme se van alejando de él.
- Dependen en gran medida de la orientación, la zona y la región operativa en la que se encuentre la cámara.

- Podrían mejorarse si la calibración previamente realizada, se efectuara en un número mayor de puntos de calibración.
- Podría reducirse en gran medida el error presente en ellos si se lograra cuantificar de manera general el error que producen las aberraciones de la cámara.

En general, los resultados obtenidos durante las pruebas realizadas son satisfactorios, ya que existe un error pequeño, lo cual es atribuido al hecho de que en las mediciones reales efectuadas influye la apreciación del observador y a una caracterización regular de los errores producidos por las condiciones ópticas de la cámara.

Por ultimo podemos concluir que el objetivo principal del presente trabajo se cumplió de manera satisfactoria, aunque es posible realizar ciertas mejoras en aspectos como el controlador de apoyo, el empleo de una cámara más sofisticada, migración del sistema completo a otro lenguaje, como por ejemplo Java o C++, introducir ciertos elementos en el espacio de trabajo, entre otros, teniendo estos en conjunto como posibles opciones de trabajo futuro.

5.4 Trabajo futuro

Pensando en que el campo de la visión artificial no tiene mucha difusión entre los estudiantes de ingeniería, decidimos agregar esta sección, con el fin de que el proyecto desarrollado sirva de motivación y a su vez pueda ser retomado en un futuro para el desarrollo de otras tesis o proyectos de investigación.

Como se menciono anteriormente existen algunas modificaciones que se le pueden realizar al sistema con la finalidad de robustecerlo, las cuales quedan fuera del objetivo del presente trabajo. A continuación mencionamos algunas mejoras que consideramos factibles:

Migrar el sistema a otro lenguaje.

Dado que existen ciertas librerías enfocadas al procesamiento de imágenes, pertenecientes a otros lenguajes, es factible migrar el sistema desarrollado a uno de ellos, además de que la cámara con la que se trabajó cuenta con soporte para algunos de ellos, lo cual se vería reflejado en una reducción de costos debido a que algunos de esos lenguajes son de distribución libre, y además se podría lograr también con ello que el sistema fuese multiplataforma (funcionaria en sistemas operativos distintos a Windows).

Introducción de objetos físicos dentro del espacio de trabajo.

Actualmente el sistema tiene la capacidad de trabajar sin ningún problema dentro del espacio de trabajo ideal construido, pero se pueden realizar las modificaciones convenientes a los algoritmos desarrollados, de tal manera que el sistema sea capaz de discriminar objetos colocados en ciertos puntos del espacio de trabajo y obtener la localización de la cámara sin que estos influyan en sus resultados, este aspecto tiene un gran potencial a futuro, ya que si pensamos en implementarlo para la ayuda de personas invidentes o en el ambiente industrial, resulta de gran ayuda que el sistema interactúe con el usuario, en un ambiente de condiciones reales.

Implementación de un algoritmo de obtención de coordenada Z.

El sistema desarrollado, actualmente funciona en un sistema de coordenadas de dos dimensiones, el cual está limitado a movimientos de la cámara en un solo plano, por lo cual surge la opción de poder desarrollar un algoritmo de apoyo que se pueda integrar al sistema, el cual permita al sistema obtener la posición de la cámara en tres dimensiones.

Implementación de un sistema de control robusto.

Por cuestiones de practicidad, el dispositivo controlador desarrollado está muy limitado en cuanto a su funcionamiento e interacción con el sistema, sin embargo, es posible diseñar e implementar un dispositivo controlador basado en un microcontrolador comercial que pueda ser manipulado desde el propio sistema, vía inalámbrica.

Apéndice A

MATLAB *Automation Server*

MATLAB Automation Server

Es un protocolo que permite a una aplicación o componente (el cliente) controlar otra aplicación o componente (el servidor). Así un servidor Matlab puede ser controlado por cualquier programa en Windows que pueda ser un controlador de automatización como Excel, Access, Visual Basic y Visual C++.

Tipos de MATLAB Automation Server

Matlab Automation Server puede ser inicializado de dos formas distintas:

- Compartido. Una o mas aplicaciones cliente se conectan al mismo servidor MATLAB, el cual es compartido entre todos los clientes.
- Dedicado. Cada aplicación cliente crea su propio servidor MATLAB.

Directorio de arranque

Por *default* MATLAB Automation Server inicia en el subdirectorio `\bin\win32` perteneciente al directorio donde se instala MATLAB. Si se desea cambiar de directorio desde la aplicación cliente, se debe de utilizar la función “*cd*”.

Crear un MATLAB Automation Server desde Visual Basic

Para crear un MATLAB Automation Server desde una aplicación cliente desarrollada en Visual Basic se puede hacer uso de dos diferentes métodos:

- [NewMLApp.MLApp](#)
- [CreateObject](#)

El primer método requiere la creación de la referencia a *MATLAB Automation Server Type Library* en el proyecto de Visual Basic. Para crear la referencia se utiliza el siguiente procedimiento.

1. Ir al menú Proyecto
2. Seleccionar Referencias en el submenú
3. Seleccionar la casilla **MATLAB Automation Server Type Library** de la ventana Referencias
4. Click en Aceptar

Una vez hecho lo anterior se procede a iniciar el servidor con el siguiente código:

- `Matlab = New MLab.MLab`

Si se utiliza el segundo método, no es necesario crear la referencia a MATLAB Automation Server Type Library en el proyecto, basta con colocar las siguientes líneas de código:

```
Dim Matlab As Object
```

```
Set Matlab = CreateObject("Matlab.Application")
```

Funciones para ejecutar comandos en el servidor

La aplicación cliente puede ejecutar comandos en el servidor MATLAB con las siguientes funciones:

- `Execute`
- `Feval`

Por lo general se utiliza la función `Execute` cuando se quiere ejecutar un comando que puede ser expresado con una simple cadena, aunque se puede utilizar sin ningún problema con la concatenación de varias cadenas.

Para ejemplificar las dos distintas maneras de utilizar `Execute` se tiene el siguiente código:

```
Dim Result As String
```

```
Dim R, C As Integer
```

```
R=10
```

```
C=20
```

```
Result = Matlab.Execute("cd C:\MATLAB7\work\")
```

```
Result = Matlab.Execute("a=zeros(" & R & ", " & C & ");")
```

El equivalente al código anterior en Matlab es:

```
a=zeros(10,20)
```

Por otro lado, se utiliza la función `Feval` cuando se quiere ejecutar un comando en el servidor y no se puede expresar este en una simple cadena, por lo general, cuando se requiere pasar parámetros del cliente al servidor.

Un ejemplo de lo antes descrito es el siguiente:

Dim rows, cols As Integer

rows = 3

cols = 6

Matlab.Feval('reshape', 0, 'A=', rows, cols)

El equivalente al código anterior en Matlab es:

`reshape(A,3,6)`

Donde el parámetro 'A=' significa que es la variable "A" dentro del espacio de trabajo de MATLAB.

Cuando un comando devuelve algún resultado, como por ejemplo un arreglo, el cual corresponde a cierta variable y se desea obtener esa variable para procesarla de alguna forma en la aplicación cliente se utiliza la función "GetWorkspaceData".

Dim B(3, 6) As Double

B = Matlab.GetWorkspaceData('A', 'base',B)

B =

```
0.6208  0.2344  0.6273  0.3716  0.7764  0.7036
0.7313  0.5488  0.6991  0.4253  0.4893  0.4850
0.1939  0.9316  0.3972  0.5947  0.1859  0.1146
```

Intercambio de datos con el servidor

MATLAB proporciona varias funciones para leer y escribir cualquier tipo de datos en el espacio de trabajo del servidor MATLAB. In cada una de esas funciones se le pasa como parámetro el nombre de la variable que se va a leer o a escribir y el nombre del espacio de trabajo en el cual se encuentra la variable, por lo general 'base'.

- [GetCharArray](#). Obtiene un arreglo de caracteres desde el servidor
- [GetFullMatrix](#). Obtiene una matriz desde el servidor
- [GetWorkspaceData](#). Obtiene cualquier tipo de datos desde el servidor
- [PutCharArray](#). Almacena un arreglo de caracteres en el servidor
- [PutFullMatrix](#). Almacena una matriz en el servidor
- [PutWorkspaceData](#). Almacena cualquier tipo de datos en el servidor

Un ejemplo de cómo se utiliza una de estas funciones es el siguiente.

```
Dim Matlab As Object
Dim S As String
Set Matlab = CreateObject("Matlab.Application")
Matlab.PutCharArray('str', 'base', 'Este es un ejemplo');
S = Matlab.GetCharArray('str', 'base')
S = Este es un ejemplo
```

Las funciones antes descritas pueden ser utilizadas de igual manera que en el ejemplo anterior.

Controlar la ventana del servidor

Estas funciones te permiten ejecutar el servidor MATLAB en estado maximizado o minimizado

- [MaximizeCommandWindow](#). Muestra la ventana del servidor en el escritorio
- [MinimizeCommandWindow](#). Muestra la ventana del servidor minimizada

Modo de uso:

```
Dim Matlab As Object
Set Matlab = CreateObject("Matlab.Application")
Matlab.MinimizeCommandWindow
```

Cerrar el servidor de MATLAB

La siguiente función se utiliza cuando se ha terminado de utilizar el servidor y sirve para cerrar la sesión en el y además cerrarlo.

Para cerrar el servidor se utiliza la siguiente sentencia

- [Matlab.Quit](#)

Hacer invisible la Terminal del servidor

El servidor cuenta con la propiedad “*visible*” que sirve para mostrar o no mostrar una terminal de este, en caso de que se requiera interactuar desde esta con el servidor.

Por default la propiedad visible esta habilitada con valor igual a uno, por lo tanto, si se desea ocultar la terminal se tiene que introducir el siguiente código.

```
Dim Matlab As Object
```

```
Set Matlab = CreateObject("Matlab.Application")
```

```
Matlab.Visible = 0
```

Ejemplo. Ejecutar un programa de MATLAB desde Visual Basic

Este ejemplo ejecuta un archivo de MATLAB (correspondiente a una función creada por el usuario) llamado “*resuelve_bvp.m*” desde una aplicación cliente (desarrollada en Visual Basic), a través del servidor de MATLAB, también realiza una grafica con la ayuda de la función “*surf(peaks)*” en una nueva ventana y por ultimo ejecuta un procesamiento sencillo con una matriz.

El siguiente código puede ser introducido en un botón o algún otro control perteneciente a una aplicación de Visual Basic, y se ejecutara al presentarse algún evento determinado, como un click, un doble click etc.

```
Dim MatLab As Object
```

```
Dim Result As String
```

```
Dim MReal(1, 3) As Double
```

```
Dim MImag(1, 3) As Double
```

```
'Creamos el servidor
```

```
MatLab = CreateObject("Matlab.Application")
```

```
'Llamada al archivo de MATLAB
```

```
'se asume que el archivo resuelve_bvp existe en la ruta especificada
```

```
Result = MatLab.Execute("cd C:\MATLAB7\work\bvp")
```

```
Result = MatLab.Execute("resuelve_bvp")
```

```
'Llamada a la funcion de MATLAB surf(peaks)
```

```
Result = MatLab.Execute("surf(peaks)")
```

```
'Creamos unas matrices directamente en el servidor desde la 'aplicación y ademas operamos con ellas
```

```
Result = MatLab.Execute("a = [1 2 3 4; 5 6 7 8]")
```

```
Result = MatLab.Execute("b = a + a ")
```

```
'Obtenemos el valor de la variable b y lo almacenamos en variables de 'la aplicación
```

```
MatLab.GetFullMatrix("b", "base", MReal, MImag)
```

Apéndice B

Curva de ajuste y regresión

Curva de ajuste

En la práctica se encuentra que existe una relación entre dos o más variables, y se desea expresar esta relación en forma matemática determinando una ecuación que conecte las variables.

Un primer paso es la colección de datos, indicando los valores correspondientes de las variables. Por ejemplo, si x e y denotan la estatura y peso de un adulto, entonces una muestra de n individuos estaría dada por las estaturas x_1, x_2, \dots, x_n y los pesos y_1, y_2, \dots, y_n .

El paso siguiente es dibujar los puntos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ en un sistema de coordenadas rectangulares. El conjunto resultante es llamado a veces diagrama de dispersión, en el cual es posible visualizar una curva que se aproxime a los datos. Esta curva es llamada curva de aproximación. En la figura 1 se observa que los datos se aproximan bien por una recta y decimos que existe una relación lineal entre las variables.

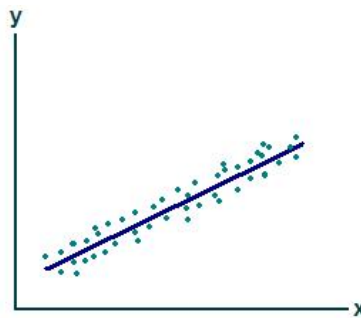


Figura 1.

El procedimiento general de hallar ecuaciones de curvas de aproximación que se ajusten a conjuntos de datos se denomina curva de ajuste. En la práctica el tipo de ecuación se sugiere frecuentemente del diagrama de dispersión. Así en la figura 1 podríamos utilizar una recta $y = a + bx$.

Regresión

Uno de los propósitos principales de la curva de ajuste es estimar una de las variables (dependiente) de la otra (independiente). El proceso de estimación se conoce como regresión. Si y se va a estimar a partir de x por medio de una ecuación la llamamos ecuación de regresión de y sobre x y a la curva correspondiente curva de regresión de y sobre x .

Método de mínimos cuadrados

Generalmente, más de una curva de un tipo dado parece ajustar a un conjunto de datos. Para evitar el juicio individual en la construcción de estas curvas de aproximación, es necesario obtener una definición de la “mejor curva de ajuste”.

Considérese la figura 2 en la cual los puntos de datos son (x_1, y_1) , $(x_2, y_2), \dots, (x_n, y_n)$. Para un valor dado de x , como por ejemplo x_1 , habrá una diferencia entre el valor de y_1 y el valor correspondiente determinado de la curva C . Denotamos esta diferencia por d_1 , que algunas veces se conoce como desviación, error o residuo y puede ser positivo, negativo o nulo. Análogamente, correspondiendo a los valores x_2, \dots, x_n obtenemos las desviaciones d_2, \dots, d_n .

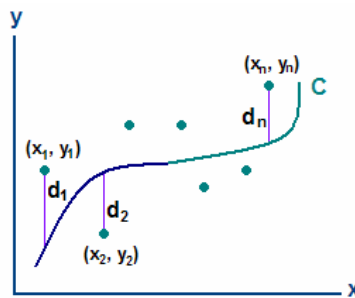


Figura 2.

Se dice que de todas las curvas de aproximación de un conjunto de puntos de datos, la curva que cumpla la propiedad, que a continuación se muestra, será la mejor curva de ajuste.

$$d_1^2 + d_2^2 + \dots + d_n^2 = \text{un mínimo}$$

Una curva con esta propiedad se dice que ajusta los datos en el sentido de mínimos cuadrados y se llama *curva de regresión de mínimos cuadrados* o simplemente curva de mínimos cuadrados. Se acostumbra emplear esta definición cuando x es la variable independiente e y la variable dependiente.

Recta de mínimos cuadrados

Empleando la definición anterior se puede demostrar que la recta de mínimos cuadrados de aproximación al conjunto de datos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, tiene la ecuación:

$$y = a + bx \quad \dots(1)$$

Donde las constantes a y b se determinan solucionando simultáneamente las siguientes ecuaciones:

$$\begin{aligned} \sum y &= an + b \sum x \\ \sum xy &= a \sum x + b \sum x^2 \end{aligned} \quad \dots(2)$$

Estas ecuaciones se conocen como las ecuaciones normales para la recta de mínimos cuadrados.

Obsérvese que por brevedad hemos utilizado $\sum y$, $\sum xy$ en lugar de $\sum_{i=1}^n y_i$, $\sum_{i=1}^n x_i y_i$.

Los valores de a y b obtenidos de (2) están dados por:

$$a = \frac{(\sum y(\sum x^2) - \sum x(\sum xy))}{n \sum x^2 - (\sum x)^2} \quad b = \frac{n \sum xy - (\sum x)(\sum y)}{n \sum x^2 - (\sum x)^2} \quad \dots(3)$$

El resultado para b en (3) también puede escribirse como

$$b = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2} \quad \dots(4)$$

Aquí, como es común, una barra encima de una literal indica media, es decir, $\bar{x} = \frac{(\sum x)}{n}$. Al

dividir ambos lados de la primera ecuación normal en (2) por n resulta:

$$\bar{y} = a + b\bar{x} \quad \dots(5)$$

Por tanto, podemos hallar primero b de (3) o (4) y luego emplear (5) para hallar $a = \bar{y} - b\bar{x}$.

Esto es equivalente a escribir la recta de mínimos cuadrados como

$$y - \bar{y} = b(x - \bar{x}) \quad \text{ó} \quad y - \bar{y} = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2} (x - \bar{x}) \quad \dots(6)$$

El resultado (6) indica que la constante b , que es la pendiente de la recta (1), es la constante fundamental para determinar la recta. De (6) también se ve que la recta de mínimos cuadrados pasa a través del punto (\bar{x}, \bar{y}) , denominado *centroide* o *centro de gravedad* de los datos.

Apéndice C

Cálculo de errores

Tipos de errores

Los errores más comunes involucrados en una medición instrumental son:

- Errores sistemáticos
- Errores no sistemáticos o aleatorios

Errores sistemáticos Son aquellos que proceden de la instrumentación utilizada y del propio observador como por ejemplo un metro mal marcado. Dentro de estos errores encontramos los siguientes tipos:

- Error de precisión. Es debido a las limitaciones de escala del aparato de medida: mínima diferencia que podemos apreciar con el aparato de medida.
- Error de calibración. La calibración es el proceso de hacer que las indicaciones de un instrumento de medida se correspondan con los valores reales de la magnitud medida. El error cometido en este proceso es el error de calibración.
- Error del observador que toma medidas. Por ejemplo la respuesta del observador en las medidas de tiempo de un experimento es siempre mayor que la limitación de escala del propio cronómetro

Errores aleatorios

Los instrumentos de medición tienen algunos errores identificados, ellos son:

- Exactitud
- Precisión
- Repetibilidad
- Reproducibilidad

Exactitud

El error de exactitud es la diferencia entre el valor medio medido por el instrumento y el valor exacto correspondiente a la magnitud medida. El valor exacto se conoce por la certificación del patrón empleado.

$$\% EE = \frac{|V_T - V_E|}{V_T} \times 100$$

V_T Valor teórico

V_E Valor experimental

Precisión

La precisión de un instrumento está relacionada con el error medio (desviación estándar) de cada medición. Para medirlo se toma una serie de mediciones de un mismo objeto en el mismo momento y lugar, se obtiene el promedio y su dispersión. Esta dispersión será la magnitud de su precisión.

$$\% EP = \frac{|V_T - V_{MA}|}{V_T} \times 100$$

V_T Valor teórico

V_{MA} Valor mas alejado

Repetibilidad

Es el error de un instrumento al medir un mismo objeto del mismo modo a lo largo del tiempo, ya sea de manera diaria, semanal, mensual, anual, etc.

Reproducibilidad

Es el error de medición entre diferentes instrumentos para medir el mismo objeto.

Apéndice D

Circuito Temporizador 555 Astable

Temporizador 555

El temporizador 555 es un circuito TTI integrado compatible que se puede utilizar como oscilador para generar una forma de onda de reloj. Básicamente es un circuito de conmutación que tiene dos niveles distintos de salida. Estando conectados correctamente los componentes externos, ninguno de los niveles de salida. Este circuito oscila y la salida tiene una forma de onda rectangular periódica. Como ningún estado de salida es estable, se dice que este circuito es astable y se le llama también multivibrador astable o de funcionamiento libre. La frecuencia de oscilación así como el ciclo de trabajo son exactamente controlados por dos resistores externos y un solo condensador temporizador.

El símbolo lógico de un temporizador LM555 conectado como oscilador está representado en la figura 1.

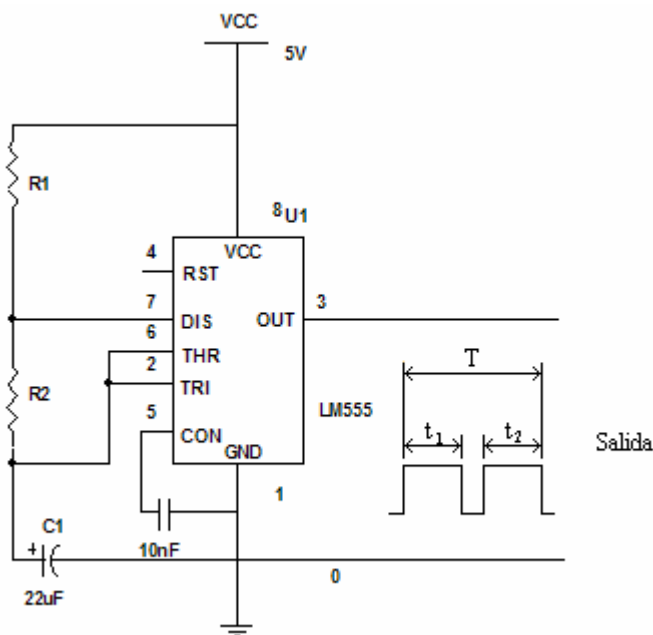


Figura 1. Circuito Astable

El condensador temporizador C se carga hacia $+V_{cc}$ a través de los resistores R_1 y R_2 . Las hojas de especificaciones proporcionan las ecuaciones para los componentes de tiempo R_1 , R_2 , y C. El tiempo de carga t_1 es:

$$t_1 = 0.693(R_1 + R_2)C$$

Este es el tiempo durante el cual la salida es alta, como muestra la figura 1. El condensador temporizador C es luego descargado hasta el nivel de masa (GND) a través del resistor R_2 . El tiempo de descarga t_2 es:

$$t_2 = 0.693R_2C$$

Este es el tiempo durante el cual la salida es baja, como muestra la figura1. Las ecuaciones para el cálculo de la frecuencia, el periodo y el ciclo de trabajo son las siguientes:

$$T = t_1 + t_2 = 0.693(R_1 + 2R_2)C$$

$$f = \frac{1}{T} = \frac{1.44}{(R_1 + 2R_2)C}$$

$$C.T. = \frac{t_1}{t_1 + t_2} = \frac{t_1}{T} = \frac{R_2}{R_1 + 2R_2}$$

Para mayor información sobre éste circuito integrado, véase su hoja de especificaciones en el apéndice F.

Apéndice E

Artículos consultados

DETERMINACIÓN DE LA POSICIÓN DE UN OBJETIVO POR ANÁLISIS DE PERSPECTIVA, APLICADA A UN ROBOT BASKETBOLISTA

RESUMEN

El presente trabajo introduce un método para la determinación de la posición de un objetivo mediante el análisis de la deformación por perspectiva que experimenta su imagen. Este método fue desarrollado y utilizado en la construcción de un robot capaz de lanzar y encestar una pelota en un aro de basketball.

INTRODUCCIÓN

El robot presentado consiste en una máquina capaz de lanzar y encestar un balón en un aro de basketball, desde cualquier punto en el que sea colocado, vía control remoto, por el usuario. El "área de juego" válida es una región con forma de sector de corona circular, con las dimensiones mostradas a continuación en la figura 1:



Figura 1(vista aérea)

Una vez posicionado el robot dentro del área de juego, el usuario acciona un botón de disparo. De allí en adelante, el robot se encarga de realizar todas las tareas necesarias para lanzar el balón y encestar. Este documento no pretende describir en detalle todos los procesos involucrados, sino más bien resaltar uno en particular: la determinación de la posición del tablero relativa al robot mediante el análisis de la deformación por perspectiva que sufre la imagen de un objetivo al ser proyectada sobre un plano.

MARCO TEORICO

Visión por computadora

Definición: Conjunto de técnicas y modelos que permiten el procesamiento, análisis y explicación de cualquier tipo de información espacial obtenida a través de imágenes digitales.

La visión por computadora se ocupa del estudio de los métodos que permiten emular la percepción visual. Aunque algunos de estos métodos son biológicamente plausibles, otros se limitan simplemente a reproducir el resultado final, como por ejemplo, determinar la distancia a un objeto o identificar un patrón.

La visión por computadora ha basado su desarrollo en el estudio del sistema visual humano, el cual sugiere la existencia de diferentes tipos de tratamiento de la información visual dependiendo de metas u objetivos específicos. Esto significa que la información visual percibida es procesada de distintas formas sobre la base de las características particulares de la tarea a realizar. Ello ha motivado a la visión por computadora a proponer varias técnicas para la obtención de una representación del mundo a partir del análisis de imágenes obtenidas empleando cámaras de video.

Debido a que la información visual es una de las principales fuentes de datos del mundo real, resulta útil el proveer a una computadora del "sentido de la vista" (a partir de imágenes tomadas con cámaras digitales o analógicas), que junto a otros mecanismos, tales como el aprendizaje, la hagan una herramienta capaz de detectar y ubicar objetos en el mundo real.

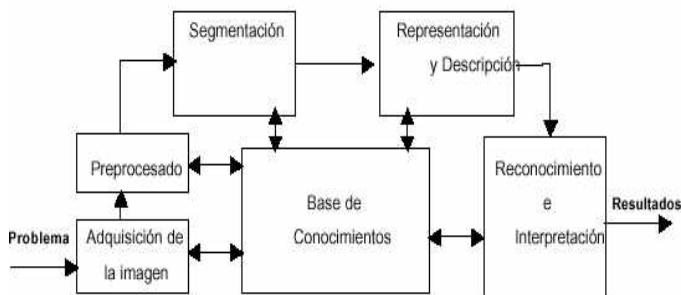
Componentes de un sistema de visión artificial

En todo sistema de visión por computadora podemos distinguir las siguientes seis etapas:

- Captación: Es el proceso a través del cual se obtienen las imágenes visuales.
- Preprocesamiento: Incluye técnicas tales como la reducción de ruido y realce de detalles.
- Segmentación: Es el proceso que divide a una imagen en objetos de interés.

- **Descripción:** Es el proceso mediante el cual se obtienen características convenientes para diferenciar un tipo de objeto de otro, como por ejemplo, tamaño y forma.
- **Reconocimiento:** Es el proceso que identifica a los objetos de una escena. Un ejemplo de ello podría ser el reconocimiento de diferentes tipos de piezas en un tablero de juego.
- **Interpretación:** Es el proceso que asocia un significado a un conjunto de objetos reconocidos.

El diagrama a continuación mostrado ilustra las relaciones que guardan las etapas antes mencionadas.



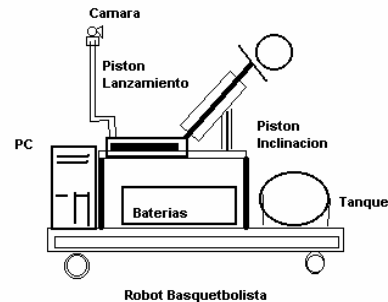
Principales etapas en un sistema de visión por computadora

MÉTODOS Y MATERIALES

Métodos:

El robot consta de 3 partes mecánicas importantes:

- ❖ La base del robot la cual soporta a todo el sistema mecánico, electrónico y de software.
- ❖ Base Secundaria en la cual se apoya el plato giratorio, el sistema pistón – pelota de basket como también las electroválvulas.
- ❖ Por último el sistema giratorio el cual incluye el pistón de lanzamiento, las electroválvulas, la cámara digital y el soporte de la pelota de basket.



La adquisición de información acerca de la posición del tablero se realiza mediante una única cámara digital (WebCam) montada sobre el plato giratorio del cañón y alineada con el mismo. En el tablero y debajo del aro se encuentra situada una figura conocida. El sistema de visión analiza los cuadros provenientes de la cámara y busca la figura, girando la plataforma que contiene a la cámara y al cañón. Una vez encontrada la figura objetivo, el plato se detiene justo cuando la cámara se encuentra alineada con ella. En estas condiciones se procede a realizar un análisis de perspectiva sobre la figura objetivo a fin de determinar la distancia al tablero y su orientación.

El robot necesita pasar por un proceso de calibración o “entrenamiento”. En este proceso se obtienen a prueba y error valores de presión y corrección de orientación necesaria para encestar desde varios puntos “con tablero”. Durante el entrenamiento, se debe informar al sistema luego de cada disparo si logró encestar o si se necesita variar la presión y / o el ángulo de desviación.

Luego del periodo de entrenamiento el robot está listo para interpolar los valores adecuados de presión y ángulo de corrección necesarios en posiciones de disparo no conocidas por él.

A continuación describimos los métodos empleados:

Reconocimiento del Objeto: El robot ha sido programado para reconocer una única figura objetivo, que consiste de tres franjas de colores verde, azul y rojo sobre un fondo blanco (Ver Figura 2):

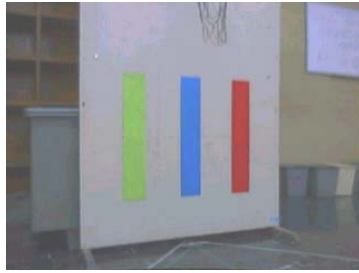


Figura 2

El proceso consiste en reconocer individualmente cada franja y comprobar el orden y la adecuada separación de las mismas. Estos criterios permiten la identificación de la figura objetivo independientemente de la distancia y ángulo con que sea observada.

Alineación del Cañón en lazo cerrado: El cañón y la cámara se encuentran alineados sobre un plato giratorio accionado por un motor. Esto permite el sentido del error en la orientación del cañón respecto de la orientación deseada. En función de este error son controlados, durante el proceso de alineación, el sentido y la velocidad de giro del motor, empleando modulación de ancho de pulso generada en la PC.

Análisis de deformación por Perspectiva: El objetivo del análisis de perspectiva es, en este caso, la determinación de la posición del robot respecto del tablero (Ver Figura 3):

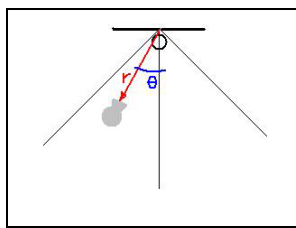


Figura 3

Para ello, el robot toma medidas de las alturas en píxeles de las tres franjas y la distancia en píxeles entre las franjas laterales. Debido a leyes de la perspectiva, las magnitudes aparentan ser menores conforme crece la distancia al observador (Ver Figura 4):

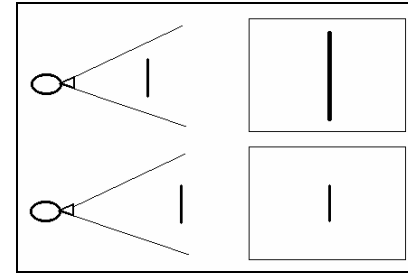


Figura 4

Basados en este fenómeno encontramos una ecuación que relaciona la altura en píxeles de la franja central con la distancia del robot a dicha franja:

$$\text{Distancia} = k / (\text{Altura en píxeles franja central})$$

Donde "k" es una constante de proporcionalidad.

El método desarrollado para calcular el ángulo que forma el tablero con el eje óptico de la cámara, se basa en la observación de que el tamaño horizontal aparente de la figura disminuye conforme el robot se aleja del eje ortogonal al tablero manteniendo un radio constante.

La siguiente fórmula aproxima el ángulo buscado:

$$\text{Angulo} = \text{COS}^{-1}(\text{Ancho aparente} / \text{Altura aparente})$$

Control de Presión en Lazo Cerrado: El sistema de propulsión del cañón es neumático, por lo que se implementó un sistema de control de presión en lazo cerrado por software, el cual emplea un transductor de presión como elemento sensor y varias electroválvulas como actuadores, a fin de controlar la presión de disparo en un tanque "vejiga".

Interpolación Bidimensional: En el sistema se manejan dos parámetros de disparo: la presión en el tanque "vejiga" necesaria para que el balón sea lanzado hasta el tablero, y la corrección angular necesaria para que el balón impacte en el punto exacto del tablero que lo haga encestar de rebote.

Existe para cada punto de la cancha un único par (presión, corrección angular) que logra el disparo óptimo. Luego, existen dos funciones ideales, una de presión y otra de corrección angular, que tienen como dominio la cancha, considerada ésta como un espacio bidimensional de posiciones. Estas funciones pueden ser visualizadas como superficies continuas sobre la cancha. Para optimizar el proceso de calibración decidimos aproximar dichas superficies mediante planos triangulares construidos a partir de una colección reducida de puntos. El

robot debe ser “entrenado” en estos puntos de modo que pueda generar una tabla de referencia. El entrenamiento consiste en posicionar al robot y hacerlo disparar con presión y corrección ingresadas por teclado. Los ensayos se repiten varias veces hasta lograr encestes holgados. Entonces se informa al programa que el disparo fue bueno y el programa guarda los parámetros en una base de datos, asociándolos con las coordenadas del punto de lanzamiento determinadas por el método antes descrito.

Luego del entrenamiento el robot es capaz de interpolar valores adecuados de presión y corrección angular para cualquier punto nuevo, valiéndose del plano formado por los tres puntos de referencia más cercanos a él.

Sistema de propulsión: El principal problema a la hora de diseñar el sistema de propulsión fue el de establecer que tipo de pistón se iba a utilizar para poder impulsar la bola a una velocidad media de 7m/s aproximadamente. Planteamos el problema considerando que el impacto del vástago del pistón con el balón puede ser aproximado a un choque elástico. Según este planteamiento, el pistón que nos podría brindar este tipo de choque era uno de doble efecto; alimentando una de las entradas se impulsa el pistón y con la otra entrada se frena el mismo. El segundo paso era realizar el cálculo de la transferencia de momento lineal del sistema pistón-bola, mediante la siguiente relación:

$$m_p v_p = m_b v_b$$

donde:

- m_p = masa del vástago del pistón
- v_p = velocidad del pistón
- m_b = masa de la bola
- v_b = velocidad de la bola

Si despejamos de la anterior ecuación la velocidad v_b , que es lo que nos interesa, obtenemos la siguiente relación:

$$v_b = m_p v_p / m_b$$

De las hojas técnicas obtuvimos el dato de la velocidad máxima que podía alcanzar el pistón, y ésta consistía en 1m/s. Con este dato observamos que la relación entre la masa del pistón y de la masa de la bola debería ser aproximadamente de 7 a 1. El peso de la bola era de 250 gramos, y por ende el peso del pistón debía ser de 1.5 Kilos, aproximadamente. Debido a este requerimiento se

tuvo que adaptar una pieza al vástago del pistón para aumentar su masa.

Descripción del sistema neumático: (Ver figura 5)

El compresor se encarga de alimentar tanto al pistón de propulsión como también al pistón de elevación. Expliquemos primero el tramo hacia el pistón de propulsión, el compresor alimenta a un tanque auxiliar donde se regula la presión. La carga del tanque auxiliar se controla mediante una electroválvula ON-OFF normalmente cerrada, en el tanque se tiene un sensor de presión que brinda información al controlador para poder determinar cuando se debe volver a cerrar la válvula. Si seguimos el circuito se observa que el circuito continúa hacia una válvula 5/2, por una de las entradas entra como alimentación, otras dos posiciones sirven para conectarlas al pistón y las dos ultimas como salidas de desfogue. Inicialmente la alimentación va hacia la entrada que permite que le vástago se retraiga, a la hora de realizar el disparo se hace que la electro válvula cambie de posición, la alimentación pasa a la entrada que permite mover el pistón para que se extienda e inmediatamente se vuelve a conmutar a la posición inicial y así se consigue la simulación de un choque elástico.

Luego de este ensayo de disparo el tanque auxiliar queda a una presión baja y no va a poder permitir otro disparo, esto es ideal para nuestro planteamiento ya que se tiene que regular distintas presiones para alcanzar distintas distancias. Así se podrá tener una condición inicial que nos permita cargar de nuevo el tanque para lograr el nuevo requerimiento de presión.

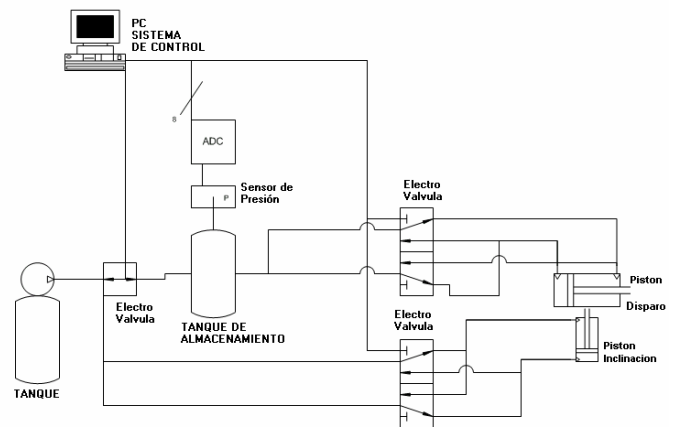


Figura 5

Si se desprecia la fricción del aire, se puede asumir que el desplazamiento de la pelota sigue una trayectoria parabólica.

Existen para la pelota infinitas posibles trayectorias parabólicas, que parten en cualquier punto de

disparo y pasan por el centro del aro. Cada una corresponde a un juego de valores de ángulo y presión de disparo.

De estas trayectorias debimos escoger las de más alta probabilidad de acierto.

El ángulo de llegada de la pelota, formado por la trayectoria y el plano que contiene al aro, debía ser grande. De este modo, el aro efectivo "visto" por la pelota (la proyección del aro sobre el plano ortogonal a la trayectoria) sería lo suficientemente grande para que la pelota atravesara el aro sin chocar con él. Según este criterio lo ideal sería lanzar la pelota lo más alto posible

Existe también un lanzamiento ideal, que requiere mínima energía pero disminuye la probabilidad de acierto por tener un ángulo de llegada demasiado riesgoso. El lanzamiento ideal forma un ángulo de llegada determinado por la relación de tamaños entre la pelota y el aro.

Tuvimos que hallar un compromiso entre una mayor probabilidad de acierto y un bajo requerimiento de energía.

Decidimos conservar para los lanzamientos un ángulo de llegada de aproximadamente 55 grados. Para lograr basta mantener fijo el ángulo de disparo y variar la presión.

Para un ángulo de disparo $\alpha = 60$ fijo y a velocidades adecuadas, los lanzamientos se ven como la figura 6:

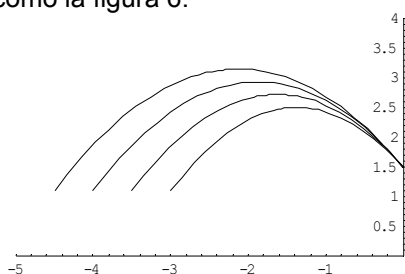


Figura 6

Sin embargo para lanzamientos desde puntos más cercanos, los requerimientos de energía disminuyen por lo que es posible mejorar la probabilidad de acierto si se busca un ángulo de llegada mayor.

Sistema de Control Basado en una PC: Para el control del robot se decidió usar una computadora bajo una plataforma Windows y emplear una cámara digital "WebCam" de Logitech, que permite realizar la adquisición de video a una velocidad de hasta 30

cuadros por segundo. Además el sistema de control se basaría en la adquisición de imágenes las cuales por un proceso e interpretación darían como resultado 2 variables, la presión de disparo para el pistón y la inclinación del sistema de disparo, por otro lado el robot debe encontrar una señal la cual lo alineará con el aro y así podrá calcular la distancia a la que se encuentra. Para poder controlar la presión se ha utilizado un conversor análogo digital (ADC) con resolución de 8 bits lo cual satisface para poder leer la presión a través de un transductor de presión y convertirla en voltaje, dicha presión es la presión del tanque el cual alimentará al pistón en el momento del disparo.

El control de presión es mediante lazo cerrado ya que se sensa la presión del tanque hasta que llegue a la adecuada y mediante electroválvulas se controla dicha presión, y por otras electroválvulas se controla el disparo del pistón..

El Robot cuenta con un control remoto para su traslación en un campo de radio de 5 metros aproximadamente

CONCLUSIÓN

La visión artificial es una excelente herramienta para la obtención de información en ciertas aplicaciones. En este trabajo se encuentra formando parte de un sistema de control, en donde cumple la importante misión de analizar e interpretar las imágenes provenientes del entorno del robot y entregar a su sistema de control las variables primordiales necesarias para la ejecución del objetivo principal: encestar el balón dentro del aro de basketball.

REFERENCIAS

Neumática Industrial
Benjamín Barriga Gamarra
Lima, PUCP 1994

Circuitos Neumáticos
Benjamín Barriga Gamarra
Lima, PUCP 1987.

www.norgren.com

Sistema de Localización y Posicionamiento de Piezas utilizando Visión Artificial

Javier de Lope, Francisco Serradilla, José G. Zato

Resumen

En el presente artículo describimos un sistema automatizado para reconocimiento, localización y posicionamiento de piezas de fabricación para la industria aeronáutica. Dichas piezas, compuestas de un material flexible, son depositadas de forma automática en una mesa de trabajo de $2 \times 2 \text{ m}^2$ provenientes de un proceso de automatización anterior. La tarea a desarrollar por el sistema consiste en una primera identificación y validación de la pieza para, posteriormente, encontrar su localización exacta y conducirla a una posición y orientación final gracias a un robot de tipo pórtico. Las especificaciones del sistema indican que el error de $\pm 0.5 \text{ mm}$. Para alcanzar la precisión requerida, el sistema dispone de dos cámaras de visión artificial; la primera obtiene una panorámica general, y la segunda, moviéndose conjuntamente con el efector del pórtico, permite tomar una imagen muy próxima al objeto y por tanto mucho más precisa. Esta arquitectura doble permite abaratar considerablemente el hardware del sistema gracias al uso

1 Introducción

Dentro de los procesos de fabricación de material aeronáutico existe una fase denominada conformado de patrones, cuyo objetivo es reforzar mediante un recubrimiento diversas partes de las aeronaves. Estos patrones están compuestos de fibra de carbono, fibra de vidrio o de kevlar. La mecanización del proceso de conformado consta de tres etapas: el cortado del patrón, su colocación en el útil de forrado, y el forrado definitivo de la pieza. Una vez cortado el patrón se deposita en una mesa de trabajo, donde deberá ser reconocido y posicionado en una zona determinada para su colocación en el útil.

Este artículo se centra en el diseño e implementación de un sistema *software* que, utilizando técnicas de visión artificial, realice las tareas de reconocimiento, localización y posicionamiento de los patrones sobre la mesa de trabajo. Como requisito inicial se cuenta con que el error de posicionamiento en las piezas debe ser menor que $\pm 0.5 \text{ mm}$ en posición y $\pm 0.5^\circ$ en orientación. A tal efecto, hemos utilizado los siguientes elementos:

- Un robot pórtico con cuatro grados de libertad (tres de traslación en los ejes X, Y, Z y uno de rotación) con su correspondiente controlador que opera sobre la mesa de trabajo de dimensiones de $2 \times 2 \text{ m}^2$.
- Dos cámaras de visión CCD Sony conectadas a una tarjeta digitalizadora Matrox IP-8. Una de las cámaras se utiliza para tomar imágenes lejanas de la zona de trabajo completa. La otra, solidaria con el efector del robot, se utiliza para tomar imágenes cercanas de una región de la pieza y refinar las
- Un ordenador 486, encargado de procesar las imágenes y enviar las instrucciones de posicionamiento al controlador del robot.

Para cumplir los objetivos definidos, el sistema *software* se ha articulado en una serie de fases o etapas que se enumeran a continuación:

Preproceso y segmentación. Dada una imagen en 256 niveles de gris debemos ser capaces de obtener una imagen binaria, donde los puntos pertenecientes a la pieza y los puntos del fondo se etiquetarán con diferentes valores.

Reconocimiento y validación. Utilizando la imagen tomada con la cámara lejana, previamente segmentada, debemos reconocer el tipo de pieza que se encuentra sobre la superficie de trabajo. Una vez efectuada

. La cámara adosada al robot p $\acute{o$ rtico se env \acute{i} a a la posici $\acute{o$ n del v \acute{e} rtice de referencia. En este punto se toma una nueva imagen cercana de la pieza con la que se puede determinar la posici $\acute{o$ n del v \acute{e} rtice de referencia con el error requerido.

Posicionamiento. Utilizando el modelo geom \acute{e} trico de la pieza se calcula su posici $\acute{o$ n, referida al centro de masas, y el movimiento (traslaci $\acute{o$ n y rotaci $\acute{o$ n) necesario para depositar la pieza en su punto de destino. Seguidamente se env \acute{i} an las instrucciones apropiadas al controlador del robot.

Una restricci $\acute{o$ n adicional del sistema es que debe ser capaz de funcionar en tiempo real. Por ello, los algoritmos encargados de llevar a cabo las diferentes tareas deben ser cuidadosamente seleccionados.

2 Caracterizaci $\acute{o$ n Geom \acute{e} trica

Para manejar de forma conveniente la informaci $\acute{o$ n aportada por las c \acute{a} maras e integrarla en comandos de movimiento al brazo mec \acute{a} nico, se ha efectuado una caracterizaci $\acute{o$ n geom \acute{e} trica del entorno de operaci $\acute{o$ n mediante la definici $\acute{o$ n de una serie de sistemas de referencia que incluye los asociados a las c \acute{a} maras lejana $R_L = \{O_L; x_L, y_L, z_L\}$ y cercana $R_C = \{O_C; x_C, y_C, z_C\}$. Un esquema de esta caracterizaci $\acute{o$ n se muestra en la Figura 1.

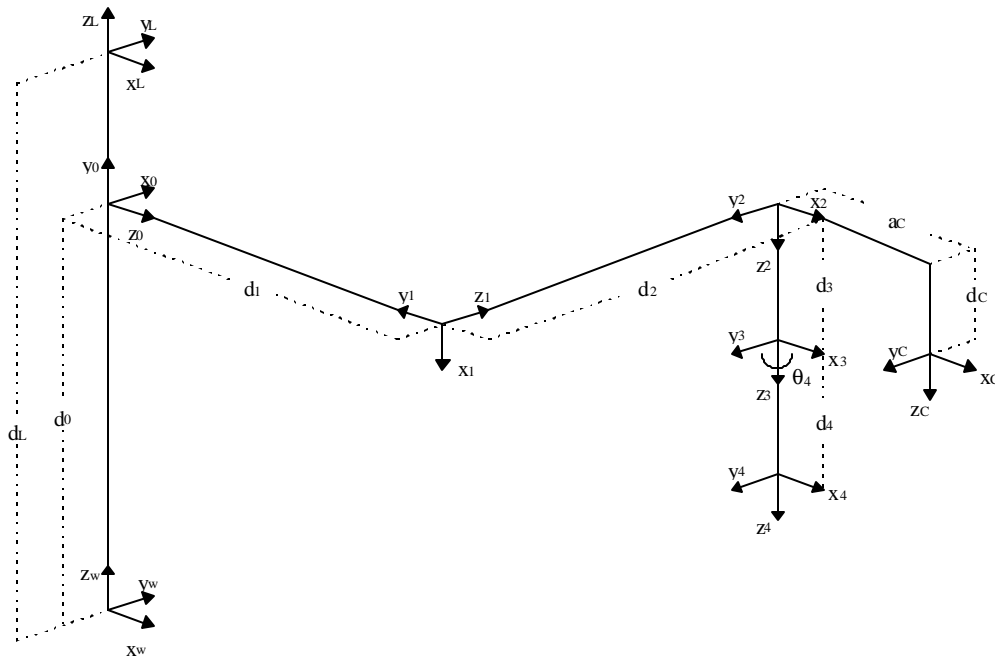


Figura 1: Esquema geométrico del entorno de operación

Utilizando los mecanismos habituales, se han obtenido los parámetros de Denavit y Hartenberg [Denavit, 55] que permitirán definir las expresiones de cambio de sistemas de referencia, que se relacionan en la Tabla 1. En la tabla se muestran entre paréntesis los parámetros que se dejan libres para calibrar algunas de las partes de los elementos mecánicos. En concreto, d_0 es la altura a la que se sitúa el p $\acute{o$ rtico del sistema de referencia base o del mundo, d_4 es la longitud del efector, a_C y d_C son las distancias de separaci $\acute{o$ n y altura de la c \acute{a} mara cercana desde el referencial final del segundo segmento del manipulador, respectivamente, y d_L es la altura a la que est \acute{a} colocada la c \acute{a} mara lejana desde el sistema de referencia del

d_1 , d_2 , d_3 y θ_4 se dejan variables para expresar las coordenadas articulares del robot y definir la posición y orientación del efector y de la cámara cercana con respecto del sistema de referencia del mundo. Las expresiones matriciales (siguiendo la notación de [Schilling, 90]) que expresan estos cambios de referenciales se muestran a continuación:

$$T_{Base}^{Efector} = \begin{pmatrix} \cos \mathbf{q}_4 & -\text{sen} \mathbf{q}_4 & 0 & d_1 \\ -\text{sen} \mathbf{q}_4 & -\cos \mathbf{q}_4 & 0 & d_2 \\ 0 & 0 & -1 & d_0 - d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

$$T_{Base}^{Cercana} = \begin{pmatrix} 1 & 0 & 0 & d_1 + a_c \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & -1 & d_0 - d_c \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2)$$

3 Preproceso y Segmentación

Antes de poder operar con la imagen de la pieza, deben ser discriminados los *pixels* correspondientes a la pieza de los del fondo. En nuestro problema estas tareas no son excesivamente complejas, debido a que las piezas son de color negro y la mesa de trabajo de color blanco. Este hecho produce diferencias acusadas entre los puntos de la imagen que corresponden a la pieza y los que corresponden a la mesa. Por tanto no es necesario ningún tipo de preproceso y la segmentación puede realizarse umbralizando la imagen original —en 256 niveles de gris— con un umbral fijo establecido en 128. A los puntos con valores de intensidad por encima o iguales a 128 asignaremos un valor de 255 (pieza) y a los puntos por debajo de 128 asignaremos un valor de 0 (fondo).

El único problema encontrado en esta fase surgió del hecho de que las piezas presentaban un tramado de agujeros apreciable en las imágenes captadas por la cámara cercana. Este problema se resolvió utilizando técnicas de morfología matemática, realizando un cierre de la imagen [Dougherty, 92].

4 Reconocimiento y Validación

Aunque las especificaciones del proyecto solicitaban solamente efectuar una validación de la pieza en términos de aceptación o rechazo, dependiendo de si la pieza era o no la esperada, esta validación requería en sí misma una identificación previa y, posteriormente, una comprobación. Desde este punto de vista, el problema puede abordarse desde la Teoría de Reconocimiento de Formas.

4.1 Piezas Prototipo y Piezas Percibidas

Según la Teoría de Reconocimiento de Formas, cada pieza que llega al sistema pertenece a alguna de las posibles clases o patrones de un universo de trabajo. Cada clase va a estar representada por un prototipo, que en nuestro caso se construirá a partir del modelo geométrico de cada tipo de pieza. Tanto los prototipos como las piezas reales percibidas se describen a través de un conjunto de valores numéricos llamados *vector de características*, que usualmente se obtienen a partir de las imágenes tomadas con la cámara. Estos valores se agrupan en un vector, llamado vector de características. Como aportación original, en nuestro proyecto extraemos el vector de características de los prototipos a partir del modelo geométrico de la pieza. En la Figura 2 se muestra el esquema general de un reconocedor de formas.

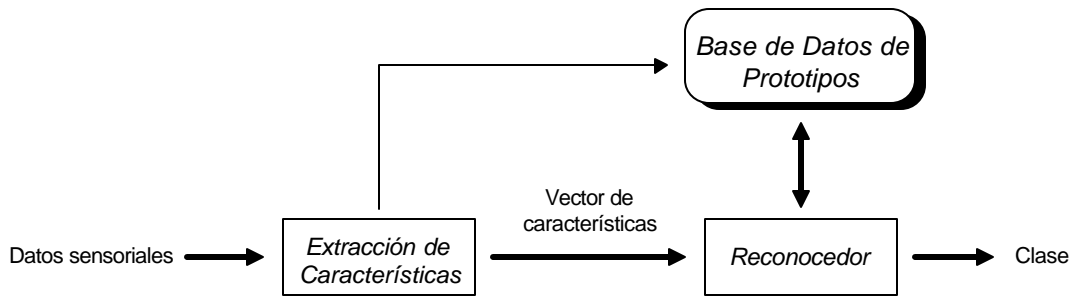


Figura 2: Esquema general de un reconocedor de formas

Tenemos, por tanto, dos clases o categorías de piezas:

- **Piezas prototipo:** capturan la esencia de todas las piezas pertenecientes a una misma clase. La práctica habitual es extraer el vector de características del prototipo de cada clase promediando los vectores de un gran número de ocurrencias de individuos de dicha clase. En nuestro problema, sin embargo, esta esencia se encuentra recogida en el modelo geométrico de la pieza, por ello el vector de características se extraerá a partir de este modelo. En primer lugar se dibujará la pieza en la pantalla, y a partir de esta imagen calcularemos los valores para las características. Algunas características, no obstante, se calcularán directamente a partir del modelo.
- **Piezas percibidas:** responden a ocurrencias de una clase, es decir, son piezas reales captadas por una cámara. El vector de características que describe a una pieza percibida se obtiene a partir de la imagen captada, previamente segmentada.

El sistema dispone de una base de datos que contiene el modelo geométrico de cada pieza prototipo. En la primera versión del sistema la geometría de las piezas está restringida a cuadriláteros. Según esto, su geometría puede describirse suministrando los cuatro vértices del modelo. A partir de ellos el sistema

4.2 Descripción de las Piezas

Como hemos comentado, el vector de características agrupa valores numéricos de características de cada pieza. En este caso, dichos valores numéricos deben ser convenientemente normalizados para evitar que influyan de modo desigual en el proceso de decisión. El vector consta de los siguientes datos:

- Área de la pieza
- Los 7 momentos de Hu

Los momentos de Hu [Sonka, 93] son momentos invariantes a traslaciones, rotaciones y cambio de escala, que pueden calcularse a partir de los momentos centrales normalizados de orden 2 y 3 utilizando la ecuación (3).

\overline{VC} que une el vértice de referencia $\overset{p}{V}_r$ con el centro de masas $\overset{p}{C}$. En segundo lugar calcularemos el eje de inercia $\overset{p}{E}_i$ del modelo geométrico. Del segmento \overline{VC} nos interesará determinar el ángulo α que forma con el eje de inercia $\overset{p}{E}_i$ y su longitud d .

Una vez determinado el par (α, d) para la pieza prototipo, estamos en condiciones de identificar el vértice de referencia en la pieza percibida. Los pasos que tendremos que dar sobre la imagen de la pieza son:

- Calcular el centro de masas.

- Calcular el eje de inercia.
- Utilizando el par (α, d) sobre el eje de inercia, obtener los dos puntos de la imagen candidatos a vértice de referencia.
- En estos dos puntos iniciaremos una búsqueda de vértice que deberá tener éxito en uno de los dos casos. Si tenemos éxito en los dos casos significa que la pieza es simétrica, no importando entonces

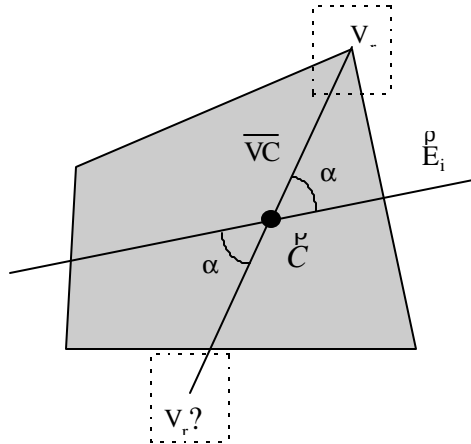


Figura 4: Determinación de la posición del vértice de referencia

5.4 Búsqueda del Vértice de Referencia

Como resultado del proceso anterior se ha obtenido una primera aproximación (dada la baja resolución proporcionada por la cámara lejana) a la posición del vértice de referencia. Para obtener un cálculo mucho más preciso, (a) desplazaremos la cámara cercana a la posición aproximada, (b) tomaremos una imagen con la cámara cercana y (c) procesaremos esta imagen para obtener con precisión la posición del vértice. El

- Se realizará un proceso morfológico de apertura [Dougherty, 92] para eliminar los agujeros que pudieran aparecer debido a la naturaleza del material.
- Se efectuará un proceso de barrido de la imagen para determinar las coordenadas de $2N$ puntos de la frontera (borde) de la pieza, N puntos por cada lado. Sin entrar en detalles, existen 12 posibles tipos de vértice, que determinan el proceso de barrido a aplicar.
- Con estos dos conjuntos de N puntos determinaremos dos rectas de regresión que aproximarán los lados que cortan en el vértice de referencia.
- Calculando el punto de corte de las rectas de regresión obtenemos la posición del vértice de referencia.

6 Posicionamiento Final de la Pieza

Una vez conocidos el vértice de referencia y el eje de mínima inercia se puede calcular con la precisión requerida el lugar en el que se encuentra el centro de masas de la pieza. A ese lugar se enviará el efector del robot para, después de una traslación en su eje vertical con objeto de aprisionar la pieza contra la mesa, efectuar los movimientos (rotación y traslación) que la llevarán desde su posición inicial hasta la final.

Estos movimientos, tanto los de acercamiento como los de posicionamiento, se realizan en base a los sistemas de referencia definidos en la caracterización geométrica del entorno y utilizando las expresiones matriciales dadas en la ecuaciones 1 y 2. Este proceso se realiza en los siguientes pasos:

- transformación de las coordenadas del centro de masas referidas al sistema de referencia R_C de la cámara cercana al sistema de referencia del mundo,
- cálculo de las coordenadas articulares finales que describen el movimiento del robot hasta ese punto, en el sistema de referencia del mundo,
- realización de la rotación y traslación de la pieza hasta la posición final, en el sistema de referencia del mundo, y
- liberación de la pieza y retirada del efector de la zona de trabajo.

7 Conclusiones

Es este artículo hemos presentado el proyecto FAMA (Fabricación Avanzada de Material Aeronáutico), un prototipo que incluye todas las fases clásicas [Fu, 89] de un sistema de visión artificial. Su mayor interés radica en la utilidad, versatilidad y funcionamiento en tiempo real (a una tasa de unas 3 piezas por minuto) y en la integración con el control de un brazo robótico de tipo pórtico.

Hemos proporcionado la definición de un proceso de producción adaptado al problema, en el que engarzan elementos de análisis de imágenes con elementos de control del brazo, permitiendo la utilización de un sistema doble de cámaras que, abaratando costes, proporciona una precisión equivalente a la de una cámara de alta

Del mismo modo, se ha caracterizado la geometría del sistema integrado brazo/cámara-cercana/cámara-lejana y obtenido sus parámetros de Denavit-Hartenberg, lo que permite la modelización del entorno de operación del robot. Así mismo, la parametrización del sistema hace posible el ajuste por calibración de los aspectos variables de los elementos constitutivos, tales como la orientación y altura de las cámaras.

Los procedimientos de clasificación utilizados proporcionan un rendimiento óptimo para el conjunto de piezas a tratar. No obstante, gracias a su diseño modular, el sistema puede ser fácilmente modificado o ampliado para incorporar nuevas características o procesos de clasificación más complejos. En lo referente al posicionamiento, los errores obtenidos están por debajo de los requisitos iniciales del proyecto.

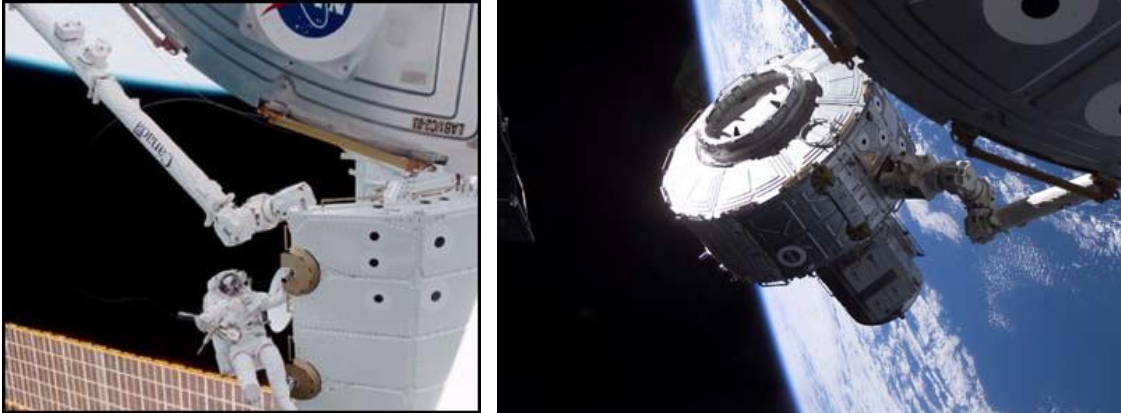
Las técnicas de coordinación de varias cámaras para conseguir un objetivo común empleadas en este proyecto podrían ser utilizadas en otros de características similares, tales como la clasificación de documentos, inspección visual de defectos de fabricación, vigilancia en sistemas de seguridad, etc.

Referencias

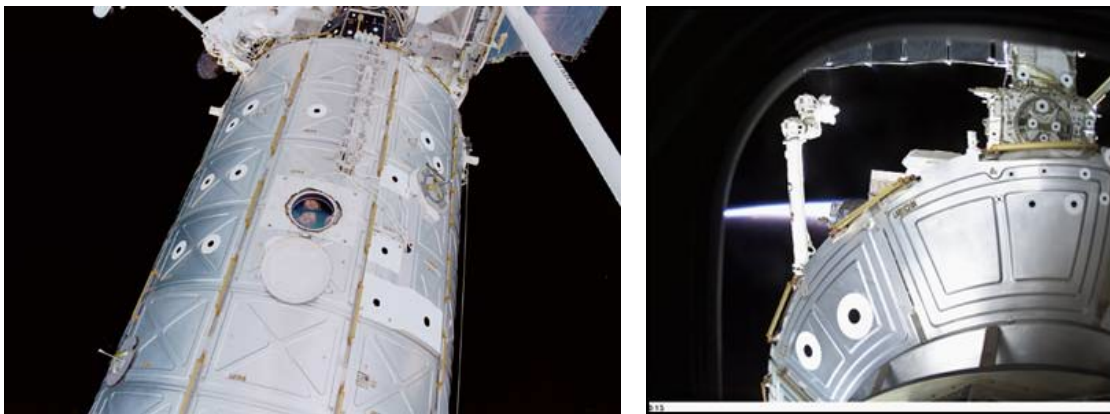
- [Denavit, 55] J. Denavit, R.S. Hartenberg. "A Kinematic Notation for Lower-Pair Mechanics based on Matrices", *ASME J. Applied Mechanics*, June 1955, pp. 215–221.
- [Dougherty, 92] R.E. Dougherty. *An Introduction to Morphological Image Processing*. SPIE-The International Society for Optical Engineering, 1992.
- [Fu, 89] K.S. Fu, R.C. González; C.S.G. Lee. *Robotics: Control, Sensing, Vision, and Intelligence*. McGraw-Hill, New York, 1989.
- [Maravall, 93] D. Maravall. *Reconocimiento de Formas y Visión Artificial*. Ra-Ma, Madrid, 1993.
- [Pratt, 91] W.K. Pratt. *Digital Image Processing*. John Wiley & Sons, New York, 1991.
- [Schilling, 90] R.J. Schilling. *Fundamentals of Robotics. Analysis and Control*. Prentice Hall, New Jersey, 1990.
- [Sonka, 93] M. Sonka, V. Hlavac, R. Boyle. *Image Processing, Analysis and Machine Vision*. Chapman & Hall, 1993.

The Space Vision System

Lloyd Pinkney's Space Vision System (SVS) represents a stellar breakthrough in space technology. Related to other Canadian space innovation such as the robotic arms Canadarm and **Canadarm2**, SVS is an eye in space, and will facilitate construction of the new **International Space Station**.



When building in space, not only is the lack of gravity a factor, but, in the black emptiness of space, there are also no points of reference to judge the location and speed of objects being used in the construction. Space Vision System provides those critical reference points, and calculates and displays the location, orientation and speed at which the objects are moving in relation to the Canadarm, the space shuttle and the space station, as well as a view of what the Canadarm, attached to a space shuttle, is doing.



Imagine yourself floating in space, looking at the outside of the International Space Station. What you may first notice are numerous black dots. These dots are reference points for the SVS, which, attached to the Canadarm, extrapolate the distance to the circles or any other objects being moved. This is accomplished by taking sequential photographs of the objects every 33 milliseconds, with SVS further calculating the location and speed of the objects before the next photograph is even taken. A display screen located inside the space shuttle or the space station presents an image of the object in terms of its location, orientation and speed.



The Space Vision System, therefore, is enabling astronauts to use the Canadarm2 to construct most of the station in orbit and from within the space station itself. The Canadarm2, larger and more dexterous than its predecessor, is capable of moving an object the size of the space shuttle itself, and SVS makes it possible to maneuver these objects with extreme accuracy. Canadian astronaut Chris Hadfield often speaks of how, during trials, SVS gave him location information of the robotic arm and its payload to within a millimetre. For example, a massive airlock was installed on the station using the arm and SVS without incident.

Astoundingly accurate, SVS is playing an essential role in the construction of the International Space Station. It also is one of Alberta's links to the study of space—Lloyd Pinkney was born in Blairmore in 1931 and later attending the

University of Alberta where he studied engineering and physics. Following his studies, Pinkney joined the National Research Council.

http://www.abheritage.ca/abinvents/inventions/inv_ot_space_vision.htm

Apéndice F

Hojas de especificaciones

DM74LS04 Hex Inverting Gates

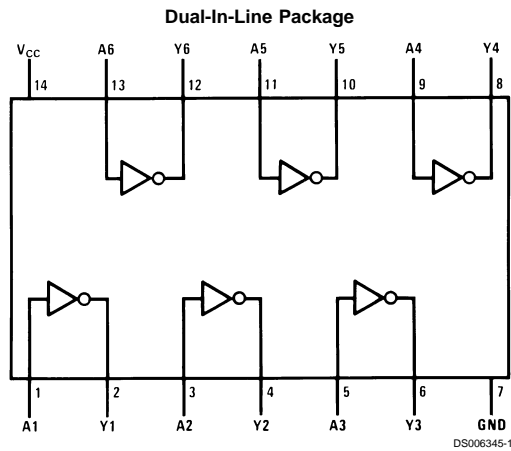
General Description

This device contains six independent gates each of which performs the logic INVERT function.

Features

- Alternate Military/Aerospace device (54LS04) is available. Contact a Fairchild Semiconductor Sales Office/Distributor for specifications.

Connection Diagram



Order Number 54LS04DMQB, 54LS04FMQB, 54LS04LMQB, DM54LS04J, DM54LS04W, DM74LS04M or DM74LS04N
See Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$$Y = \bar{A}$$

Input	Output
A	Y
L	H
H	L

H = High Logic Level
L = Low Logic Level

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V	DM54LS and 54LS	-55°C to +125°C
Input Voltage	7V	DM74LS	0°C to +70°C
Operating Free Air Temperature Range		Storage Temperature Range	-65°C to +150°C

Recommended Operating Conditions

Symbol	Parameter	DM54LS04			DM74LS04			Units
		Min	Nom	Max	Min	Nom	Max	
V _{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V _{IH}	High Level Input Voltage	2			2			V
V _{IL}	Low Level Input Voltage			0.7			0.8	V
I _{OH}	High Level Output Current			-0.4			-0.4	mA
I _{OL}	Low Level Output Current			4			8	mA
T _A	Free Air Operating Temperature	-55		125	0		70	°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = -18 mA			-1.5	V
V _{OH}	High Level Output Voltage	V _{CC} = Min, I _{OH} = Max, V _{IL} = Max	DM54 2.5	3.4		V
V _{OL}	Low Level Output Voltage	V _{CC} = Min, I _{OL} = Max, V _{IH} = Min	DM54 0.25	0.35	0.4	V
		I _{OL} = 4 mA, V _{CC} = Min	DM74 0.25		0.4	
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 7V			0.1	mA
I _{IH}	High Level Input Current	V _{CC} = Max, V _I = 2.7V			20	µA
I _{IL}	Low Level Input Current	V _{CC} = Max, V _I = 0.4V			-0.36	mA
I _{OS}	Short Circuit Output Current	V _{CC} = Max	DM54 -20		-100	mA
		(Note 3)	DM74 -20		-100	
I _{CCH}	Supply Current with Outputs High	V _{CC} = Max		1.2	2.4	mA
I _{CCL}	Supply Current with Outputs Low	V _{CC} = Max		3.6	6.6	mA

Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	R _L = 2 kΩ				Units
		C _L = 15 pF		C _L = 50 pF		
		Min	Max	Min	Max	
t _{PLH}	Propagation Delay Time Low to High Level Output	3	10	4	15	ns
t _{PHL}	Propagation Delay Time High to Low Level Output	3	10	4	15	ns

Note 2: All typicals are at V_{CC} = 5V, T_A = 25°C.

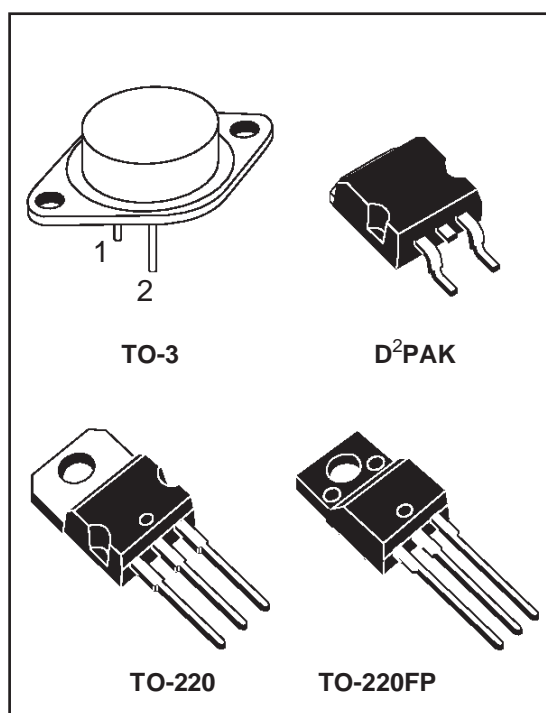
Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

POSITIVE VOLTAGE REGULATORS

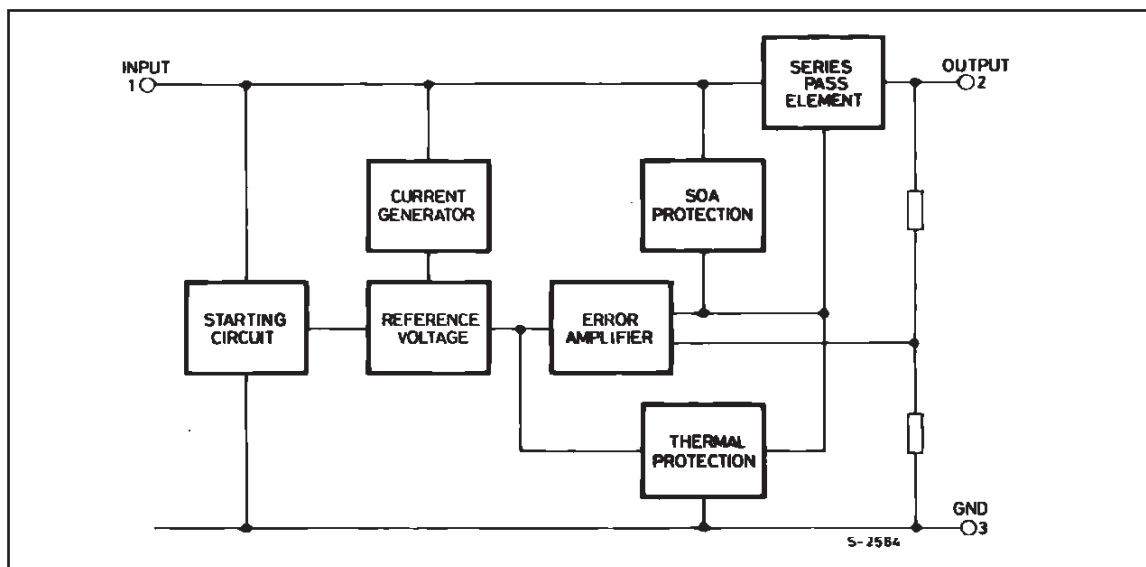
- OUTPUT CURRENT UP TO 1.5 A
- OUTPUT VOLTAGES OF 5; 5.2; 6; 8; 8.5; 9; 12; 15; 18; 24V
- THERMAL OVERLOAD PROTECTION
- SHORT CIRCUIT PROTECTION
- OUTPUT TRANSITION SOA PROTECTION

DESCRIPTION

The L7800 series of three-terminal positive regulators is available in TO-220 TO-220FP TO-3 and D²PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



BLOCK DIAGRAM



L7800

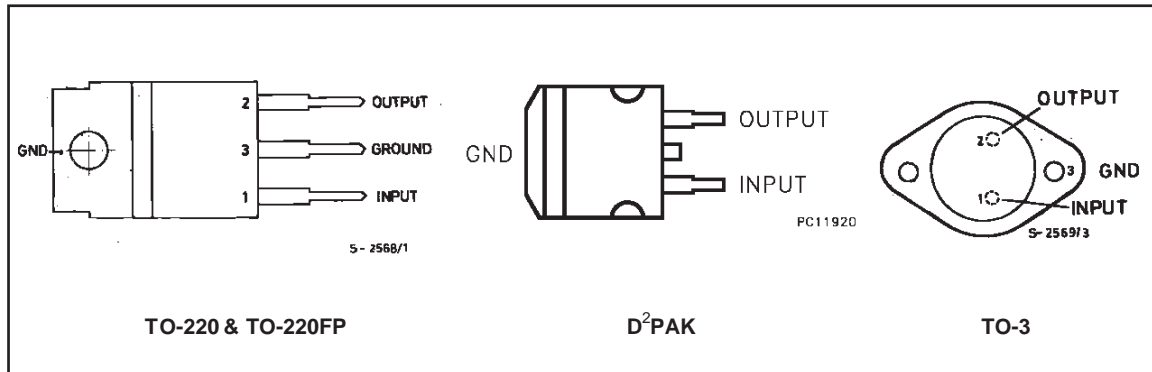
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_i	DC Input Voltage (for $V_O = 5$ to 18V) (for $V_O = 20, 24$ V)	35	V
		40	V
I_o	Output Current	Internally limited	
P_{tot}	Power Dissipation	Internally limited	
T_{op}	Operating Junction Temperature Range (for L7800) (for L7800C)	-55 to 150	°C
		0 to 150	°C
T_{stg}	Storage Temperature Range	-65 to 150	°C

THERMAL DATA

Symbol	Parameter	D ² PAK	TO-220	TO-220FP	TO-3	Unit
$R_{thj-case}$	Thermal Resistance Junction-case Max	3	3	5	4	°C/W
$R_{thj-amb}$	Thermal Resistance Junction-ambient Max	62.5	50	60	35	°C/W

CONNECTION DIAGRAM AND ORDERING NUMBERS (top view)



Type	TO-220	D ² PAK (*)	TO-220FP	TO-3	Output Voltage
L7805				L7805T	5V
L7805C	L7805CV	L7805CD2T	L7805CP	L7805CT	5V
L7852C	L7852CV	L7852CD2T	L7852CP	L7852CT	5.2V
L7806				L7806T	6V
L7806C	L7806CV	L7806CD2T	L7806CP	L7806CT	6V
L7808				L7808T	8V
L7808C	L7808CV	L7808CD2T	L7808CP	L7808CT	8V
L7885C	L7885CV	L7885CD2T	L7885CP	L7885CT	8.5V
L7809C	L7809CV	L7809CD2T	L7809CP	L7809CT	9V
L7812				L7812T	12V
L7812C	L7812CV	L7812CD2T	L7812CP	L7812CT	12V
L7815				L7815T	15V
L7815C	L7815CV	L7815CD2T	L7815CP	L7815CT	15V
L7818				L7818T	18V
L7818C	L7818CV	L7818CD2T	L7818CP	L7818CT	18V
L7820				L7820T	20V
L7820C	L7820CV	L7820CD2T	L7820CP	L7820CT	20V
L7824				L7824T	24V
L7824C	L7824CV	L7824CD2T	L7824CP	L7824CT	24V

(*) AVAILABLE IN TAPE AND REEL WITH "-TR" SUFFIX



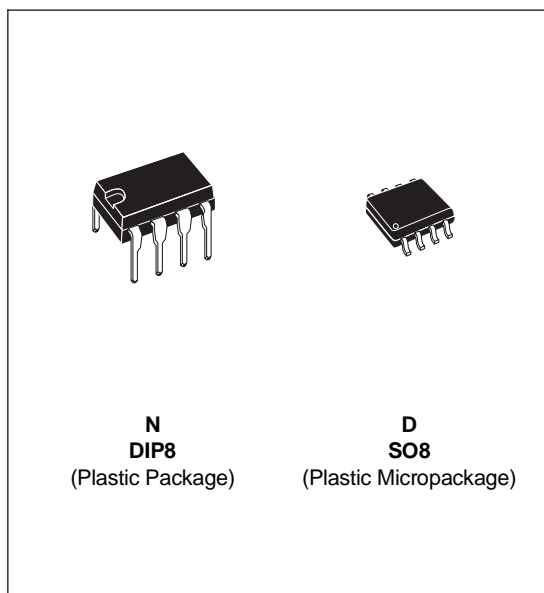
NE555 SA555 - SE555

GENERAL PURPOSE SINGLE BIPOLAR TIMERS

- LOW TURN OFF TIME
- MAXIMUM OPERATING FREQUENCY GREATER THAN 500kHz
- TIMING FROM MICROSECONDS TO HOURS
- OPERATES IN BOTH ASTABLE AND MONOSTABLE MODES
- HIGH OUTPUT CURRENT CAN SOURCE OR SINK 200mA
- ADJUSTABLE DUTY CYCLE
- TTL COMPATIBLE
- TEMPERATURE STABILITY OF 0.005% PER°C

DESCRIPTION

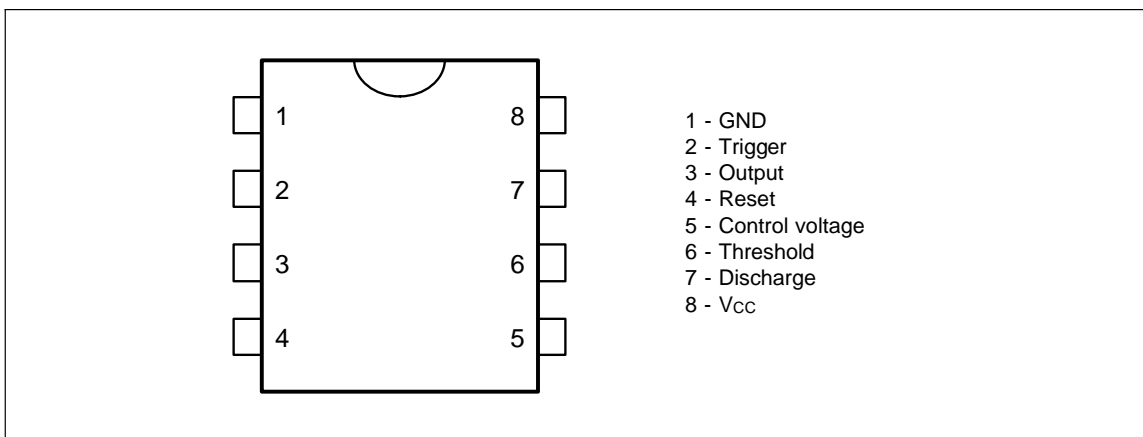
The NE555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays or oscillation. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200mA. The NE555 is available in plastic and ceramic minidip package and in a 8-lead micropackage and in metal can package version.



ORDER CODES

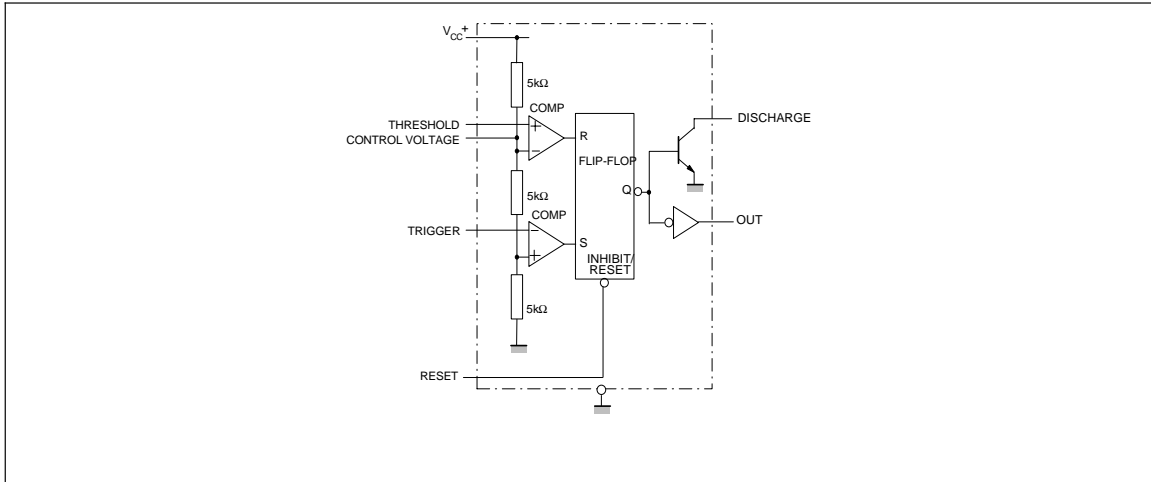
Part Number	Temperature Range	Package	
		N	D
NE555	0°C, 70°C	•	•
SA555	-40°C, 105°C	•	•
SE555	-55°C, 125°C	•	•

PIN CONNECTIONS (top view)

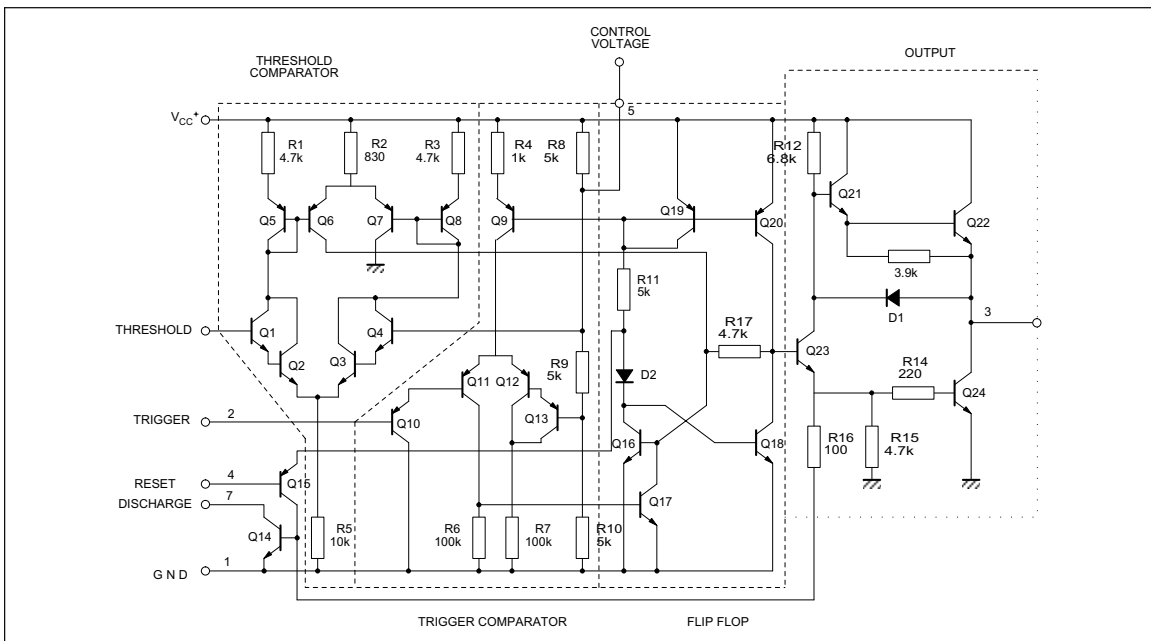


NE555/SA555/SE555

BLOCK DIAGRAM



SCHEMATIC DIAGRAM



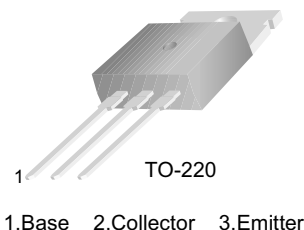
ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V _{CC}	Supply Voltage	18	V
T _{oper}	Operating Free Air Temperature Range	0 to 70 for NE555 -40 to 105 for SA555 -55 to 125	°C
T _j	Junction Temperature	150	°C
T _{stg}	Storage Temperature Range	-65 to 150	°C

TIP41 Series(TIP41/41A/41B/41C)

Medium Power Linear Switching Applications

- Complement to TIP42/42A/42B/42C



NPN Epitaxial Silicon Transistor

Absolute Maximum Ratings $T_C=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Value	Units
V_{CBO}	Collector-Emitter Voltage: TIP41	40	V
	: TIP41A	60	V
	: TIP41B	80	V
	: TIP41C	100	V
V_{CEO}	Collector-Emitter Voltage: TIP41	40	V
	: TIP41A	60	V
	: TIP41B	80	V
	: TIP41C	100	V
V_{EBO}	Emitter-Base Voltage	5	V
I_C	Collector Current (DC)	6	A
I_{CP}	Collector Current (Pulse)	10	A
I_B	Base Current	2	A
P_C	Collector Dissipation ($T_C=25^\circ\text{C}$)	65	W
P_C	Collector Dissipation ($T_a=25^\circ\text{C}$)	2	W
T_J	Junction Temperature	150	$^\circ\text{C}$
T_{STG}	Storage Temperature	- 65 ~ 150	$^\circ\text{C}$

Electrical Characteristics $T_C=25^\circ\text{C}$ unless otherwise noted

Symbol	Parameter	Test Condition	Min.	Max.	Units
$V_{CEO(sus)}$	* Collector-Emitter Sustaining Voltage	$I_C = 30\text{mA}, I_B = 0$	40		V
	: TIP41				
	: TIP41A				
	: TIP41B				
I_{CEO}	Collector Cut-off Current	$V_{CE} = 30\text{V}, I_B = 0$		0.7	mA
		$V_{CE} = 60\text{V}, I_B = 0$		0.7	mA
I_{CES}	Collector Cut-off Current	$V_{CE} = 40\text{V}, V_{EB} = 0$		400	μA
		$V_{CE} = 60\text{V}, V_{EB} = 0$		400	μA
		$V_{CE} = 80\text{V}, V_{EB} = 0$		400	μA
		$V_{CE} = 100\text{V}, V_{EB} = 0$		400	μA
I_{EBO}	Emitter Cut-off Current	$V_{EB} = 5\text{V}, I_C = 0$		1	mA
h_{FE}	* DC Current Gain	$V_{CE} = 4\text{V}, I_C = 0.3\text{A}$	30		
		$V_{CE} = 4\text{V}, I_C = 3\text{A}$	15	75	
$V_{CE(sat)}$	* Collector-Emitter Saturation Voltage	$I_C = 6\text{A}, I_B = 600\text{mA}$		1.5	V
$V_{BE(sat)}$	* Base-Emitter Saturation Voltage	$V_{CE} = 4\text{V}, I_C = 6\text{A}$		2.0	V
f_T	Current Gain Bandwidth Product	$V_{CE} = 10\text{V}, I_C = 500\text{mA}$	3.0		MHz

* Pulse Test: $PW \leq 300\mu\text{s}$, Duty Cycle $\leq 2\%$

Servomotor Futaba s3001



S3001 Standard		
Volts	Torque	Velocidad
4.8 V	44.4 oz-in.	0.23 seg / 60°
6.0V	56.9 oz-in.	0.19 seg / 60°
Dimensiones		Peso
1.6 x 0.8 x 1.4 in.		1.6 oz.

Especificaciones detalladas	
Control del Sistema:	+ Control de ancho de pulso 1520µs neutral
Pulso Requerido:	3 – 5 Volts pico a pico en señal cuadrada
Voltaje de Operación:	4.8 – 6.0 Volts
Rango de temperatura de operación:	- 20 a + 60 °C
Velocidad de operación (4.8V):	0.23 seg / 60°
Velocidad de operación (6.0V):	0.19 seg / 60°
Torque (4.8V):	44 oz / in. (3.2 Kg / cm.)
Torque (6.0V):	56.8 oz / in. (4.1 Kg / cm.)
Angulo de Operación:	45°
Modificable a 360:	Si
Dirección:	Sentido Horario a 1520-1900 µs
Consumo de corriente (4.8V):	7..2 mA
Consumo de corriente (6.0V):	8 mA
Tipo de motor:	3 polos de ferrita
Tipo de cubierta:	Cubierta de plástico
Tipo engranes:	Engranes de nylon
Longitud de cable conector:	12"
Dimensiones:	1.6 " x 8" x 1.4 " (41 x 20 x 36 mm)
Peso:	1.3 oz. (37.2 g)

Referencias

Bibliografía

Referencias Bibliográficas

- [1] Pajares M., Gonzalo., de la Cruz G, Jesús. "Visión por computador"
Ed. Ra-Ma. Colombia 2002.
- [2] R.M. Haralick and L.G. Shapiro. "Computer and robot vision"
Ed. Addison-Wesley Publishing Co., New York, 1993.
- [3] Pressman, Roger S. "Ingeniería del software. Un enfoque práctico"
Ed. McGraw-Hill. Madrid 2002
- [4] Malvino Albert p, Leach Donald P. "Principios y aplicaciones digitales"
Ed. Marcombo Boixareu editores, Barcelona 1993.
- [5] Spiegel Murria R. "Teoría y problemas de Probabilidad y Estadística" Serie Schaum. Ed.
McGraw Hill México 1982.

Referencias Electrónicas

- [a] Posgrado de la Facultad de Ingeniería, UNAM [en línea].
Disponible en: URL:<http://verona.fip.unam.mx/fardi/pagina/VISONCOM.htm>
Consultado Agosto 10, 2005.
- [b] Talledo J, Mayorga E. *Determinación de la posición de un objetivo por análisis de perspectiva, aplicada a un robot basketbolista.* [en línea]. 2002;
Disponible en: URL:<http://www.pucp.edu.pe/invest/conciencia/basket.pdf>
Consultado Julio 1, 2005
- [c] Alberta Online Encyclopedia [en línea]. 2003;
Disponible en: URL:http://www.abheritage.ca/abinvents/inventions/inv_ot_space_vision.htm
Consultado Julio 29, 2005.
- [d] De Lope J, Serradilla F. *Sistema de Localización y Posicionamiento de Piezas utilizando Visión Artificial.* [en línea]. 1997
Disponible en: URL:<http://tcs.uned.es:8080/aepia/Uploads/1/219.pdf>
Consultado Julio 6, 2005
- [e] Las imágenes digitales en dos dimensiones [en línea].
Disponible en: URL:<http://www.gusgsm.com/html/imagen.html>
- [f] *Gustavo Liñán Cembrano.* Introducción a MATLAB [en línea].
Disponible en:
URL:http://www.imse.cnm.es/tec_exp/downloads/enunciados/0506_INTRO_MATLAB.pdf
Consultado Octubre 10, 2005

- [g] Visual Basic: guía del estudiante [en línea]. 1997;
Disponible en: URL:<http://www.monografias.com/trabajos2/guiavb/guiavb.shtml>
Consultado Enero 6, 2006
- [h] The Mathworks: Matlab Automation Server Support [en línea]. 1994;
Disponible en:
URL:http://www.mathworks.com/access/helpdesk/help/techdoc/matlab_external/f27470.html
Consultado Enero 21, 2006
- [i] Color: Los errores [en línea].
Disponible en: URL:<http://www.coloryapariencia.com.ar/color.htm>
Consultado Diciembre 3, 2005